System Administration Guide, Volume 2

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part Number 805-7229–10
February 2000

Contents

About This Book 31

1. Managing Printing Services Topics 37

2. Print Management (Overview) 39

What's New in Printing? 39

Solaris Print Manager 39

Print Naming Enhancement 40

Enabling or Disabling Banner Page Printing 41

Printing in the Solaris Operating Environment 41

Choosing a Method to Manage Printers 42

The LP Print Service 43

Managing Network Printers 43

Administering Printers 44

Setting Definitions for Printers 44

Administering Character Sets, Filters, Forms, and Fonts 44

Customizing the LP Print Service 44

The Solaris Print Client-Server Process 45

The Print Client Process 45

Using Print Clients 45

Printer Configuration Resources 46

Using Print Servers 49

3. Planning Printers on Your Network (Overview) 51

Distributing Printers on the Network 51

Assigning Print Servers and Print Clients 52

Print Server Requirements and Recommendations 53

Spooling Space 53

Disk Space 53

Memory 54

Swap Space 54

Hard Disk 54

Planning for Printer Setup 54

Setting Definitions for Printers 55

Selecting a Printer Type 58

Selecting a File Content Type 59

4. Setting Up Printers (Tasks) 63

Setting Up Printing 63

Setting Up Printing Task Map 64

Setting Up a Printer With Solaris Print Manager 65

Starting Solaris Print Manager 66

▼ How to Start Solaris Print Manager 66

Setting Up a Print Server 69

▼ How to Add a New Attached Printer With Solaris Print Manager 70

Setting Up a Print Client 72

▼ How to Add Printer Access With Solaris Print Manager 72

Setting Up a `.printers` File 73

▼ How to Set Up a `.printers` File 73

Adding a Network Printer 74

Printer Vendor Supplied Software for Network Printers 75

Sun Support for Network Printers 76

Invoking the Network Printer Support 76

Selecting the Protocol 76

Selecting the Printer Node Name 77

Selecting the Destination (or Network Printer Access) Name 77

Setting the Timeout Value 78

Managing Network Printer Access 78

▼ How to Add a Network Printer Using Printer Vendor Supplied Tools 79

▼ How To Add A Network Printer Using LP Commands 79

Converting Printer Configuration Information 83

Converting Printer Configuration Information Task Map 83

Converting Existing Printer Configuration Information 84

▼ How to Convert Printer Information For a System Running the SunOS 5.5.1 Release 85

▼ How to Convert Printer Information For a System Running the SunOS 4.1 Release 85

How to Convert Printer Configuration Information in NIS+ (+xfn) to NIS+ Format 86

5. Administering Printers (Tasks) 87

Managing Printers and the Print Scheduler 88

Deleting Printers and Printer Access 88

▼ How to Delete a Printer and Remote Printer Access 88

Checking Printer Status 91

▼ How to Check the Status of Printers 91

Restarting the Print Scheduler 93

▼ How to Stop the Print Scheduler 93

▼ How to Restart the Print Scheduler 94

Setting or Resetting Miscellaneous Printer Definitions 94

▼ How to Add a Printer Description 94

Setting Up a Default Printer Destination 95

▼ How to Set a System's Default Printer 96

Printing Banner Pages 96

▼ How to Make Banner Pages Optional 97

▼ How to Turn Off Banner Pages 98

Setting Up Printer Classes 99

▼ How to Define a Class of Printers 100

Setting Up Printer Fault Alerts 100

▼ How to Set Fault Alerts for a Printer 101

Setting Up Printer Fault Recovery 102

▼ How to Set Printer Fault Recovery 104

Limiting User Access to a Printer 104

▼ How to Limit User Access to a Printer 106

Managing Print Requests 107

▼ How to Check the Status of Print Requests 108

Processing or Stopping Printing 110

▼ How to Accept or Reject Print Requests for a Printer 110

Accepting or Rejecting Print Requests 111

▼ How to Enable or Disable a Printer 113

Canceling a Print Request 114

▼ How to Cancel a Print Request 114

▼ How to Cancel a Print Request From a Specific User 115

Moving a Print Request 116

▼ How to Move Print Requests to Another Printer 117

Changing the Priority of Print Requests 118

▼ How to Change the Priority of a Print Request 118

6. Managing Character Sets, Filters, Forms, and Fonts (Tasks) 121

Managing Character Sets 122

Selectable Character Sets 122

Hardware-Mounted Character Sets 123

Tracking Print Wheels 124

Alerts for Mounting Print Wheels or Cartridges 124

▼ How to Define a Print Wheel or Font Cartridge 124

▼ How to Unmount and Mount a Print Wheel or Font Cartridge 125

▼ How to Set an Alert to Mount a Print Wheel or Font Cartridge 126

▼ How to Set Up an Alias for a Selectable Character Set 128

Managing Print Filters 130

Creating Print Filters 130

Adding, Changing, Removing, and Restoring Print Filters 131

▼ How to Add a Print Filter 132

▼ How to Delete a Print Filter 133

▼ How to View Information About a Print Filter 133

Managing Forms 135

Adding, Changing, or Deleting Forms 135

Mounting Forms 136

Tracking Forms 136

Defining Alerts for Mounting Forms 136

Checking Forms 136

Limiting Access to Forms 137

▼ How to Add a Form 137

▼ How to Delete a Form 138

▼ How to Unmount and Mount a Form 138

▼ How to Set an Alert to Mount a Form 140

▼ How to View Information About a Form 142

▼ How to View the Current Status of a Form 143

▼ How to Limit User Access to a Form 143

▼ How to Limit Printer Access to a Form 144

Managing Fonts 145

Managing Printer-Resident Fonts 146

Downloading Host-Resident Fonts 147

Installing and Maintaining Host-Resident Fonts 147

▼ How to Install Downloaded PostScript Fonts 148

▼ How to Install Host-Resident PostScript Fonts 148

7. Customizing the LP Print Service (Tasks) 151

Adjusting Printer Port Characteristics 151

▼ How to Adjust the Printer Port Characteristics 153

Adding a `terminfo` Entry for an Unsupported Printer 154

▼ How to Add a `terminfo` Entry for an Unsupported Printer 157

Customizing the Printer Interface Program 158

The Standard Printer Interface Program 159

Customizing `stty` Modes 159

Exit Codes 160

Fault Messages 161

Using a Customized Printer Interface Program 161

▼ How to Set Up a Custom Printer Interface Program 161

Creating a New Print Filter 163

Writing a Print Filter Program 163

Creating a Print Filter Definition 166

▼ How to Create a New Print Filter 172

Creating a New Printer Form 174

▼ How to Create a New Form Definition 177

8. LP Print Service Reference Information 179

The LP Print Service 179

The Structure of the LP Print Service 180

LP Print Service Commands 189

Functions of the LP Print Service 190

How LP Administers Files and Schedules Local Print Requests 191

Scheduling Network Print Requests 192

Filtering Print Files 193

What the Printer Interface Program Does 193

How the `lpsched` Daemon Tracks the Status of Print Requests 194

Cleaning Out Log Files 194

▼ How to Change Frequency of Printer Request Log Rotation 194

How Local Printing Works 195

How Remote Printing Works 196

9. Working With Remote Systems Topics 201

10. Working With Remote Systems (Tasks) 203

What is a Remote System? 203

Logging In to a Remote System (`rlogin`) 204

Authentication for Remote Logins (`rlogin`) 204

Linking Remote Logins 207

Direct vs. Indirect Remote Logins 208

What Happens After You Log In Remotely 210

▼ How to Search for and Remove `.rhosts` Files 211

▼ How to Find Out If a Remote System Is Operating 212

▼ How to Find Who Is Logged In to a Remote System 212

▼ How to Log In to a Remote System (`rlogin`) 213

▼ How to Log Out From a Remote System (`exit`) 214

Logging In to a Remote System (`ftp`) 215

Authentication for Remote Logins (`ftp`) 215

Essential `ftp` Commands 215

▼ How to Open an `ftp` Connection to a Remote System 216

▼ How to Close an `ftp` Connection to a Remote System 217

▼ How to Copy Files From a Remote System (`ftp`) 218

▼ How to Copy Files to a Remote System (`ftp`) 220

Remote Copying With `rcp` 222

Security Considerations for Copy Operations 223

Specifying Source and Target 223

▼ How to Copy Files Between a Local and a Remote System (`rcp`) 225

11. Managing Terminals and Modems Topics 231

12. Managing Terminals and Modems (Overview) 233

Terminals, Modems, Ports, and Services 233

Terminals 234

Modems 234

Ports 234

Services 235

Port Monitors 235

Tools for Managing Terminals and Modems 236

Admintool 237

Service Access Facility 238

13. Setting Up Terminals and Modems (Tasks) 239

Setting Up Terminals and Modems 239

Setting Up Terminals 242

Setting Up Modems 243

▼ How to Start Admintool 245

▼ How to Set Up a Terminal 245

▼ How to Set Up a Modem 247

▼ How to Set Up a Modem for Use With UUCP 249

▼ How to Initialize a Port 250

▼ How to Disable a Port 251

▼ How to Remove a Port Service 252

Troubleshooting Terminal and Modem Problems 253

14. Setting Up Terminals and Modems With the Service Access Facility (Tasks) 255

Using the Service Access Facility 255

Overall Administration: `sacadm` Command 257

Service Access Controller: SAC Program 257

SAC Initialization Process 257

Port Monitor Service Administrator: `pmadm` Command 258

A Port Monitor at Work: `ttymon` 258

Port Initialization Process 259

Bidirectional Service 260

Port Monitors: TTY Monitor and Network Listener 260

TTY Port Monitor: `ttymon` 260

`ttymon` and the Console Port 261

Special `ttymon`-Specific Administrative Command: `ttyadm` 262

Network Listener Service: `listen` 262

Special `listen`-Specific Administrative Command: `nlsadmin` 262

Administering `ttymon` Port Monitors 263

▼ How to Add a `ttymon` Port Monitor 263

▼ How to View `ttymon` Port Monitor Status 263

Example—Viewing `ttymon` Port Monitor Status 264

▼ How to Stop a `ttymon` Port Monitor 265

▼ How to Start a `ttymon` Port Monitor 265

▼ How to Disable a `ttymon` Port Monitor 265

▼ How to Enable a `ttymon` Port Monitor 266

▼ How to Remove a `ttymon` Port Monitor 266

Administering `ttymon` Services 267

▼ How to Add a Service 267

▼ How to View the Status of a TTY Port Service 268

Example—Viewing the Status of a TTY Port Monitor Service 269

▼ How to Enable a Port Monitor Service 271

▼ How to Disable a Port Monitor Service 271

Reference Material for Service Access Facility Administration 272

Files Associated With SAF 272

The `/etc/saf/_sactab` File 272

The `/etc/saf/`*pmtab*`/_pmtab` File 273

Service States 274

Port Monitor States 275

Port States 276

15. Managing System Security Topics 277

16. Managing System Security (Overview) 279

What's New in Solaris System Security? 279

New Default Ownerships and Permissions on System Files and Directories 279

Role-Based Access Control 280

Sun Enterprise Authentication Mechanism (SEAM) or Kerberos V5 Client Support 280

Where to Find System Security Tasks 281

Controlling Access to a Computer System 281

Maintaining Physical Site Security 282

Maintaining Login and Access Control 282

Restricting Access to Data in Files 282

Maintaining Network Control 282

Monitoring System Usage 283

Setting the Correct Path 283

Securing Files 283

Installing a Firewall 284

Reporting Security Problems 284

File Security 284

File Administration Commands 284

File Encryption 285

Access Control Lists (ACLs) 285

System Security 286

Login Access Restrictions 286

Special Logins 287

Managing Password Information 288

Using the Restricted Shell 288

Tracking Superuser (Root) Login 289

Network Security 289

Firewall Systems 290

Authentication and Authorization 291

Sharing Files 293

Restricting Superuser (Root) Access 293

Using Privileged Ports 293

Using Automated Security Enhancement Tool (ASET) 294

17. Securing Files (Tasks) 295

File Security Features 295

User Classes 296

File Permissions 296

Directory Permissions 296

Special File Permissions (`setuid`, `setgid` and Sticky Bit) 297

Default `umask` 299

Displaying File Information 299

▼ How to Display File Information 299

Changing File Ownership 301

▼ How to Change the Owner of a File 301

▼ How to Change Group Ownership of a File 302

Changing File Permissions 303

▼ How to Change Permissions in Absolute Mode 306

▼ How to Change Special Permissions in Absolute Mode 307

▼ How to Change Permissions in Symbolic Mode 308

Searching for Special Permissions 309

▼ How to Find Files With `setuid` Permissions 309

Executable Stacks and Security 311

▼ How to Disable Programs From Using Executable Stacks 311

▼ How to Disable Executable Stack Message Logging 312

Using Access Control Lists (ACLs) 312

ACL Entries for Files 313

ACL Entries for Directories 314

▼ How to Set an ACL on a File 315

▼ How to Copy an ACL 317

▼ How to Check If a File Has an ACL 317

▼ How to Modify ACL Entries on a File 318

▼ How to Delete ACL Entries From a File 319

▼ How to Display ACL Entries for a File 320

18. Securing Systems (Tasks) 323

Displaying Security Information 323

▼ How to Display a User's Login Status 323

▼ How to Display Users Without Passwords 325

Temporarily Disabling User Logins 325

▼ How to Temporarily Disable User Logins 326

Saving Failed Login Attempts 326

▼ How to Save Failed Login Attempts 327

Password Protection Using Dial-up Passwords 327

▼ How to Create a Dial-up Password 330

▼ How to Temporarily Disable Dial-up Logins 332

Restricting Superuser (root) Access on the Console 332

▼ How to Restrict Superuser (root) Login to the Console 332

Monitoring Who Is Using the su Command 333

▼ How to Monitor Who Is Using the su Command 333

▼ How to Display Superuser (root) Access Attempts to the Console 334

Modifying a System's Abort Sequence 334

▼ How to Disable or Enable a System's Abort Sequence 334

19. Role-Based Access Control 335

Overview of Role-Based Access Control 335

Extended User Attributes Database (user_attr) 337

Authorizations 339

Execution Profiles 341

Execution Attributes 343

▼ How to Assume Role-Based Access Control 346

Tools for Managing Role-Based Access Control 347

20. Using Authentication Services (Tasks) 349

Overview of Secure RPC 349

NFS Services and Secure RPC 350

DES Encryption 350

Kerberos Authentication 350

Diffie-Hellman Authentication 351

Administering Diffie-Hellman Authentication 354

▼ How to Restart the Keyserver 354

▼ How to Set Up NIS+ Credentials for Diffie-Hellman Authentication 355

▼ How to Set Up NIS Credentials With Diffie-Hellman Authentication 357

▼ How to Share and Mount Files With Diffie-Hellman Authentication 358

Introduction to PAM 359

Benefits of Using PAM 359

Overview of PAM 360

PAM Module Types 360

Stacking Feature 360

Password-Mapping Feature 361

PAM Functionality 361

PAM Library 362

PAM Modules 362

PAM Configuration File 363

Configuring PAM 369

Planning for PAM 369

▼ How to Add a PAM Module 370

▼ How to Prevent Unauthorized Access From Remote Systems With PAM 370

▼ How to Initiate PAM Error Reporting 370

21. SEAM Overview 373

What Is SEAM? 373

SEAM Terminology 374

Kerberos-Specific Terminology 374

Authentication-Specific Terminology 375

SEAM Components 376

How SEAM Works 377

Principals 377

Realms 378

Security Services 380

22. Configuring SEAM 381

SEAM Administration Task Map 381

Configuring SEAM Clients 382

▼ How to Configure a SEAM Client 382

▼ How to Finish the Configuration of a SEAM Client 385

Configuring SEAM NFS Servers Task Map 385

▼ How to Configure SEAM NFS Servers 386

▼ How to Change the Back-end Mechanism for the `gsscred` Table 387

▼ How to Create a Credential Table 387

▼ How to Add a Single Entry to the Credential Table 388

▼ How to Set Up a Secure NFS Environment With Multiple Kerberos Security Modes 389

Synchronizing Clocks Between KDCs and SEAM Clients 391

SEAM Client Error Messages 392

23. SEAM Reference 393

Ticket Management 393

Do You Need to Worry About Tickets? 393

▼ How to Create a Ticket 394

▼ How to View Tickets 395

▼ How to Destroy Tickets 396

Password Management 397

Advice on Choosing a Password 397

Changing Your Password 398

SEAM Files 401

PAM Configuration File 402

SEAM Commands 403

Changes to the `share` Command 403

SEAM Daemons 404

Ticket Reference 404

Types of Tickets 404

How the Authentication System Works 408

Gaining Access to a Service Using SEAM 409

Obtaining a Credential for the Ticket-Granting Service 409

Obtaining a Credential for a Server 410

Obtaining Access to a Specific Service 411

Using the `gsscred` Table 412

Which Mechanism to Select for the `gsscred` Table 412

24. Using Automated Security Enhancement Tool (Tasks) 415

Automated Security Enhancement Tool (ASET) 415

ASET Security Levels 416

ASET Tasks 417

ASET Execution Log 420

ASET Reports 421

ASET Master Files 424

ASET Environment File (`asetenv`) 425

Configuring ASET 425

Restoring System Files Modified by ASET 428

Network Operation Using the NFS System 428

ASET Environment Variables 429

ASET File Examples 433

Running ASET 435

▼ How to Run ASET Interactively 435

▼ How to Run ASET Periodically 436

▼ How to Stop Running ASET Periodically 437

▼ How to Collect ASET Reports on a Server 437

Troubleshooting ASET Problems 439

ASET Error Messages 439

25. Managing System Resources Topics 445

26. Managing System Resources (Overview) 447

Where to Find System Resource Tasks 447

What's New in Managing System Resources? 448

Displaying and Changing System Information 448

What Are Quotas? 448

Executing Routine Tasks Automatically 449

Scheduling Repetitive Jobs: `crontab` 449

Scheduling a Single Job: `at` 450

What is System Accounting? 450

Accounting Components 451

How Accounting Works 451

27. Examining and Changing System Information (Tasks) 453

Using Commands to Display System Information 454

▼ How to Determine Whether a System Can Run the 64–bit Solaris Operating Environment 454

▼ How to Determine Whether a System Has 64-bit Solaris Capabilities Enabled 455

▼ How to Display System and Software Release Information 456

▼ How to Display General System Information (`uname`) 457

▼ How to Display a System's Host ID Number 457

▼ How to Display a System's Installed Memory 458

▼ How to Display the Date and Time 458

Using Commands to Change System Information 459

Using Network Time Protocol (NTP) in Your Network 459

▼ How to Set Up an NTP Server 460

▼ How to Set Up an NTP Client 460

▼ How to Synchronize Date and Time From Another System 461

▼ How to Set a System's Date and Time Manually 461

▼ How to Set Up a Message of the Day 462

▼ How to Set the Number of Processes per User 463

▼ How to Increase Shared Memory Segments 464

28. Managing Disk Use (Tasks) 467

Displaying Blocks and Files Used 467

▼ How to Display Information About Blocks, Files, and Disk Space 467

Checking the Size of Files 470

▼ How to Display the Size of Files 470

▼ How to Find Large Files 471

▼ How to Find Files That Exceed a Given Size Limit 472

Checking the Size of Directories 473

▼ How to Display the Size of Directories, Subdirectories, and Files 473

▼ How to Display the User Allocation of Local UFS File Systems 474

Finding and Removing Old and Inactive Files 475

▼ How to List the Newest Files 476

▼ How to Find and Remove Old or Inactive Files 476

▼ How to Clear Out Temporary Directories 478

▼ How to Find and Delete `core` Files 478

▼ How to Delete Crash Dump Files 479

29. Managing Quotas (Tasks) 481

Using Quotas 481

Soft Limits and Hard Limits 482

Difference Between Disk Block and File Limits 482

Setting Up Quotas 483

Guidelines for Setting Up Quotas 484

Setting Up Quotas Task Map 484

▼ How to Configure File Systems for Quotas 485

▼ How to Set Up Quotas for a User 486

▼ How to Set Up Quotas for Multiple Users 487

▼ How to Check Quota Consistency 487

▼ How to Turn Quotas On 488

Checking Quotas 489

▼ How to Check for Exceeded Quotas 489

▼ How to Check Quotas on a File System 490

Changing and Removing Quotas 492

▼ How to Change the Soft Time Limit Default 492

▼ How to Change Quotas for a User 493

▼ How to Disable Quotas for a User 494

▼ How to Turn Quotas Off 495

30. Scheduling System Events (Tasks) 497

Commands for Scheduling System Events 497

Scheduling a Repetitive System Event (`cron`) 498

Inside a `crontab` File 498

How the `cron` Daemon Handles Scheduling 499

Syntax of `crontab` File Entries 500

Creating and Editing `crontab` Files 501

▼ How to Create or Edit a `crontab` File 501

▼ How to Verify a `crontab` File 502

Displaying `crontab` Files 503

▼ How to Display a `crontab` File 503

Removing `crontab` Files 504

▼ How to Remove a `crontab` File 504

Controlling Access to `crontab` 505

▼ How to Deny `crontab` Access 506

▼ How to Limit `crontab` Access to Specified Users 507

▼ How to Verify Limited `crontab` Accesss 508

Scheduling a Single System Event (at) 508

at Command Description 509

▼ How to Create an at Job 509

▼ How to Display the at Queue 511

▼ How to Verify an at Job 511

▼ How to Display at Jobs 511

▼ How to Remove at Jobs 512

Controlling Access to at 513

▼ How to Deny at Access 513

▼ How to Verify at Access Is Denied 514

31. Managing System Accounting (Tasks) 515

Setting Up System Accounting 515

▼ How to Set Up System Accounting 516

Billing Users 518

▼ How to Bill Users 519

Maintaining Accounting Information 519

Fixing Corrupted Files and wtmpx Errors 519

▼ How to Fix a wtmpx File 520

Fixing tacct Errors 520

▼ How to Fix tacct Errors 520

Restarting runacct 521

▼ How to Restart runacct 521

Stopping and Disabling System Accounting 522

▼ How to Temporarily Stop System Accounting 522

▼ How to Permanently Disable System Accounting 523

32. System Accounting (Reference) 525

Daily Accounting 525

Connect Accounting 525

Process Accounting 526

Disk Accounting 526

Calculating User Fees 527

How Daily Accounting Works 527

Accounting Reports 529

Daily Accounting Reports 529

The `runacct` Program 538

Accounting Files 540

Files Produced by `runacct` 543

33. Managing System Performance Topics 545

34. System Performance (Overview) 547

What's New in Managing System Performance? 547

SPARC: `busstat` 547

`The cpustat and cputrack Commands` 548

`prstat` 548

Obsolete Interprocess Communication Parameters 549

Where to Find System Performance Tasks 549

System Performance and System Resources 549

Sources of Performance Tuning Information 550

Processes and System Performance 551

Commands for Managing Processes 552

About Monitoring Performance 553

Monitoring Tools 553

35. Managing Processes (Tasks) 555

Displaying Information About Processes 555

The `ps` Command 555

▼ How to List Processes 557

The `/proc` File System and Commands 558

Displaying Information About Processes (/proc Tools) 559

▼ How to Display Information About Processes 560

Controlling Processes (/proc Tools) 561

▼ How to Control Processes 563

Killing a Process (pkill) 564

▼ How to Kill a Process 565

Managing Process Class Information 565

Changing the Scheduling Priority of Processes With priocntl 566

▼ How to Display Basic Information About Process Classes 566

▼ How to Display the Global Priority of a Process 567

▼ How to Designate a Process Priority 567

▼ How to Change Scheduling Parameters of a Timeshare Process 568

▼ How to Change the Class of a Process 569

Changing the Priority of a Timesharing Process With nice 570

▼ How to Change the Priority of a Process 571

Process Troubleshooting 572

36. Monitoring Performance (Tasks) 573

Displaying Virtual Memory Statistics (vmstat) 574

▼ How to Display Virtual Memory Statistics (vmstat) 574

▼ How to Display System Event Information (vmstat -s) 576

▼ How to Display Swapping Statistics (vmstat -S) 577

▼ How to Display Cache Flushing Statistics (vmstat -c) 578

▼ How to Display Interrupts Per Device (vmstat -i) 578

Displaying Disk Utilization Information (iostat *n*) 579

▼ How to Display Disk Utilization Information (iostat) 579

▼ How to Display Extended Disk Statistics (iostat -xtc) 581

Displaying Disk Usage Statistics (df) 582

▼ How to Display File System Information (df) 582

Monitoring System Activities (sar) 583

▼ How to Check File Access (sar -a) 584

▼ How to Check Buffer Activity (sar -b) 584

▼ How to Check System Call Statistics (sar -c) 586

▼ How to Check Disk Activity (sar -d) 587

▼ How to Check Page-Out and Memory (sar -g) 588

▼ How to Check Kernel Memory Allocation (sar -k) 590

▼ How to Check Interprocess Communication (sar -m) 592

▼ How to Check Page-In Activity (sar -p) 593

▼ How to Check Queue Activity (sar -q) 594

▼ How to Check Unused Memory (sar -r) 595

▼ How to Check CPU Utilization (sar -u) 596

▼ How to Check System Table Status (sar -v) 598

▼ How to Check Swap Activity (sar -w) 599

▼ How to Check Terminal Activity (sar -y) 600

▼ How to Check Overall System Performance (sar -A) 602

Collecting System Activity Data Automatically (sar) 602

Collecting System Activity Data (sar) 603

▼ How to Set Up Automatic Data Collection 604

37. Troubleshooting Solaris Software Topics 607

38. Troubleshooting Software Problems (Overview) 609

Where to Find Software Troubleshooting Tasks 609

What's New in System Troubleshooting? 610

apptrace 610

Improved Core File Management 610

New Remote Console Messaging Features 611

Troubleshooting a System Crash 611

What to Do if the System Crashes 611

Gathering Troubleshooting Data 612

Troubleshooting a System Crash Checklist 613

Viewing System Messages 614

▼ How to View System Messages 615

Customizing System Message Logging 616

▼ How to Customize System Message Logging 618

Enabling Remote Console Messaging 619

Using Auxiliary Console Messaging During Run Level Transitions 619

Using the `consadm` Command During an Interactive Login Session 620

▼ How to Enable an Auxiliary (Remote) Console 621

▼ How to Display a List of Auxiliary Consoles 622

▼ How to Enable an Auxiliary (Remote) Console Across System Reboots 622

▼ How to Disable an Auxiliary (Remote) Console 623

39. Managing System Crash Information 625

System Crashes 625

System Crash Files and Core Files 626

Managing Core Files (`coreadm`) 626

Configurable Core File Paths 626

Expanded Core File Names 627

Setting the Core File Name Pattern 628

Enabling `setuid` Programs to Produce Core Files 628

▼ How to Display the Current Core Dump Configuration 629

▼ How to Set a Core File Name Pattern 629

▼ How to Display a Core File Name Pattern 630

▼ How to Enable a Per-Process Core File Path 630

▼ How to Enable a Global Core File Path 630

Troubleshooting Core File Problems 631

Managing System Crash Dump Information (`dumpadm`) 631

System Crash Dump Features 632

The dumpadm Command 632

Saving Crash Dumps 633

Managing System Crash Information Task Map 634

▼ How to Display the Current Crash Dump Configuration 634

▼ How to Modify a Crash Dump Configuration 635

▼ How to Examine a Crash Dump 637

▼ How to Recover From a Full Crash Dump Directory (Optional) 638

▼ How to Disable or Enable Saving Crash Dumps (Optional) 638

40. Troubleshooting Miscellaneous Software Problems 641

What to Do If Rebooting Fails 641

SPARC: Troubleshooting 64–bit Solaris Boot Problems 642

What to Do if a System Hangs 643

What to Do if a File System Fills Up 644

File System Fills Up Because a Large File or Directory Was Created 644

A TMPFS File System is Full Because the System Ran Out of Memory 644

What to Do if File ACLs Are Lost After Copy or Restore 645

Troubleshooting Backup Problems 645

The root (/) File System Fills Up After You Back Up a File System 645

Make Sure the Backup and Restore Commands Match 646

Check to Make Sure You Have the Right Current Directory 646

Use the Old restore Command to Restore Multivolume Diskette Backups 646

41. Troubleshooting File Access Problems 649

Solving Problems With Search Paths (Command not found) 649

▼ How to Diagnose and Correct Search Path Problems 650

Solving File Access Problems 652

Changing File and Group Ownerships 652

Recognizing Problems With Network Access 653

42. Troubleshooting Printing Problems 655

Tips on Troubleshooting Printing Problems 655

Troubleshooting No Output (Nothing Prints) 656

Troubleshooting Incorrect Output 658

Troubleshooting Hung LP Commands 659

Troubleshooting Idle (Hung) Printers 659

Troubleshooting Conflicting Status Messages 661

Troubleshooting Printing Problems 661

▼ How to Troubleshoot No Printer Output 661

▼ How to Troubleshoot Incorrect Output 675

▼ How to Unhang the LP Print Service 681

▼ How to Troubleshoot an Idle (Hung) Printer 682

▼ How to Resolve Conflicting Printer Status Messages 684

43. Troubleshooting File System Problems 687

`fsck` Error Messages 687

General `fsck` Error Messages 689

Initialization Phase `fsck` Messages 690

Phase 1: Check Blocks and Sizes Messages 694

Phase 1B: Rescan for More DUPS Messages 698

Phase 2: Check Path Names Messages 698

Phase 3: Check Connectivity Messages 707

Phase 4: Check Reference Counts Messages 709

Phase 5: Check Cylinder Groups Messages 713

Cleanup Phase Messages 714

44. Troubleshooting Software Administration Problems 717

What's New in Troubleshooting Software Administration Problems? 717

Specific Software Administration Errors 718

General Software Administration Problems 719

Index 721

About This Book

System Administration Guide, Volume 2 is part of a three-volume set that covers a significant part of the Solaris™ system administration information. It includes information for both SPARC™ based and IA based systems.

This book assumes that you have already installed the SunOS™ 5.8 operating system, and you have set up any networking software that you plan to use. The SunOS 5.8 operating system is part of the Solaris 8 product family, which also includes many features, including the Solaris Common Desktop Environment (CDE). The SunOS 5.8 operating system is compliant with AT&T's System V, Release 4 operating system.

For the Solaris 8 release, new features interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

Note - The Solaris operating environment runs on two types of hardware, or platforms—SPARC and IA. The Solaris operating environment runs on both 64–bit and 32–bit address spaces. The information in this document pertains to both platforms and address spaces unless called out in a special chapter, section, note, bullet, figure, table, example, or code example.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Solaris 8 release. To use this book, you should have 1-2 years of UNIX® system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Volumes Are Organized

Here is a list of the topics covered by the three volumes of the System Administration Guides.

System Administration Guide, Volume 1

- "Managing Users and Groups Topics" in *System Administration Guide, Volume 1*
- "Managing Server and Client Support Topics" in *System Administration Guide, Volume 1*
- "Shutting Down and Booting a System Topics" in *System Administration Guide, Volume 1*
- "Managing Removable Media Topics" in *System Administration Guide, Volume 1*
- "Managing Software Topics" in *System Administration Guide, Volume 1*
- "Managing Devices Topics" in *System Administration Guide, Volume 1*
- "Managing Disks Topics" in *System Administration Guide, Volume 1*
- "Managing File Systems Topics" in *System Administration Guide, Volume 1*
- "Backing Up and Restoring Data Topics" in *System Administration Guide, Volume 1*

System Administration Guide, Volume 2

- "Managing Printing Services Topics" in *System Administration Guide, Volume 2*
- "Working With Remote Systems Topics" in *System Administration Guide, Volume 2*
- "Managing Terminals and Modems Topics" in *System Administration Guide, Volume 2*
- "Managing System Security Topics" in *System Administration Guide, Volume 2*
- "Managing System Resources Topics" in *System Administration Guide, Volume 2*
- "Managing System Performance Topics" in *System Administration Guide, Volume 2*
- "Troubleshooting Solaris Software Topics" in *System Administration Guide, Volume 2*

System Administration Guide, Volume 3

- “Network Services Topics” in *System Administration Guide, Volume 3*
- “IP Address Management Topics” in *System Administration Guide, Volume 3*
- “Modem-Related Network Services Topics” in *System Administration Guide, Volume 3*
- “Accessing Remote File Systems Topics” in *System Administration Guide, Volume 3*
- “Mail Services Topics” in *System Administration Guide, Volume 3*
- “Monitoring Network Services Topics” in *System Administration Guide, Volume 3*

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at `http://www1.fatbrain.com/documentation/sun`.

Accessing Sun Documentation Online

The docs.sun.com℠ Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

What Typographic Conventions Mean

The following table describes the typographic conventions used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls –a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename.*
AaBbCc123	Book titles, new words or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide.* These are called *class* options. Do *not* save changes yet.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	#

General Conventions

Be aware of the following conventions used in this book.

- When following steps or using examples, be sure to type double-quotes ("), left single-quotes (‘), and right single-quotes (’) exactly as shown.
- The key referred to as Return is labeled Enter on some keyboards.
- It is assumed that the root path includes the `/sbin`, `/usr/sbin`, `/usr/bin`, and `/etc` directories, so the steps in this book show the commands in these directories without absolute path names. Steps that use commands in other, less common, directories show the absolute path in the example.
- The examples in this book are for a basic SunOS 5.8 software installation without the Binary Compatibility Package installed and without `/usr/ucb` in the path.

Caution - If `/usr/ucb` is included in a search path, it should always be at the end of the search path. Commands like `ps` or `df` are duplicated in `/usr/ucb` with different formats and options from the SunOS 5.8 commands.

CHAPTER 1

Managing Printing Services Topics

This section provides instructions for managing printing services in the Solaris environment. This section contains these chapters.

Chapter 2	Provides overview information for managing printing services on a network. This chapter provides information on print servers, print clients, and the LP print service.
Chapter 3	Provides overview information for planning printing services on a network, which includes information on allocating system resources and defining printers on a network.
Chapter 4	Provides step-by-step instructions for setting up a printer on a system and making it available to other systems on the network.
Chapter 5	Provides step-by-step instructions for administering printers, such as deleting printers, setting print policies, and managing print requests.
Chapter 6	Provides step-by-step instructions for setting up and maintaining character sets, print filters, forms, and fonts.
Chapter 7	Provides step-by-step instructions for customizing the LP print service, such as adjusting printer port characteristics or adding a `terminfo` entry for a unsupported printer.
Chapter 8	Provides background information on the LP print service.

CHAPTER 2

Print Management (Overview)

This chapter provides information about managing printers in the Solaris environment. This is a list of the overview information in this chapter.

- "What's New in Printing?" on page 39
- "Printing in the Solaris Operating Environment" on page 41
- "The LP Print Service" on page 43
- "The Solaris Print Client-Server Process" on page 45

For step-by-step instructions on print management tasks, see:

- Chapter 4
- Chapter 5
- Chapter 6
- Chapter 7

What's New in Printing?

This section describes new printing features in the Solaris 8 release.

Solaris Print Manager

Solaris Print Manager is a Java-based graphical user interface that enables you to manage local and remote printer configuration. This tool can be used in the following name service environments: NIS, NIS+, NIS+ with Federated Naming Service (xfn), and files. You must be logged in as superuser to use this tool.

Using Solaris Printer Manager is the preferred method for managing printer configuration information. It is preferred over Admintool: Printers because it centralizes printer information when used in conjunction with a name service. Using a name service for storing printer configuration information is desirable because it makes printer information available to all systems on the network, making printing administration easier.

In this release, you can manage printer configuration information in the NIS+ name service without the underlying xfn application layer with Solaris Printer Manager. This provides better performance when accessing printer configuration information. See "How to Convert Printer Configuration Information in NIS+ (+xfn) to NIS+ Format" on page 86 for information on converting NIS+ (xfn) printer information to NIS+ printer information.

Solaris Print Manager recognizes existing printer information on the printer servers, print clients, and in the name service databases. There are no conversion tasks required to use the new Solaris Print Manager as long as the print clients are running either the Solaris 2.6 release or compatible versions.

The Solaris Print Manager package is `SUNWppm`.

Print Naming Enhancement

This Solaris release supports the `printers` database in `/etc/nsswitch.conf`, the name service switch file. The `printers` database provides centralized printer configuration information to print clients on the network.

By including the `printers` database and corresponding sources of information in the name service switch file, print clients automatically have access to printer configuration information without having to add it to their own systems.

The default `printers` entry in the `/etc/nsswitch.conf` file for files, NIS, and NIS+ environments are described in the following table. The `nisplus` keyword represents the `printers.org_dir` table. The `xfn` keyword represents the FNS printer contexts.

If Your Name Service Is ...	**The Default `printers` Entry Is ...**
`files`	`printers: user files`
`nis`	`printers: user files nis`
`nis+`	`printers: user nisplus files xfn`

For example, if your name service is NIS, printer configuration information on print clients is looked up in the following sources in this order:

- `user` - Represents the user's `$HOME/.printers` file
- `files` - Represents the `/etc/printers.conf` file
- `nis` - Represents the `printers.conf.byname` table

See `nsswitch.conf`(4) and *Solaris Naming Administration Guide* for more information.

Enabling or Disabling Banner Page Printing

You can use the `lpadmin` command to enable or disable system-wide banner page printing in this Solaris release.

You can specify whether a banner page is always printed, never printed, or whether banner page printing is optional with the `lpadmin`'s new `–banner` option arguments (`always`, `never`, and `optional`). If banner page printing is set to optional, the banner is printed by default, but users can disable banner page printing with the `lp -o nobanner` command.

See "How to Make Banner Pages Optional" on page 97 and `lpadmin`(1M) for more information.

Printing in the Solaris Operating Environment

The Solaris printing software provides an environment for setting up and managing client access to printers on a network.

The Solaris printing software contains these components:

- Solaris Print Manager, a graphic user interface, provides the ability to manage printing configuration on a local system or in a name service.
- Admintool, a graphical user interface, manages printing on a local system.
- The LP print service commands, a command line interface used to set up and manage printers. They also provide functionality above and beyond the other print management tools.

Even if you do use Solaris Print Manager to set up printing, you will have to use some of the LP commands to completely manage printing in the Solaris environment. See Chapter 5 for more information.

The Solaris print software limitations include:

- No support for print servers defined as `s5` (the System V print protocol) in previous Solaris releases.

- No print filtering on print clients.

Choosing a Method to Manage Printers

Adding printer information to a name service makes access to printers available to all systems on the network and generally makes printer administration easier because all the information about printers is centralized.

If You ...	To Centralize Printer Information, Then ...
Use a name service	Adding the printer to the NIS, NIS+, or NIS+ (xfn) database makes the printer available to all systems on the network.
Don't use a name service	Adding the printer adds the printer information to the printer server's configuration files only. Print clients will not know about the printer automatically. You will have to add the printer information to every print client that needs to use the printer.

The following table describes the major printer-related tasks and the tools available to perform the printing tasks.

TABLE 2–1 Solaris Printing Component Features

Component	Available In ...	Graphical User Interface?	Configures Network Printers?	Manages Print Clients and Servers?	Uses NIS, NIS+, or NIS+ (xfn)?
Solaris Print Manager	Solaris 8 and Solaris Easy Access Server 3.0	Yes	Yes	Yes	Yes
Admintool	Solaris 8 and compatible versions	Yes	No	Yes	No
LP commands	Solaris 8 and compatible versions	No	Yes	Yes	Yes

After using the table above to determine which printing tool is best for your network environment, see Chapter 4 for printer setup information.

Most printing configuration tasks can be accomplished with Solaris Print Manager. However, if you have special needs, such as writing interface scripts or adding your own filters, you can use the LP print service commands, which underlie Solaris Print Manager and Admintool directly. Performing printing administration tasks with LP commands are described in Chapter 5.

The LP Print Service

The *LP print service* is a set of software utilities that allows users to print files while they continue to work. Originally, the print service was called the LP spooler. (LP stood for line printer, but its meaning now includes many other types of printers, such as laser printers. Spool is an acronym for system peripheral operation off-line.)

The print service consists of the LP print service software and spooler, which includes Solaris Print Manager; any print filters you might provide; and the hardware (the printer, system, and network connections).

See Chapter 8 for background information about the LP print service.

Other LP print service topics covered in this part and their chapter references are described below.

Managing Network Printers

A *network printer* is a hardware device that is connected directly to the network. It transfers data directly over the network to the output device. The printer or network connection hardware has its own system name and IP address.

Network printers often have software support provided by the printer vendor. If your printer has printer vendor supplied software it is strongly advised that the printer vendor software be utilized. If the network printer vendor does not provide software support, Sun supplied software is available. This software provides generic support for network attached printers but is not capable of providing full access to all possible printer capabilities.

See Chapter 4 for step-by-step instructions on setting up a network printer.

Administering Printers

After you set up print servers and print clients, there are a number of administration tasks you might need to perform frequently:

- Deleting a printer and remote printer access
- Checking the status of printers
- Restarting the print scheduler

See Chapter 5 for step-by-step instructions on how to perform the printer administration tasks.

Setting Definitions for Printers

Establishing definitions for the printers on your network is an ongoing task that lets you provide a more effective print environment for users. For example, you can assign printer descriptions for all your site's printers to help users find where a printer is located, or you can define a class of printers to provide the fastest turnaround for print requests.

See Chapter 3 for information on setting up printer definitions.

Administering Character Sets, Filters, Forms, and Fonts

Depending on your site's requirements and the types of printers you have on the network, you might have to set up and administer printer-specific features of the LP print service. For example, you can assign different print wheels, filters, and forms to different printers. See Chapter 6 for background information and step-by-step instructions on how to set up and administer character sets, print filters, forms, and fonts.

Customizing the LP Print Service

Although the LP print service is designed to be flexible enough to handle most printers and printing needs, it does not handle every possible situation. You might have a printing request that is not accommodated by the standard features of the LP print service. Or you can have a printer that does not quite fit into the way the LP print service handles printers.

You can customize the LP print service in the following ways:

- Adjust the printer port characteristics
- Adjust the `terminfo` database

- Customize the printer interface program
- Create a print filter
- Define a form

See Chapter 7 for detailed descriptions and step-by-step instructions to customize the LP print service.

The Solaris Print Client-Server Process

This section provides an overview of how Solaris printing works.

The Print Client Process

The figure below illustrates the path of a print request from the time the user initiates the request until it is printed.

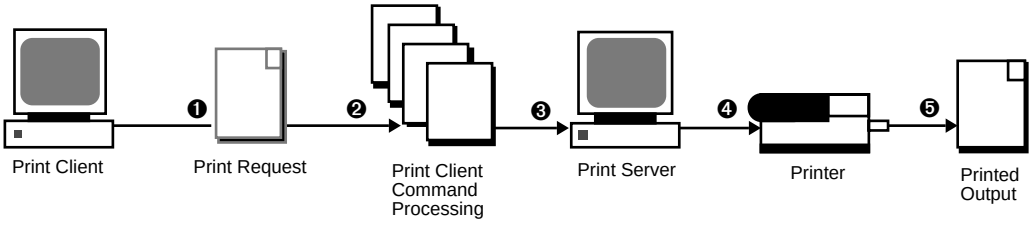


Figure 2–1 Overview of the Print Client Process

1. A user submits a print request from a print client.
2. The print command checks a hierarchy of print configuration resources to determine where to send the print request.
3. The print command sends the print request directly to the appropriate print server. A print server can be any server that accepts BSD printing protocol, including SVR4 (LP) print servers and BSD print servers (such as the SunOS 4.1 BSD print server).
4. The print server sends the print request to the appropriate printer.
5. The print request is printed.

Using Print Clients

This section of the overview focuses on the *print client*, a system that can send print requests to a print server, and print commands, which enable the print client to initiate print requests.

The figure below highlights the part of the print process in which the user submits a print request from a print client.

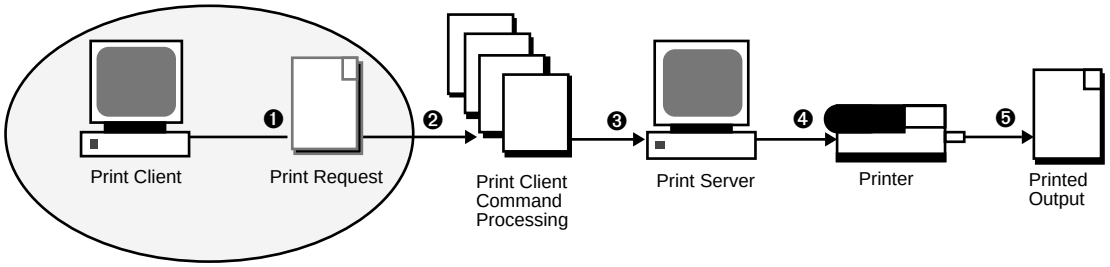


Figure 2–2 The User Submits a Print Request from a Print Client

What Is a Print Client?

A system becomes a print client when you install the Solaris print software and enable access to remote printers on the system.

The Solaris print software checks the following resources to locate printers and printer configuration information:

- The command-line interface using the `lp -d` command (atomic or POSIX format)
- A user's `LPDEST` or `PRINTER` variables
- The `_default` variable in the sources configured for the `printers` database in the `/etc/nsswitch.conf` file
- The `$HOME/.printers` file for users
- The local `/etc/printers.conf` file for the NIS name service
- The `printers.org_dir` table for the NIS+ name service
- FNS printing contexts for the NIS+ (xfn) name service

The print client sends its requests to the print server's queue; the client does not have a local queue. The client writes the print request to a temporary spooling area only if the print server is not available or if an error occurs. This streamlined path to the server decreases the print client's use of resources, reduces the chances for printing problems, and improves performance.

Printer Configuration Resources

This section describes the resources that the print software use to locate printer names and printer configuration information.

The print software can use a name service, which is a network (shared) resource for storing printer configuration information for all printers on the network. The name

service (NIS, NIS+, or NIS+ (xfn)) simplifies printer configuration maintenance: When you add a printer in the name service, all print clients on the network can access it.

The figure below highlights the part of the print process in which the print software checks a hierarchy of printer configuration resources to determine where to send the print request.

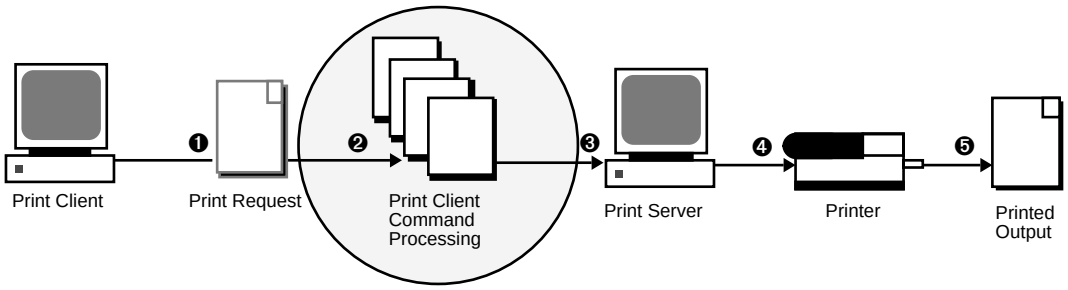


Figure 2–3 The Print Client Checks Resources to Locate Printers

How the Print Software Locates Printers

As shown in the figure below, the print software use more options to locate printers and printer configuration information.

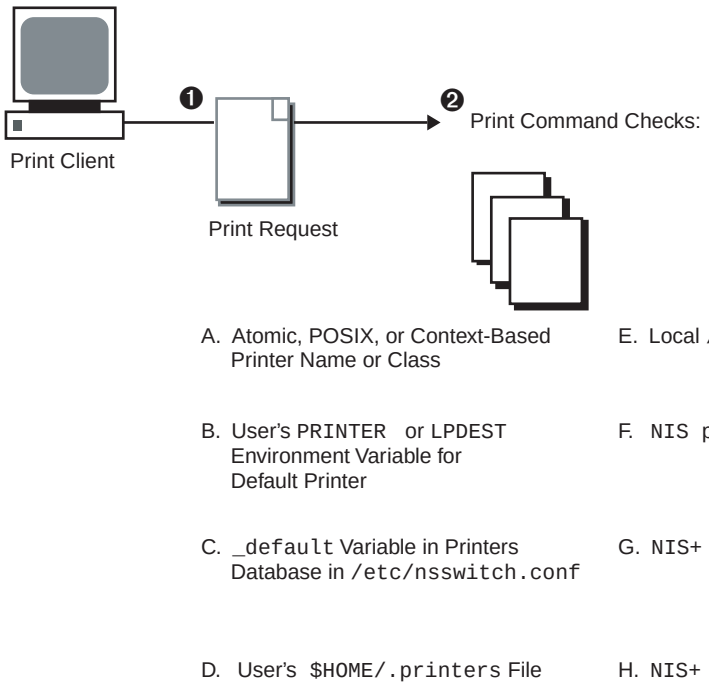


Figure 2–4 How the Print Client Software Locates Printers

1. A user submits a print request from a print client by using the `lp` or `lpr` command. The user can specify a destination printer name or class in any of three styles:

   - Atomic style, which is the print command and option followed by the printer name or class, as shown in this example.

     ```
     % lp -d neptune filename
     ```

   - POSIX style, which is the print command and option followed by *server:printer*, as shown in the following example.

     ```
     % lpr -P galaxy:neptune filename
     ```

   - Context-based style, as defined in the *Federated Naming Service Programming Guide*, shown in this example.

     ```
     % lpr -d thisdept/service/printer/printer-name filename
     ```

2. The print command locates a printer and printer configuration information as follows:

   - It checks to see if the user specified a destination printer name or printer class in one of the three valid styles.

- If the user didn't specify a printer name or class in a valid style, the command checks the user's `PRINTER` or `LPDEST` environment variable for a default printer name.
- If neither environment variable for the default printer is defined, the command checks the sources configured for the `printers` database in the `/etc/nsswitch.conf` file.

Using Print Servers

This section of the overview focuses on the print server, a system that has a local printer connected to it and makes the printer available to other systems on the network. The figure below highlights the part of the print process in which the print server sends the print request to the printer.

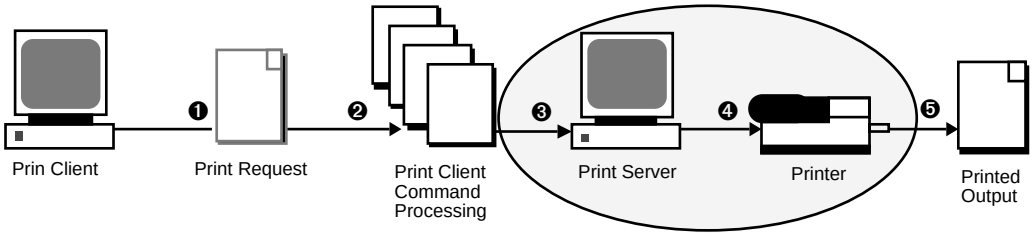


Figure 2–5 The Print Server Sends a Print Request to the Printer

The BSD Printing Protocol

The print commands use the BSD printing protocol. One of the big advantages of this protocol is that it can communicate with a variety of print servers:

- SunOS 4.1 BSD (LPD) print servers
- SunOS 5.8 and compatible SVR4 (LP) print servers
- Any other print server or printer that accepts the BSD printing protocol

The BSD printing protocol is an industry standard. It is widely used and it provides compatibility between different types of systems from various manufacturers. Sun has chosen to support the BSD printing protocol to provide interoperability in the future.

Where to Go From Here

Go to Chapter 4 for step-by-step instructions on setting up new printers with Solaris Print Manager. If you need printer planning information, see Chapter 3.

CHAPTER 3

Planning Printers on Your Network (Overview)

The goal of setting up printers on a network is to give users access to one or more printers. This section provides information about distributing printers across your network to gain the best efficiency and about planning for printer setup.

- "Distributing Printers on the Network" on page 51
- "Assigning Print Servers and Print Clients" on page 52
- "Print Server Requirements and Recommendations" on page 53

For step-by-step instructions on print management tasks, see:

- Chapter 4
- Chapter 5
- Chapter 6
- Chapter 7

Distributing Printers on the Network

As an administrator, you must determine whether each printer would be best used if it is dedicated to one system or available to many systems. In a network environment, it usually works best to distribute your printers on several print servers. The advantage of setting up several print servers is that when one print server has a problem, you can route print requests to other print servers.

If you use a centralized print configuration, you can still connect printers to users' systems for convenience or for improved response. A printer that is connected to a user's system is still available to other systems on the network.

The figure below shows an example of how you can have a centralized print configuration and still connect printers to users' systems.

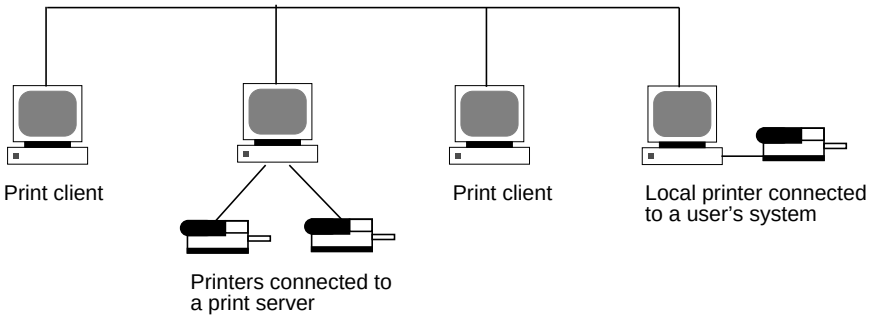


Figure 3–1 How to Distribute Printers on a Network

Assigning Print Servers and Print Clients

You must decide which systems will have local printers physically attached to them, and which will systems use printers on other systems. A system that has a local printer attached to it and makes the printer available to other systems on the network is called a *print server*. A system that sends its print requests to a print server is called a *print client*.

The LP print service software provides printing services in the Solaris environment. Besides physically connecting a printer to a system, you must define the printer characteristics to the LP print service and make the system a print server. Once you have print servers set up, you can set up other systems as print clients.

Print servers and print clients can run different versions of the SunOS operating system. Systems running the SunOS 5.8 release and compatible versions can print to existing print servers running the SunOS 4.1 operating system, and systems running the SunOS 4.1 operating system can print to print servers running the SunOS 5.8 release and compatible versions.

Note - SunOS 5.8 is part of the Solaris 8 operating environment.

The figure below shows example print configurations on a network with systems running the SunOS 5.8 and SunOS 4.1 operating systems.

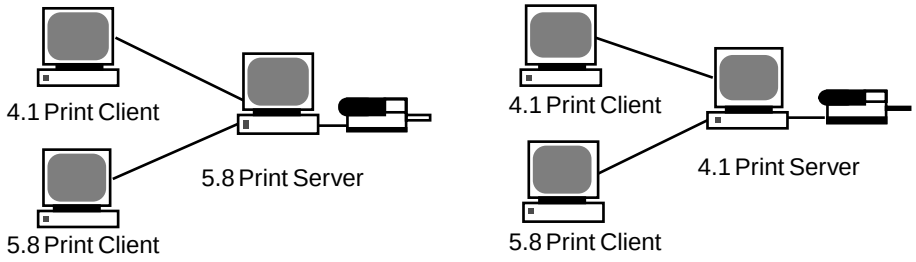


Figure 3–2 Example Print Configurations on SunOS 5.8 and SunOS 4.1 Systems

Print Server Requirements and Recommendations

You can attach a printer to a standalone system or to any system on the network. Any networked system with a printer can be a print server, as long as the system has adequate resources to manage the printing load.

Spooling Space

Spooling space is the amount of disk space that is used to store and process requests in the print queue. Spooling space is the single most important factor to consider when deciding which systems to designate as print servers. When users submit files for printing, the files are stored in the `/var/spool/lp` directory until they have been printed. The size of the `/var` directory depends on the size of the disk and how the disk is partitioned. Spooling space can be allocated in the `/var` directory on the print server hard disk, or mounted from a file server and accessed over the network.

Note - If `/var` is not created as a separate file system, the `/var` directory uses space in the root (`/`) file system, which is likely to be insufficient.

Disk Space

When evaluating systems as possible print servers, consider their available disk space. A large spool directory can consume 600 Mbytes of disk space. Look at the size and division of disk space on systems that can be designated as print servers.

Also, carefully evaluate the printing needs and use patterns of print client systems. If users in a small group typically print only short email messages—simple ASCII files

without sophisticated formatting requirements—a print server with 20 to 25 Mbytes of disk space allocated to `/var` is probably sufficient. If, however, many print client users are printing large documents or bit-mapped or raster images, they will likely fill up the spooling space quite frequently. When users cannot queue their jobs for printing, work flow is interrupted. Requests for more spooling space can force you to either add disk space for spooling or designate a different system as the print server.

If the print server has a `/var` directory that resides in a small partition, and if a large amount of disk space is available elsewhere, you can use that space as spooling space by mounting it on the `/var` directory on the print server. See "Mounting and Unmounting File Systems (Tasks)" in *System Administration Guide, Volume 1* for information about mounting file systems and editing the `vfstab` file.

Memory

The Solaris environment requires a minimum of 64 Mbytes of memory to run. A print server does not require additional memory. However, you might find that more memory improves performance in filtering print requests.

Swap Space

The swap space allocation on the print server should be sufficient to handle LP print service requirements. See "Configuring Additional Swap Space (Tasks)" in *System Administration Guide, Volume 1* for information about how to increase swap space.

Hard Disk

For optimal performance, the print server should have a hard disk and a local `/var` directory. You should mount spooling space for a print server on a local hard disk. If a print server has its own hard disk and a local `/var` directory, printing is much faster, and you can more accurately predict the time needed to process print requests.

Planning for Printer Setup

This section provides an overview of planning for printing in the Solaris environment that includes:

- Setting definitions for printers such a printer name, printer description, printer port

- Selecting a printer type and file content type
- Setting up fault notification and default printer destination
- Determining whether you want to print banner pages or limit user access to a printer
- Setting up printer classes and fault recovery

Setting Definitions for Printers

Establishing definitions for the printers on your network is an ongoing task that lets you provide a more effective print environment for users. For example, you can assign parameters for all your site's printers to help users find where a printer is located, or you can define a class of printers to provide the fastest turnaround for print requests.

The `lpadmin` command lets you set all of the print definitions, while Solaris Print Manager lets you set only some of them when you install or modify a printer. The table below lists the print definitions and shows whether you can assign the definition with Solaris Print Manager.

TABLE 3–1 Print Definitions Set With Solaris Print Manager

Print Definition	Can You Set It With Solaris Print Manager?
Printer name	Yes
Printer description	Yes
Printer port	Yes
Printer type	Yes
File contents	Yes, but with less functionality than the `lpadmin` command
Fault notification	Yes, but with less functionality than the `lpadmin` command
Default printer destination	Yes
Printing banner pages	Yes, but with less functionality than the `lpadmin` command
Limiting user access to a printer	Yes, but with less functionality than the `lpadmin` command

TABLE 3–1 Print Definitions Set With Solaris Print Manager *(continued)*

Print Definition	Can You Set It With Solaris Print Manager?
Printer class	No
Fault recovery	No

Printer Name

When adding a printer to a system, you specify a *printer name* for the printer. A printer name must be:

- Unique among all printers within the bounds of an administrative domain
- A maximum of 14 alphanumeric characters, which can include dashes and underscores
- Easy to remember and can identify the type of printer, its location, or the print server name

Establish a naming convention that works for your site. For example, if you have different types of printers on the network, including the printer type as part of the printer name can help users choose an appropriate printer. For instance, you could identify PostScript™ printers with the letters PS. If, however, all of the printers at your site are PostScript printers, you would not need to include the initials PS as part of the printer name.

Printer Description

You can assign a description to a printer by using the `lpadmin -D` command or Solaris Print Manager. The printer's description should contain information to help users identify the printer. You might include the room number where the printer is located, the type of printer, the manufacturer, or the name of the person to call if there are printing problems.

Users can look at a printer description by using the following command:

```
$ lpstat -D -p printer-name
```

Printer Port

When you install a printer or later change its setup, you can specify the device, or the *printer port*, to which the printer is connected, by using Solaris Print Manager or the `lpadmin -p` *printer-name* `-v` *device-name* command.

Most systems have two serial ports and a parallel port. Unless you add ports, you cannot directly connect more than two serial printers and a parallel printer to one system.

With Solaris Print Manager, you can select either `/dev/term/a` or `/dev/term/b`, or choose `Other` and specify any port name that the print server recognizes. These options give you as much flexibility as the `lpadmin` command.

The LP print service initializes the printer port using the settings from the standard printer interface program. See "Managing Print Filters" on page 130 for more information about printer interface programs. If you have a parallel printer or a serial printer for which the default settings do not work, see "Adjusting Printer Port Characteristics" on page 151 for information about customizing the port settings.

IA platform only - If you use multiple ports on an IA based system, only the first port is enabled by default. The second and any subsequent ports are disabled by default. To use more than one port, you must manually edit the device driver port configuration file for each additional `asy` (serial) port or `lp` (parallel) port. The pathnames for the IA port configuration files are:

`/platform/i86pc/kernel/drv/asy.conf`

`/platform/i86pc/kernel/drv/lp.conf`

See the *Solaris 8 (Intel Platform Edition) Device Configuration Guide* for information about configuring serial and parallel ports on IA based systems.

Printer Type

The printer type is a generic name for a type of printer. It identifies the `terminfo` database entry that contains various control sequences for the printer. By convention, printer type is usually derived from the manufacturer's model name. For example, the printer type name for the DECwriter™ printer is `decwriter`. However, the common printer type `PS` does not follow this convention. `PS` is used as the printer type for many models of PostScript™ printers, such as Apple LaserWriter®I and Apple LaserWriterII printers.

You can specify the printer type by using the `lpadmin -T` command or Solaris Print Manager.

Solaris Print Manager lets you select a printer type from a menu or choose `Other` and specify any printer type in the `terminfo` database. This provides you as much capability as the `lpadmin` command.

Printer Names in the `terminfo` *Database*

Information about each printer type is stored in the `terminfo` database (`/usr/share/lib/terminfo`). This information includes the printer capabilities

and initialization control data. The printer you install must correspond to an entry in the `terminfo` database.

```
$ pwd
/usr/share/lib/terminfo
$ ls
1  3  5  7  9  B  H  P  a  c  e  g  i  k  m  o  q  s  u  w  y
2  4  6  8  A  G  M  S  b  d  f  h  j  l  n  p  r  t  v  x  z
$
```

Each subdirectory contains compiled database entries for terminals or printers. The entries are organized by the first letter of the printer or terminal type. For example, if you have an Epson® printer, look in `/usr/share/lib/terminfo/e` to find your particular model of Epson printer.

```
$ cd /usr/share/lib/terminfo/e
$ ls
emots           ep2500+high   ep48            ergo4000      exidy2500
env230          ep2500+low    epson2500       esprit
envision230     ep40          epson2500-80    ethernet
ep2500+basic    ep4000        epson2500-hi    ex3000
ep2500+color    ep4080        epson2500-hi80  exidy
$
```

The entries for Epson printers are included in the preceding example.

If you have a NEC® printer, look in the `/usr/share/lib/terminfo/n` directory for your NEC printer model.

```
$ cd /usr/share/lib/terminfo/n
$ ls
ncr7900         ncr7900iv      netronics      network        nuc
ncr7900-na      ncr7901        netty          netx           nucterm
ncr7900i        nec            netty-Tabs     newhp
ncr7900i-na     net            netty-vi       newhpkeyboard
$
```

The entry in this directory for NEC is included in the preceding example.

Selecting a Printer Type

For a local PostScript printer, use a printer type of either PostScript (PS) or Reverse PostScript (PSR). If your printer supports PostScript, choose PS or PSR even if the specific printer type is listed in the `terminfo` database.

If your PostScript printer prints pages face up, documents appear to be printed backwards—the first page is at the bottom of the stack and the last page is on the

top. If you specify the printer's type as PSR, the LP print service reverses the order of the pages before sending them to the printer; the last page is printed first, and the pages are stacked in forward order. However, the LP print service can reliably change the page order only for PostScript files that conform to the Adobe® Document Structuring Conventions in Appendix C of the *PostScript Language Reference Manual* (written by Adobe Systems Incorporated, and published by Addison-Wesley, 1990).

If a printer can emulate more than one kind of printer, you can assign it several types by using the `lpadmin -T` command. If you specify more than one printer type, the LP print service uses the type that is appropriate for each print request.

You might not find the printer type in the appropriate `terminfo` directory. The type of printer is not necessarily linked to the manufacturer's name on the printer. For example, for any type of PostScript printer, you can use the PS or PSR entry (found in the `/usr/share/lib/terminfo/P` directory) instead of an entry specific to manufacturer or product names.

If you have an unusual type of printer, you might need to try different entries before you can determine whether a particular `terminfo` entry works for your model of printer. If possible, find an entry in the `terminfo` database that works for your printer. It will be much easier than trying to create an entry. If you have to create your own entry, "Adding a `terminfo` Entry for an Unsupported Printer" on page 154 contains some useful tips.

Selecting a File Content Type

Print filters convert the content type of a file to a content type that is acceptable to the destination printer. The *file content type* tells the LP print service the type of file contents that can be printed directly, without filtering. To print without filtering, the necessary fonts must also be available in the printer. (You must set up and use filtering for other types of files.)

You can specify the file content type for a printer by using the `lpadmin -I` command or Solaris Print Manager. With Solaris Print Manager, you can select a file content type from a menu. Not all available file content types are listed on the menu. You must use the `lpadmin` command to specify file content types that are not included on the Solaris Print Manager menu.

Many printers can print two types of files directly:

- The same type as the printer type (for example, PS for a PostScript printer)
- The type `simple` (an ASCII text file)

When submitting a file for printing, the user can indicate the content type of the file (`lp -T` *content-type*). If no file content type is supplied when the request is submitted, the LP server looks at the first file in the request to determine the content type. If the file begins with `^D%!` or `%!`, the request is considered to contain PostScript™ data. Otherwise, the request is assumed to contain `simple` (ASCII) text.

The LP print service uses the file content type to determine which filters to use to convert the file contents into a type the printer can handle.

Solaris Print Manager provides a list of file content types from which you can choose when installing or modifying a local printer. The choices are translated to the names that the LP print service uses. The table below describes the file content types you can choose with Solaris Print Manager.

TABLE 3–2 Choosing File Content Type With Solaris Print Manager

File Contents Choice	LP Print Service Name	Description
PostScript	`postscript`	PostScript files do not require filtering.
ASCII	`simple`	ASCII files do not require filtering.
Both PostScript and ASCII	`simple`, `postscript`	PostScript files and ASCII files do not require filtering.
None	`""`	All files require filtering, except those matching the printer's type.
Any	`any`	No filtering required. If the printer cannot handle a file content type directly, the file will not be printed.

Choose the file content type that best matches the printer's capabilities. PostScript (which means filtering is not needed for PostScript files) is the default choice in Solaris Print Manager and is probably correct most of the time.

Frequently Used Printers

This section provides the printer type and file content type for the printers most commonly used with Solaris software. Although not shown, many of these printers can also directly print files with `simple` content type.

If you have a PostScript printer, use a printer type of PS or PSR and a content type of `postscript`. `PSR` reverses the pagination and prints the banner page last.

The table below lists additional non-PostScript printers and shows the printer type to use for configuring each printer. For all these printers, the file content type is `simple`.

Note - Sun Microsystems does not supply filtering software for the printers listed in the table below, among others. However, you can use unsupported printers if you supply filtering or if the printer can directly print the file content type. If you have questions about any printer for which Sun Microsystems does not supply filters, contact the printer manufacturer.

TABLE 3–3 Some Non-PostScript Printers for Which Sun Does Not Supply Filters

Printer	Printer Type
Daisy	`daisy`
Datagraphix	`datagraphix`
DEC LA100	`la100`
DEC LN03	`ln03`
DECwriter	`decwriter`
Diablo	`diablo`
	`diablo-m8`
Epson 2500 variations	`epson2500`
	`epson2500-80`
	`epson2500-hi`
	`epson2500-hi80`
Hewlett-Packard HPCL printer	`hplaser`
IBM Proprinter	`ibmproprinter`

If you want to set up a printer that is not in the `terminfo` database, see “How to Add a `terminfo` Entry for an Unsupported Printer” on page 157.

CHAPTER 4

Setting Up Printers (Tasks)

This chapter describes how to set up a printer and make it accessible to systems on the network with Solaris Print Manager, which was previously available in the Solaris™ Easy Access Server (SEAS) 3.0 release. This is a list of the step-by-step instructions in this chapter.

- "How to Start Solaris Print Manager" on page 66
- "How to Add a New Attached Printer With Solaris Print Manager" on page 70
- "How to Add Printer Access With Solaris Print Manager" on page 72
- "How to Set Up a `.printers` File" on page 73
- "How to Add a Network Printer Using Printer Vendor Supplied Tools" on page 79
- "How To Add A Network Printer Using LP Commands" on page 79

For overview information about printers, see Chapter 2.

Setting Up Printing

The table below provides an overview of the tasks necessary to set up print servers (Add a Printer) and print clients (Add Access to the Printer). A local printer is one which is physically cabled to the print server; a network printer is physically attached to the network. Adding access to a printer, or adding remote access, is the process of giving print clients (all those machines which are not the server) access to the printer.

Setting Up Printing Task Map

TABLE 4–1 Task Map: Setting Up Printing

Task	Description	For Instructions, Go To
1. Add New Attached Printer	*Using Solaris Print Manager* After physically attaching the printer to a system, make the printer available for printing.	"How to Add a New Attached Printer With Solaris Print Manager" on page 70
2. Add Access to a Printer	*Using Solaris Print Manager* Add printer access on the print client.	"How to Add Printer Access With Solaris Print Manager" on page 72
3. Set Up a `.printers` File	*Optional.* Using a `$HOME/.printers` file enables users to establish their own custom printer aliases.	"How to Set Up a `.printers` File" on page 73
4. Add a New Network Printer	*Using Printer Vendor Supplied Tools* After physically connecting the printer to the network, use vendor-supplied software to configure the network printer.	"How to Add a Network Printer Using Printer Vendor Supplied Tools" on page 79
	Using LP Commands After physically connecting the printer to the network, use Solaris software commands to configure the network printer.	"How To Add A Network Printer Using LP Commands" on page 79
5. Turn Off Banner Pages	*Optional.* You can turn off banner pages so they are never printed.	"How to Turn Off Banner Pages" on page 98
6. Set Up Fault Alerts	*Optional.* You can set up more specific fault alerts for the printer than Solaris Print Manager provides.	"How to Set Fault Alerts for a Printer" on page 101

TABLE 4–1 Task Map: Setting Up Printing *(continued)*

Task	Description	For Instructions, Go To
7. Set Up Fault Recovery	*Optional.* Solaris Print Manager does not enable you to set up how a printer should recover after it faults.	"How to Set Printer Fault Recovery" on page 104
8. Limit Access to the Printer	*Optional.* Solaris Print Manager enables you to set up an allow list, but if you want to limit a few users' access to the printer, you might want to set up a deny list.	"How to Limit User Access to a Printer" on page 106

Setting Up a Printer With Solaris Print Manager

The following table describes each printer attribute to help you determine the information needed to set up a printer with Solaris Print Manager.

Printer Attribute	Description	Example	Default Setting	Required or Optional?
Printer Name	Name of printer	`laser1`	N/A	Required to install an attached or network printer and to add access to a printer
Printer server	Name of printer server	`venus`	The local system	Required to install an attached or network printer and to add access to a printer
Description	User defined string	`laser printer near breakroom`	N/A	Optional
Printer Port	Device printer is attached to	`/dev/term/a`	`/dev/term/a`	Required to install an attached printer
Printer Type	Type of printer	`unknown`	`PostScript`	Required to install an attached or network printer
File Contents	Content to be printed	`any`	`PostScript`	Required to install an attached or network printer

Printer Attribute	Description	Example	Default Setting	Required or Optional?
Destination	Destination name for network printers	See "Selecting the Destination (or Network Printer Access) Name" on page 77 for examples	N/A	Required to install a network printer
Protocol	Protocol used to communicate with printer	`TCP`	`BSD`	Required to install a network printer
Fault Notification	How to notify user of errors	`Mail to superuser`	`Write to superuser`	Optional
Default Printer	Identifies the default printer	N/A	N/A	Optional
Always Print Banner	Print banner with print job?	N/A	Banner is printed	Optional
User Access List	List of users allowed to print	`rimmer,lister`	All users can print	Optional

Starting Solaris Print Manager

To use Solaris Print Manager to set up your printers, start Solaris Print Manager either by selecting Printer Administration from CDE Workspace menu or by starting it from the command line. See the following section for details.

▼ How to Start Solaris Print Manager

1. **Verify that the following prerequisites are met. To use Solaris Print Manager, you must:**
   - Have a bit-mapped display monitor. Solaris Print Manager can be used only on a system with a console that is a bit-mapped screen, such as a standard display monitor that comes with a Sun workstation.
   - Be running an X Window System, such as the CDE environment, or be using the remote display feature on a system running an `xhost` environment.
   - Be logged in as superuser on the printer server to install an attached or network printer, or on the print client to add access to a printer.

- Have the required access privileges for managing the NIS, NIS+, or NIS+ (xfn) database:
  - If your name service is NIS, you must have the root password for the NIS master.
  - If you name service is NIS+, you might need to do the following:
    1. Log in to the NIS+ master as superuser.
    2. Identify the group that owns the printers table:

```
# niscat -o printers.org_dir.domain_name.com
            .
            .
            .
         Group   : "admin.domain_name.com"
```

    3. If necessary, add the system that runs Solaris Print Manager to the NIS+ admin group authorized to update the `printers.org_dir.`<*domain*> file.

```
# nisgrpadm -a admin.domain_name.com host_name
```

    4. Log in to the system that runs Solaris Print Manager as superuser. Depending on your NIS+ configuration, you might also need to run the `/usr/bin/keylogin` command. See `keylogin(1)` for more information.
  - If your name service is NIS+ (xfn), you might need to do the following:
    1. Log in to the NIS+ master as superuser.
    2. Identify the group that owns the federated naming table:

```
# niscat -o fns.ctx_dir.domain_name.com
            .
            .
            .
         Group   : "admin.domain_name.com"
```

    3. If necessary, add the system that runs Solaris Print Manager to the NIS+ admin group authorized to update the `fns.ctx_dir.`<*domain*> file.

```
# nisgrpadm -a admin.domain_name.com host_name
```

    4. Log in to the system that runs Solaris Print Manager as superuser. Depending on your NIS+ configuration, you might also need to run the

`/usr/bin/keylogin` command. See `keylogin(1)` for more information.

- Have the `SUNWppm` package installed.

```
# pkginfo | grep SUNWppm
system      SUNWppm        Solaris Print Manager
```

2. **Start Solaris Print Manager by selecting Printer Administration from the Tools option of the CDE Workspace menu. Or, select the Applications menu from the CDE front panel, and click the Printer Administration icon in the Application Manager's System_Admin window. You can also start Solaris Print Manager by using the following command.**

```
# /usr/sadm/admin/bin/printmgr &
```

The Select Naming Service window overlays the Solaris Print Manager main window.

If you want to use Solaris Print Manager from a remote system, set the `DISPLAY` environment variable, and then start Solaris Print Manager:

```
# DISPLAY=hostname:display_number
# export DISPLAY
# /usr/sadm/admin/bin/printmgr &
```

Note - If Solaris Print Manager fails to start from the CDE menu or from the command line, check the following:

1. Superuser (root) might not have permission to connect to the Xserver process on the local or remote system. If this happens, type this command:

```
$ xhost +hostname
$ su
(Enter root's password)
# /usr/sadm/admin/bin/printmgr &
```

   Replace *hostname* with either the local or remote system name before restarting Solaris Print Manager.

2. Verify the SUNWppm package is installed on the local or remote system.

```
$ pkginfo | grep SUNWppm
```

3. **Select the name service used in your network from the Select Naming Service window. Choices are: NIS+ (xfn), NIS+, NIS, or files.**

4. **Check that the domain name is correct.**

   The Solaris Print Manager main menu is displayed after the name service is loaded successfully.

Setting Up a Print Server

When you install an attached printer and/or a network printer to a system, the printer is made accessible to the local system. The system on which you install the printer becomes the *printer server*.

The following sections describe how to use the new Solaris Print Manager to add an attached printer or a network printer on a printer server. The example that follows each Solaris Print Manager procedure describes how to add a printer with LP commands.

▼ How to Add a New Attached Printer With Solaris Print Manager

1. **Select the system which is to be the printer server.**

2. **Connect the printer to the printer server and turn on the power to the printer.**

   Consult the printer vendor's installation documentation for information about the hardware switches and cabling requirements.

3. **Start Solaris Print Manager on the printer server where you connected the printer.**

   See "How to Start Solaris Print Manager" on page 66 for instructions.

4. **Select New Attached Printer from the Printer menu.**

   The New Attached Printer window is displayed.

5. **Fill in the window.**

   If you need information to complete a field, click the Help button.

6. **Click OK.**

7. **Verify that the printer has been installed by checking for the new printer entry in the Solaris Print Manager main window. Verify the printer can print requests by using the following command:**

   ```
   $ lp -d printer-name filename
   ```

8. **Exit Solaris Print Manager.**

   Choose Exit from the Print Manager Menu.

Example—Adding a New Attached Printer With LP Commands

This example shows how to make a local PostScript printer available for printing on a print server. The commands in this example must be executed on the print server where the printer is connected. The following information is used as an example. The information you provide will vary:

- Printer name: `luna`
- Port device: `/dev/term/b`
- Printer type: `PS`
- File content type: `postscript`

```
# chown lp /dev/term/b
# chmod 600 /dev/term/b 1
# lpadmin -p luna -v /dev/term/b 2
# lpadmin -p luna -T PS 3
# lpadmin -p luna -I postscript 4
# cd /etc/lp/fd
# for filter in *.fd;do
    > name=‘basename $filter .fd‘
    > lpfilter -f $name -F $filter
    > done 5
# accept luna
 destination ‘‘luna’’ now accepting requests
# enable luna 6
printer ‘‘luna’’ now enabled
# lpadmin -p luna -D "Room 1954 ps" 7
# lpstat -p luna 8
 printer luna is idle. enabled since Jul 12 11:17 1999. available.
```

1. Gives `lp` ownership and sole access to a port device.
2. Defines the printer name and the port device the printer will use.
3. Sets the printer type of the printer.
4. Specifies the file content types to which the printer can print directly.
5. Adds print filters to the print server.
6. Accepts print requests for the printer and enables the printer.
7. Adds a description for the printer.
8. Verifies that the printer is ready.

Where to Go From Here

Use the following table to determine which tasks to complete next.

If You Need To ...	See ...
Add access to the newly installed printer on the print clients because you did not add the printer information to the name service database	“How to Add Printer Access With Solaris Print Manager” on page 72
Set up a `.printers` file	“How to Set Up a `.printers` File” on page 73

Setting Up a Print Client

A print client is a system that is not the server for the printer, yet has access to the printer. A print client uses the services of the print server to spool, schedule and filter the print jobs. Note that one system can be a print server for one printer and be a print client for another printer.

Access to a printer can be configured on a domain-wide basis or on a per-machine basis depending on whether you add the printer information to the name service database.

The following sections describe how to use the new Solaris Print Manager to add access to a printer on a print client. The example that follows this procedure describes how to add printer access with LP commands.

▼ How to Add Printer Access With Solaris Print Manager

1. **Start Solaris Print Manager on the system where you want to add access to a remote printer.**

   See "How to Start Solaris Print Manager" on page 66 for instructions.

2. **Select Add Access to Printer from the Printer menu.**

   The Add Access to Printer window is displayed.

3. **Fill in the window.**

   If you need information to complete a field, click the Help button.

4. **Click OK.**

5. **Verify that access to the printer is added by checking for the new printer entry in the Solaris Print Manager main window. Verify the printer can print requests by using the following command:**

   ```
   $ lp -d printer-name filename
   ```

6. **Exit Solaris Print Manager.**

   Choose Exit from the Print Manager Menu.

Example—Adding Printer Access With LP Commands

If you want to print to a remote printer, you must add access to the remote printer. This example shows how to configure access to a printer named `luna`, whose print server is `saturn`. The system `saturn` becomes a print client of the printer `luna`.

```
# lpadmin -p luna -s saturn [1]
# lpadmin -p luna -D "Room 1954 ps" [2]
# lpadmin -d luna [3]
# lpstat -p luna [4]
  printer luna is idle. enabled since Jul 12 11:17 1999. available.
```

1. Identifies the printer and the print server.
2. Adds a description for the printer.
3. Sets the printer as the system's default printer destination.
4. Verifies that the printer is ready.

Setting Up a `.printers` File

There is no need to set up a `.printers` file in your users' home directories if they don't need customized printer information. However, the `.printers` file enables users to establish their own custom printer aliases. You can use the `_default` alias to make a printer the default. You can also set up a special `_all` alias to define a list of printers affected when you cancel a print request or check the status of printers.

Keep in mind use of the .printers file by the LP print service is controlled by the name service switch (`/etc/nsswitch.conf`). The default configuration is that the print service checks a user's home directory to locate printer configuration information before its checks the other name services. This means you can tailor a user's printer configuration file to use custom printer information rather than the shared information in the name service.

See `printers`(4) for more information about the `.printers` file. See `nsswitch.conf`(4) for more information about the name service switch.

▼ How to Set Up a `.printers` File

1. **Log in to the user's system as superuser.**
2. **Start the file editor you want to use to create a `.printers` file in the user's home directory.**

3. **(Optional) Set up the `_default` alias to make a specific printer your default printer, using an entry similar to the one shown in the following example.**

```
_default printer_name
```

4. **(Optional) Set up the `_all` alias to define the printers affected when you cancel a print request or check the status of printers, using an entry similar to the one shown in the next example.**

```
_all printer1,printer2,printer3
```

5. **Save the file as `.printers`.**

Adding a Network Printer

A *network printer* is a hardware device this is directly connect to the network. This means it can be accessed from a print server without actually connecting it the print server with a cable. It has its own system name and IP address. Even though a network printer is not connected to a print server, it is necessary to set up a print server for it. The print server provides queuing capabilities, filtering, and printing administration for the network printer.

Network printers might use one or more special protocols that require a vendor-supplied printing program. The procedures to set up the vendor-supplied printing program can vary. If the printer does not come with vendor supplied support, the Solaris network printer support can be used with most devices; it is strongly advised to use the print vendor supplied software when possible.

The vendor might supply an SVR4 printer interface script to replace the standard printer interface script. If so, their SVR4 interface script will call the vendor-supplied printing program to send the job to the printer. If not, you will need to modify the standard interface script to call the vendor-supplied printing program. You can do this by editing the per-printer copy of the standard interface script to call the vendor-supplied printing program.

The terms used in network printer configuration are:

- Print server: The machine which spools and schedules the jobs for a printer. This is the machine on which the printer is configured.
- Printer-host device: The printer-host device is the software and hardware supplied by a vendor which provides network printer support for a non-network capable

printer. The combination of the printer-host device with one or more printers attached to it creates a *network printer*.

- Printer node: This is either the physical printer or the printer-host device. It is the physical printer when the network support resides in the physical printer. It is the printer-host device when an external box is used to provide the network interface. The printer node name is the machine name given with the IP address. This name is selected by the system administrator and has no default or vendor requirement. The printer nodename, as with all nodes, must be unique.
- Printer name: The name entered on the command line when using any of the printer commands. It is selected by the system administrator at the time of printer configuration. Any one physical printer can have several printer or queue names; each provides access to the printer.
- Destination or network printer access name: The internal name of the printer node port that is used by the printer sub-system to access the printer. It is the name of the printer node, or the name of the printer node with a printer vendor port designation. Any printer vendor port designation is explicitly defined in the printer vendor documentation. It is printer specific. In the case where the printer is a printer-host device and a printer, the port designation is documented in the printer-host device documentation. The format is:

  printer_node_name

  or

  printer_node_name:*port_designation*
- Protocol: the over-the-wire protocol used to communicate with the printer. The printer vendor documentation supplies the information regarding the protocol to select. The network printer support supplies both BSD Printer Protocol and raw TCP. Due to implementation variations, you might want to try both.
- Timeout, or retry interval: Timeout is a seed number representing the number of seconds to wait between attempting connections to the printer. This seed number is the smallest amount of time to wait between attempted connections, and increases with an increase in failed connections. After repeated failures to connect to the printer, a message is returned to the user requesting possible human intervention. Attempts to reconnect continue until successful or the job is cancelled by the job owner.

Printer Vendor Supplied Software for Network Printers

Network printers often have software support provided by the printer vendor. If your printer has printer vendor supplied software it is strongly advised that the printer vendor software be utilized. The software is designed to support the attributes of the printer and can take full advantage of the printer capabilities. Read the printer vendor documentation to install and configure the printer under an LP print system.

Sun Support for Network Printers

If the network printer vendor does not provide software support, the Sun supplied software is available. The software provides generic support for network printers and is not capable of providing full access to all possible printer attributes.

A general discussion of how to add a network printer is provided in Chapter 4. The following is a discussion of printer management using the Sun supplied software.

Invoking the Network Printer Support

The software support for network printers is called through the interface script. Configuring a network printer with the network interface script, `netstandard`, causes the network printer support module to be called. The command to configure the printer with the network support is:

```
lpadmin -p printer_name -m netstandard
```

Selecting the Protocol

The print sub-system uses BSD print protocol and raw TCP to communicate with the printer. The printer vendor documentation will provide the information about which protocol to use. In general, we have found that the TCP protocol is more generic across printers.

The command to select the protocol is:

```
lpadmin -p printer_name -o protocol=bsd
```

or

```
lpadmin -p printer_name -o protocol=tcp
```

If the protocol selected is the BSD print protocol, you can further select the order of sending the control file to the printer. Some printers expect the control file, then the data file; others the reverse. See the printer vendor documentation for this information. The default is to send the control file first.

The command to select the ordering is:

```
lpadmin -p printer_name -o bsdctrl=first
```

or

```
lpadmin -p printer_name -o bsdctrl=last
```

Selecting the Printer Node Name

The system administrator selects the printer node name. This name must be unique, as with any node on the network. The printer node name is associated with the IP address of the printer.

Selecting the Destination (or Network Printer Access) Name

The print subsystem requires access information for the printer. This is the name that the subsystem uses when making the network connection to the printer. This name is supplied by the system administrator to the print sub-system through the `lpadmin` command. It becomes part of the printer configuration database. The printer access name is the name of the printer node, sometimes qualified by a port name. Port designation varies across printer vendors. You will find information about port designation in the documentation that is provided with the printer by the printer vendor. The format of printer access name is:

printer_node-name[:*port_designation*]

Example 1—Destination (or Network Printer Access Name) With Port Designation (Number)

A common port designation with TCP is 9100. If the printer node name is pn1, and the printer vendor defines the port as 9100, then the printer access name is: `pn1:9100`. To configure a printer in this case use:

```
lpadmin -p printer_name -o dest=pn1:9100
```

Example 2—Destination (or Network Printer Access Name) With Port Designation (Name)

When using the BSD protocol, the port designation might not be a number, but some name defined by the printer vendor, for example: *xxx*`_parallel_1`. If the printer node name is *cardboard*, then the printer access name is: `cardboard:xxx_parallel_1`. To configure a printer in this case use:

```
lpadmin -p printer_name -o dest=cardboard:xxx_parallel_1
```

Example 3—Destination (or Network Printer Access Name) With No Port Designation

If there is no port designation, and the printer node name is newspaper, the printer access name is the printer node name: `newspaper`. To configure a printer in this case use:

```
lpadmin -p printer_name -o dest=newspaper
```

Setting the Timeout Value

The timeout option is provided to allow for individual selection of the amount of time (in seconds) to wait between successive attempts to connect to the printer. Some printers have a long warm up time and a longer timeout value is advised. The default is 10 seconds.

The timeout value does not impact the success or failure of the print process. It is a seed value which the software uses as the initial timeout count; on repeated failures, this count is increased. A message is sent to the spooler when repeated attempts to connect to the printer fail. This alerts the user that intervention might be required. This could be anything from the printer being turned off, to out of paper. Should these messages be produced too often, for example when the printer is warming up, increasing the timeout value will eliminate spurious messages.

The system administrator can experiment to find the optimal timeout value. The command to set the timeout is:

```
lpadmin -p printer_name -o timeout=n
```

Managing Network Printer Access

Each network printer should have one and only one server that provides access to it. This enables the server to manage the access to the printer and keep jobs coherent.

The default device for the network printer is `/dev/null`. This is sufficient when there is only one queue for the printer. Should more queues be required, set the device to a file. This enables the print system to restrict access to the printer across queues. The following commands create a device file and configure it as the network printer device.

```
touch /path/filename
chmod 600 /path/filename
lpadmin -p printer_name -v /path/filename
```

The following is an example of creating a device file called `devtreedown`.

```
# touch /var/tmp/devtreedown
# chmod 600 /var/tmp/devtreedown
# lpadmin -p treedown -v /var/tmp/devtreedown
```

▼ How to Add a Network Printer Using Printer Vendor Supplied Tools

1. **Connect the printer to the network and turn on the power to the printer.**

   Consult the printer vendor's installation documentation for information about the hardware switches and cabling requirements. Get an IP address and select a name for the printer node. This is equivalent to adding any node to the network.

2. **Follow the printer vendor instructions to add the network printer to a SunOS system that has an SVR4 LP print spooler.**

   Use the printer vendor instructions to configure the network printer. These will be specific to the vendor and printer.

3. **Add client access to the new printer.**

   Now that the printer has been added, create access to the printer for the clients. See "Setting Up a Print Client" on page 72.

4. **Optional tasks to complete.**

   There are several optional tasks you might want to complete when setting up a network printer. See "Setting Up Printing Task Map" on page 64 for pointers to the remaining tasks.

▼ How To Add A Network Printer Using LP Commands

Note - This describes the steps necessary to setup a network printer using the network printer support software. The use of this software is intended for those printers that do not come with vendor supplied software.

---

1. **Connect the printer to the network and turn on the power to the printer.**

   Consult the printer vendor's installation documentation for information about the hardware switches and cabling requirements. Get an IP address and select a name for the printer node. This is equivalent to adding any node to the network.

2. **Collect the information required to configure a network printer.**
   - Printer name
   - Printer server
   - Network printer access name
   - Protocol

- Timeout

See the terms described in “Adding a Network Printer” on page 74 for more information.

3. Define the printer name, the device, the printer type and content type by using the `lpadmin`(1M) command.

a. Define the printer name and the port device the printer will use.

```
# lpadmin -p printer-name -v /dev/null
```

The device to use is `/dev/null`.

b. Identify the interface script the printer will use.

```
# lpadmin -p printer-name -m netstandard
```

The interface script that is supplied with the network printer support software is `/usr/lib/lp/model/netstandard`.

c. Set the printer destination, protocol, and timeout values.

```
# lpadmin -p printer-name -o dest=access-name:port -o protocol=protocol
-o timeout=value
```

–p *printer-name*	Specifies the network printer name.
–o `dest=`*access-name:port*	Sets the printer destination to the network printer access name and a designated printer vendor port, if it is defined in the printer vendor documentation. See “Adding a Network Printer” on page 74 for more information.
–o `protocol:`*protocol*	Sets the over-the-wire protocol used to communicate with the printer. Both BSD and raw TCP are supported.
–o `timeout:`*value*	Sets a retry timeout value that represents a number of seconds to wait between attempting connections to the printer. See “Adding a Network Printer” on page 74 for more information.

d. Specify the file content types of the printer and the printer type.

```
# lpadmin -p printer-name -I content-type -T printer-type
```

4. Add filters to the print server by using the `lpfilter`(1M) command.

```
# cd /etc/lp/fd
# for filter in *.fd;do
    > name='basename $filter .fd'
    > lpfilter -f $name -F $filter
    > done
```

5. Enable the printer to accept printer requests and to print the requests.

```
# accept printer-name
# enable printer-name
```

6. Verify the printer is correctly configured by using the `lpstat`(1M) command.

```
# lpstat -p printer-name
```

7. Add client access to the new printer.

Now that the printer has been added, create access to the printer for the clients. See "Setting Up a Print Client" on page 72.

8. Optional tasks to complete.

There are several optional tasks you might want to complete when setting up a printer. See "Setting Up Printing Task Map" on page 64 for pointers to the remaining tasks.

The commands in this example must be executed on the print server. The following information is used as an example. The information you provide will vary.

- Printer name: `luna1`

- Server: `saturn`
- Network printer access name: `nimquat:9100`
- Protocol: `tcp`
- Timeout: 5
- Interface: `/usr/lib/lp/model/netstandard`
- Printer type: `PS`
- Content types: `postscript`
- Device: `/dev/null`

```
# lpadmin -p luna1 -v /dev/null 1
# lpadmin -p luna1 -m netstandard 2
# lpadmin -p luna1 -o dest=nimquat:9100 -o protocol=tcp  -o timeout=5 3
# lpadmin -p luna1 -I postscript -T PS 4
# cd /etc/lp/fd
# for filter in *.fd;do
   > name=‘basename $filter .fd‘
   > lpfilter -f $name -F $filter
   > done 5
# accept luna1
 destination "luna1" now accepting requests
# enable luna1 6
printer "luna1" now enabled
# lpadmin -p luna1 -D "Room 1954 ps" 7
# lpstat -p luna1 8
  printer luna1 is idle. enabled since Jul 12 11:17 1999.  available.
```

1. Defines printer name and sets the device to `/dev/null`.
2. Defines the interface script for network printers.
3. Sets the destination, protocol and timeout.
4. Specifies the file content types to which the printer can print directly, and the printer type.
5. Adds print filters to the print server.
6. Accepts print requests for the printer and enables the printer.
7. Adds a description for the printer.
8. Verifies that the printer is ready.

Converting Printer Configuration Information

This section explains how to convert the printer configuration information from systems running the SunOS 5.5.1 release and copy this information to print clients so they can access existing printers.

Note - If you have only a few existing printers, it might be easier to add access to the printers by using Solaris Print Manager rather than convert the printer configuration information and distribute it to print clients. See Table 4–1 information on adding access to printers.

Converting Printer Configuration Information Task Map

The table below provides an overview of converting printer configuration information.

TABLE 4–2 Converting Printer Configuration Information Task Map

Task	Description	For Instructions, Go To
Convert Existing Printer Configuration Information	*Convert Printer Configuration Information For Systems Running the SunOS 5.5.1 Release* If your site uses SunOS 5.5.1 release, convert the printer configuration information in the `/etc/lp/printers` directory to the `/etc/printers.conf` configuration file. This is usually a one-time task. *Convert Printer Configuration Information For a System Running the SunOS 4.1 Release* If your site uses SunOS 4.1 software, convert the printer configuration information in a 4.1 system's `/etc/printcap` file to the `/etc/printers.conf` configuration file. This is usually a one-time task.	"How to Convert Printer Information For a System Running the SunOS 5.5.1 Release" on page 85 "How to Convert Printer Information For a System Running the SunOS 4.1 Release" on page 85
Convert Printer Configuration Information in NIS+ (+xfn) to NIS+ Format	Managing printer configuration information in the NIS+ name service without the underlying xfn application layer provides better performance.	"How to Convert Printer Configuration Information in NIS+ (+xfn) to NIS+ Format" on page 86

Converting Existing Printer Configuration Information

Existing printer configuration information will automatically be converted when installing or upgrading to the Solaris 8 release. This section explains how to convert the printer configuration information for a system running SunOS 5.5.1 release or a system running the SunOS 4.1 release to the `/etc/printers.conf` printer configuration file. You'll use one of two print administration commands to automate the conversion task:

- The `conv_lp`(1M) command enables you to convert information in the `/etc/lp/printers` directory on a SunOS 5.8 system to entries in the system's `/etc/printers.conf` file. See "How to Convert Printer Information For a System Running the SunOS 5.5.1 Release" on page 85 for instructions.
- The `conv_lpd`(1M) command enables you to convert information in a `/etc/printcap` configuration file from a SunOS 4.1 system to entries in a

`/etc/printers.conf` file. See "How to Convert Printer Information For a System Running the SunOS 4.1 Release" on page 85 for instructions.

If you are not using a name service, you should create a master `/etc/printers.conf` file that includes the existing printers at your site. You can then copy the master file to all the print clients or by loading it into the NIS or NIS+ name service. This is a good way to initially enable all the new print clients access to the existing printers at your site.

Caution - If you are using the NIS or NIS+ name service to configure printer information, do not use a `/etc/printers.conf` file on your print clients. A print client uses the `/etc/printers.conf` file first to locate a printer; however, the `/etc/printers.conf` file might conflict with the printer information in the NIS or NIS+ maps and cause unexpected results. To avoid this problem, remove the `/etc/printers.conf` file on print clients when you want them to use NIS or NIS+ for printer information.

▼ How to Convert Printer Information For a System Running the SunOS 5.5.1 Release

1. **Log in as superuser on a system running the SunOS 5.8 release.**

2. **Convert the printer configuration information in the system's `/etc/lp/printers` directory to the `/etc/printers.conf` file.**

```
# /usr/lib/print/conv_lp
```

▼ How to Convert Printer Information For a System Running the SunOS 4.1 Release

1. **Copy the `/etc/printcap` file from a SunOS 4.1 system to a system running the SunOS 5.8 release.**

2. **Log in as superuser on the system running the SunOS 5.8 release to which you copied the `/etc/printcap` file.**

3. **Convert the printer configuration information in the `/etc/printcap` file to the `/etc/printers.conf` file.**

```
# /usr/lib/print/conv_lpd
```

How to Convert Printer Configuration Information in NIS+ (+xfn) to NIS+ Format

The following conversion script can only be run on a system running the Solaris 8 release.

1. **Log in as superuser on the NIS+ master.**

2. **Copy the following conversion script to system and name it something like** `/tmp/convert`**.**

```
#!/bin/sh
#
# Copyright (C) 1999 by Sun Microsystems, Inc.
# All Rights Reserved
#
PRINTER=""

for LINE in ‘lpget -n xfn list | tr "\t " "^A^B"‘ ; do
 LINE=‘echo ${LINE} | tr "^A^B" "\t " | sed -e ’s/^  \t//g’‘
 case "${LINE}" in
  *:)
   PRINTER=‘echo ${LINE} | sed -e ’s/://g’‘
   ;;
  *=*)
   lpset -n nisplus -a "${LINE}" ${PRINTER}
   ;;
 esac
done
```

Note - If you cut and paste this script into a text file, change both ^A^B (caratA caratB) sequences to `Control A Control B`.

3. **Make the script executable.**

```
# chmod 755 /tmp/convert
```

4. **Run the conversion script.**

```
# /tmp/convert
```

CHAPTER 5

Administering Printers (Tasks)

This chapter provides the procedures to administer printers. This is a list of the step-by-step instructions in this chapter.

- "How to Delete a Printer and Remote Printer Access" on page 88
- "How to Check the Status of Printers" on page 91
- "How to Stop the Print Scheduler" on page 93
- "How to Restart the Print Scheduler" on page 94
- "How to Add a Printer Description" on page 94
- "How to Set a System's Default Printer" on page 96
- "How to Make Banner Pages Optional" on page 97
- "How to Turn Off Banner Pages" on page 98
- "How to Define a Class of Printers" on page 100
- "How to Set Fault Alerts for a Printer" on page 101
- "How to Set Printer Fault Recovery" on page 104
- "How to Limit User Access to a Printer" on page 106
- "How to Check the Status of Print Requests" on page 108
- "How to Accept or Reject Print Requests for a Printer" on page 110
- "How to Enable or Disable a Printer" on page 113
- "How to Cancel a Print Request" on page 114
- "How to Cancel a Print Request From a Specific User" on page 115
- "How to Move Print Requests to Another Printer" on page 117
- "How to Change the Priority of a Print Request" on page 118

For overview information about printing and the LP print service, see Chapter 2.

Managing Printers and the Print Scheduler

This section provides instructions for day-to-day tasks you perform to manage printers and the print scheduler.

Deleting Printers and Printer Access

If a printer needs to be replaced or you want to move the printer to a different location, you must delete the printer information from the LP print service before you physically remove it from the print server. You should also make sure that all the current print requests on the printer are printed or moved to another printer to be printed.

Not only does the printer information need to be deleted from the print server, but it also needs to be deleted from the print clients or network name service. If you delete a local printer from a print server, you should delete the remote printer entry from the print clients or network name service. If you move a printer to another print server, you need to delete the old remote print entry from the print clients or network name service and add access to the remote printer in its new location.

See “How to Delete a Printer and Remote Printer Access” on page 88 for detailed information on how to delete a local and remote printer. You can use Solaris Print Manager to delete a local or remote printer; however, Solaris Print Manager does not enable you to move queued print requests to another printer.

▼ How to Delete a Printer and Remote Printer Access

1. **Log in as superuser or lp on a print client that has access to the printer you want to delete.**

2. **Delete information about the printer from the print client.**

```
print-client# lpadmin -x printer-name
```

`-x`	Deletes the specified printer.
printer-name	Name of the printer you want to delete.

Information for the specified printer is deleted from the print client's `/etc/lp/printers` directory.

3. **If the print client does not use another printer on the same print server, delete information about the print server from the print client.**

```
print-client# lpsystem -r print-server
```

`–r`	Removes the specified print server.
print-server	Name of the print server you want to delete.

The print server is deleted from the print client's `/etc/lp/Systems` file.

4. **Repeat Step 2 on page 88 through Step 3 on page 89 on each print client that has access to the printer.**

5. **Log in as superuser or lp on the print server.**

6. **Stop accepting print requests on the printer.**

```
print-server# reject printer-name
```

`reject` *printer-name*	Rejects print requests for the specified printer.

This step prevents any new requests from entering the printer's queue while you are in the process of removing the printer. See "How to Accept or Reject Print Requests for a Printer" on page 110 for a detailed description.

7. **Stop the printer.**

```
print-server# disable printer-name
```

This step stops print requests from printing. See “How to Enable or Disable a Printer” on page 113 for a detailed description on how to stop printing.

8. **Move any print requests that are still in the queue to another printer.**

   See “How to Move Print Requests to Another Printer” on page 117 for a detailed description on how to move print requests to another printer.

9. **Delete the printer from the print server.**

```
print-server# lpadmin -x printer-name
```

   Configuration information for the printer is deleted from the print server’s `/etc/lp/printers` directory.

10. **Delete information about the print clients that were using the printer you just deleted, unless they are still using another printer on the print server.**

```
print-server# lpsystem -r print-client1 [,print-client2...]
```

`–r`	Removes the specified print client.
print-client	Name of the print client you want to delete from the print server. You can specify multiple print clients in this command. Use a space or a comma to separate print client names. If you use spaces, enclose the list of print clients in quotes.

   The specified print clients are deleted from the print server’s `/etc/lp/Systems` file.

11. **Verify the printer information has been deleted.**

    a. **Check the printer information has been deleted on the print client.**

```
print-client$ lpstat -p printer-name -l
```

       You should receive an error indicating that the printer does not exist in the output of the above command.

    b. **Check the printer information has been deleted on the print server.**

```
print-server$ lpstat -p printer-name -l
```

You should receive an error indicating that the printer does not exist in the output of the above command.

Example—Deleting a Printer and Remote Printer Access

In the following example, the commands delete the printer `luna` from the print client `terra` and from the print server `jupiter`, and also delete the print client `terra` from the print server.

```
terra# lpadmin -x luna
Removed ‘‘luna’’.
terra# lpstat -p luna -l
jupiter# lpadmin -x luna
jupiter# lpsystem -r terra
Removed ‘‘terra’’.
jupiter# lpstat -p luna -l
```

Checking Printer Status

Many routine printer administration tasks require information about the status of the LP print service or a specific printer. For example, you can determine which printers are available for use and examine the characteristics of those printers. You can use the `lpstat` command to find out status information about the LP print service or a specific printer.

▼ How to Check the Status of Printers

1. **Log in on any system on the network.**

2. **Check the status of printers by using the `lpstat` command.**

   Only the most commonly used options are shown here. See `lpstat`(1) for other options.

   ```
   $ lpstat [-d] [-p printer-name [-D] [-l]] [-t]
   ```

`–d`	Shows the system's default printer.
`–p` *printer-name*	Shows if a printer is active or idle, when it was enabled or disabled, and whether it is accepting print requests. You can specify multiple printer names with this command. Use a space or a comma to separate printer names. If you use spaces, enclose the list of printer names in quotes. If you don't specify *printer-name*, the status of all printers is displayed.
`–D`	Shows the description of the specified *printer-name*.
`–l`	Shows the characteristics of the specified *printer-name*.
`–t`	Shows status information about the LP print service, including the status of all printers: whether they are active and whether they are accepting print requests.

Examples—Checking the Status of Printers

In the following example, the command requests the name of the system's default printer.

```
$ lpstat -d
system default destination: luna
```

In the following example, the command requests the status of the printer `luna`.

```
$ lpstat -p luna
printer luna is idle. enabled since Jul 12 11:17 1999. available.
```

In the following example, the command requests a description of the printers `asteroid` and `luna`.

```
$ lpstat -p "asteroid luna" -D
printer asteroid faulted. enabled since Jul 12 11:35 1999. available.
unable to print: paper misfeed jam

Description: Printer by break room
printer luna is idle. enabled since Jul 12 11:36 1999. available.
Description: Printer by server room.
```

In the following example, the command requests the characteristics of the printer luna.

```
$ lpstat -p luna -l
printer luna is idle. enabled since Mon Jul 12 15:02:32 ...
        Form mounted:
        Content types: postscript
        Printer types: PS
        Description:
        Connection: direct
        Interface: /usr/lib/lp/model/standard
        After fault: continue
        Users allowed:
                (all)
        Forms allowed:
                (none)
        Banner not required
        Character sets:

        Default pitch:
        Default page size: 80 wide 66 long
        Default port settings:
```

Restarting the Print Scheduler

The print scheduler, lpsched, handles print requests on print servers. However, there might be times when the print scheduler stops running on a system, so print requests stop being accepted or printed.

To restart the print scheduler, you can use the /usr/lib/lp/lpsched command. If a print request was printing when the print scheduler stopped running, the print request will be printed in its entirety when you restart the print scheduler.

▼ How to Stop the Print Scheduler

1. **Log in as superuser or lp on the print server.**

2. **Check to see if the print scheduler is running.**

   ```
   # lpstat -r
   ```

   If the print scheduler is not running, the message scheduler is not running is displayed.

3. **If the print scheduler is running, stop it.**

   ```
   # /usr/lib/lp/lpshut
   ```

▼ How to Restart the Print Scheduler

1. **Log in as superuser or lp on the print server.**

2. **Check to see if the print scheduler is running.**

```
# lpstat -r
```

   If the print scheduler is not running, the message `scheduler is not running` is displayed.

3. **If the print scheduler is not running, start it.**

```
# /usr/lib/lp/lpsched
```

Setting or Resetting Miscellaneous Printer Definitions

This section provides step-by-step instructions on setting or resetting printer definitions. Some of the following printer definitions can be set using Solaris Print Manager. The procedures below use the LP commands to quickly set or reset printer definitions.

▼ How to Add a Printer Description

1. **Log in as superuser or lp on the print server.**

2. **Add a printer description by using the `lpadmin`(1M) command.**

```
# lpadmin -p printer-name -D "comment"
```

–p *printer-name*	Name of the printer for which you are adding a description.
–D *"comment"*	Specifies the characteristics of the printer, such as location or administrative contact. Enclose characters that the shell might interpret (like *, ?, \, !, ^) in single quotation marks.

The printer description is added in the print server's `/etc/lp/printers/`*printer-name*`/comment` file.

3. **Verify the `Description` information is correct.**

```
$ lpstat -p printer-name -l
```

Example—Adding a Printer Description

In the following example, the command adds a printer description for the printer `luna`.

```
# lpadmin -p luna -D "Nathans office"
```

Setting Up a Default Printer Destination

You can specify a default printer destination for a user so the user doesn't need to type the printer name when using the print commands. Before you can designate a printer as the default, the printer must be known to the print service on the system. You can set a user's default printer destination by setting any of the following:

- `LPDEST` environment variable
- `PRINTER` environment variable
- The `_default` variable in the user's `.PRINTERS` file
- System's default printer (by using the `lpadmin -d` command or Admintool)

When an application provides a printer destination, that destination is used by the print service, regardless of whether you have set a system's default printer destination. If an application doesn't provide a printer destination or if you don't provide a printer name when using a print command, the print command searches for the default printer in a specific order. The table below shows the search order for a system's default printer destination.

TABLE 5–1 Search Order for Default Printer Destinations

Search Order	**Using `/usr/bin/lp` Command**	**Using SunOS/BSD Compatibility Commands (`lpr`, `lpq`, and `lprm`)**
First	`LPDEST` variable	`PRINTER` variable
Second	`PRINTER` variable	`LPDEST` variable
Third	System's default printer	System's default printer

TABLE 5–1 Search Order for Default Printer Destinations *(continued)*

▼ How to Set a System's Default Printer

1. **Log in as superuser or lp on the system for which you want to set a default printer.**

2. **Set the system's default printer by using the `lpadmin` command.**

```
# lpadmin -d [printer-name]
```

`-d` *printer-name*	Name of the printer you are assigning as the system's default printer. If you don't specify *printer-name*, the system is set up with no default printer.

The default printer name is entered in the system's `/etc/lp/default` file.

3. **Check the system's default printer by using the `lpstat` command.**

```
$ lpstat -d
```

Example—Setting a System's Default Printer

In the following example, the command sets the printer `luna` as the system's default printer. This means that `luna` will be used as the system's default printer if the `LPDEST` or `PRINTER` environment variables are not set.

```
# lpadmin -d luna
# lpstat -d
system default destination: luna
```

Printing Banner Pages

A banner page identifies who submitted the print request, the print request ID, and when the request was printed. A banner page will also have a modifiable title to help users identify their printouts.

Banner pages make identifying the owner of a print job easy, which is especially helpful when many users submit jobs to the same printer. Printing banner pages uses more paper, however, and might not be necessary if a printer has only a few users. In some cases, printing banner pages is undesirable. For example, if a printer has

special paper or forms mounted, like paycheck forms, printing banner pages might cause problems.

By default, the print service forces banner pages to be printed. However, you can give users a choice to turn off printing of a banner page when they submit a print request. You can set this choice through the `lpadmin` command or through Admintool. If you give the users a choice, they have to use the `–o nobanner` option to turn off printing of a banner page.

Also, you can turn off banner pages for a printer so they are never printed. This is important if you have a situation where you don't need or want banner pages. You can turn off banner page printing by using the `lpadmin` command.

TABLE 5–2 Banner Page Printing

With This Command	Banner Page Printing Is ...	Override?
`lpadmin -p` *printer* `–o banner` or `lpadmin -p` *printer* `–o banner=always`	Required and printed	If you are a regular user and use `lp -o nobanner`, the request is printed, but the `nobanner` argument is ignored. If you are root or lp, the `nobanner` argument is honored.
`lpadmin -p` *printer* `–o nobanner` `lpadmin -p` *printer* `–o banner=optional`	On by default, but can be disabled on a per request basis with the `lp -o nobanner` command.	N/A
`lpadmin -p` *printer* `–o banner=never`	Disabled	No

For step-by-step command-line instructions, see "How to Turn Off Banner Pages" on page 98.

▼ How to Make Banner Pages Optional

1. **Log in as superuser or lp on the print server.**

2. **Make banner pages optional by using the `lpadmin` command.**

```
# lpadmin -p printer-name -o nobanner=optional
```

–p *printer-name*	Name of the printer for which you are making banner pages optional.
–o nobanner=optional	Enables users to specify no banner page when they submit a print request.

If you want to force a banner page to print with every print request, specify the –o banner=always option.

The banner page setting is entered in the print server's /etc/lp/printers/*printer-name*/configuration file.

3. **Verify the output from the following command contains the line** Banner not required**.**

```
$ lpstat -p printer-name -l
```

Example—Making Banner Pages Optional

In the following example, the command enables users to request no banner page on the printer luna.

```
# lpadmin -p luna -o nobanner=optional
```

▼ How to Turn Off Banner Pages

1. **Log in as superuser or lp on the print server.**

2. **Turn off banner printing by using the** lpadmin **command.**

```
lpadmin -p printer-name -o nobanner=never
```

–p *printer-name*	Name of the printer for which you are making banner pages optional.
–o nobanner=never	Disables banner page printing under all circumstances.

The banner page setting is entered in the print server's /etc/lp/printers/*printer-name*/configuration file.

3. **Verify the output from the following command contains the line** `Banner not printed`**.**

```
$ lpstat -p printer-name -l
```

4. **Submit a print request to the printer to make sure a banner page does not print.**

Example—Turning Off Printing Banner Pages

In the following example, the command disables printing banner pages on the printer `luna`.

```
# lpadmin -p luna -o nobanner=never
```

Setting Up Printer Classes

The print service enables you to group several locally attached printers into one class. You can perform this task only by using the `lpadmin -c` command.

When you have set up a printer class, users can then specify the class (rather than individual printers) as the destination for a print request. The first printer in the class that is free to print is used. The result is faster turnaround because printers are kept as busy as possible.

There are no default printer classes known to the print service; printer classes exist only if you define them. Here are some ways you could define printer classes:

- By printer type (for example, PostScript)
- By location (for example, 5th floor)
- By work group or department (for example, Accounting)

Alternatively, a class might contain several printers that are used in a particular order. The LP print service always checks for an available printer in the order in which printers were added to a class. Therefore, if you want a high-speed printer to be accessed first, you would add it to the class before you add a low-speed printer. As a result, the high-speed printer would handle as many print requests as possible. The low-speed printer would be reserved as a backup printer when the high-speed printer is in use.

Note - Print requests are balanced between printers in a class only for local printers.

Class names, like printer names, must be unique and can contain a maximum of 14 alphanumeric characters and underscores.

You are not obligated to define printer classes. You should add them only if you determine that using printer classes would benefit users on the network.

▼ How to Define a Class of Printers

1. **Log in as superuser or lp on the print server.**

2. **Define a class of printers by using the `lpadmin` command.**

```
# lpadmin -p printer-name -c printer-class
```

–p *printer-name*	Name of the printer you are adding to a class of printers.
–c *printer-class*	Name of a class of printers.

The specified printer is added to the end of the list in the class in the print server's `/etc/lp/classes/`*printer-class* file. If the printer class does not exist, it is created.

3. **Verify the printers in a printer class by using the `lpstat` command.**

```
$ lpstat -c printer-class
```

Example—Defining a Class of Printers

In the following example, the command adds the printer `luna` in the class `roughdrafts`.

```
# lpadmin -p luna -c roughdrafts
```

Setting Up Printer Fault Alerts

If you choose, the print service can notify you when it detects a printer fault. You can select any of the following methods to receive printer fault notification with the `lpadmin -A` command or with Solaris Print Manager:

- Write a message to the terminal on which root is logged in
- Electronic mail to root
- No notification

However, the `lpadmin -A` command offers you an additional option of receiving a message specified by the program of your choice. It also enables you to selectively turn off notification for an error that you already know about.

Unless you specify a program to deliver fault notification, the content of the fault alert is a predefined message that says the printer has stopped printing and needs to be fixed.

The table below lists the alert values that you can set for a printer with the `lpadmin -A` command. These alert values can also be set for print wheels, font cartridges, and forms.

TABLE 5–3 Values for Printing Problem Alerts

Value for –A `alert`	**Description**
'`mail` [*user-name*]'	Send the alert message by email to root or `lp` on the print server, or the specified *user-name*, which is a name of a user.
'`write` [*user-name*]'	Send the alert message to the root or `lp` console window on the print server, or to the console window of the specified *user-name*, which is a name of a user. The specified user must be logged in to the print server to get the alert message.
'*command*'	Run the *command* file for each alert. The environment variables and current directory are saved and restored when the file is executed.
`quiet`	Stop alerts until the fault is fixed. Use this when you (root or specified user) receive repeated alerts.
`none`	Do not send any alerts. This is the default if you don't specify fault alerts for a printer.

▼ How to Set Fault Alerts for a Printer

1. **Log in as superuser or lp on the print server.**

2. **Set fault alerts for a printer with the `lpadmin` command.**

```
# lpadmin -p printer-name -A alert [-W minutes]
```

–p *printer-name*	Name of the printer for which you are specifying an alert for printer faults.
–A *alert*	Specifies what kind of alert will occur when the printer faults. See Table 5–3 for detailed information about the valid values for *alert*. Some valid values are `mail`, `write`, and `quiet`.
–W *minutes*	Specifies how often (in minutes) the fault alert will occur. If you don't specify this option, the alert is sent once.

The fault alert setting is entered in the print server's `/etc/lp/printers/`*printer-name*`/alert.sh` file.

3. **Check the information following the `On fault` heading from the output of the following command.**

```
$ lpstat -p printer-name -l
```

Examples—Setting Fault Alerts for a Printer

In the following example, the command sets up the printer `mars` to send fault alerts by email to a user named `joe`, with reminders every 5 minutes.

```
# lpadmin -p mars -A 'mail joe' -W 5
```

In the following example, the command sets up the printer `venus` to send fault alerts to the console window, with reminders every 10 minutes.

```
# lpadmin -p venus -A write -W 10
```

In the following example, the command stops fault alerts for the printer `mercury`.

```
# lpadmin -p mercury -A none
```

In the following example, the command stops fault alerts until the printer `venus` has been fixed.

```
# lpadmin -p venus -A quiet
```

Setting Up Printer Fault Recovery

If you choose not to send any fault notification, you can find out about printing faults so you can correct the problem. The LP print service will not continue to use a printer that has a fault. In addition to alerts for printer faults, you can also provide

alerts that tell the system administrator to mount print wheels, font cartridges, and forms when print requests require them.

You can define the fault recovery options for a printer only by using the `lpadmin -F` command. This task is not available in Solaris Print Manager.

Printer faults can be as simple as running out of paper or needing to replace a toner cartridge. Other more serious problems can include complete printer failure or power failure. After you fix a printer fault, the print request that was active when the fault occurred begins printing in one of three ways:

- Starts printing from the beginning
- Continues printing from the top of the page where printing stopped
- After you enable the printer, continues printing from the top of the page where the printing stopped

A print filter is required to continue printing from the top of a page where the printing stopped. A print filter records the control sequences used by the printer to track page boundaries, which the default filters used by the print service cannot do. You will be notified by the print service if recovery cannot proceed with the specified print filter. For information about writing filters, see "How to Create a New Print Filter" on page 172.

If you want printing to resume immediately after a printer fault is fixed, enable the printer by using the `enable` command.

The table below lists the fault recovery values you can set for a printer with the `lpadmin -F` command.

TABLE 5–4 Values for Printer Fault Recovery

Value for –F ***recover-options***	**Description**
`beginning`	After a fault recovery, printing restarts from the beginning of the file.
`continue`	After a fault recovery, printing starts at the top of the page where the printing stopped. This recovery option requires a print filter.
`wait`	After a fault recovery, printing stops until you enable the printer. After you enable the printer (`enable` command), printing starts at the top of the page where printing stopped. This recovery option requires a print filter.

▼ How to Set Printer Fault Recovery

1. **Log in as superuser or lp on the print server.**

2. **Set up fault recovery for the printer with the `lpadmin`(1M) command.**

```
# lpadmin -p printer-name -F recovery-options
```

–p *printer-name*	Name of the printer for which you are specifying fault recovery.
–F *recovery-options*	One of the three valid recovery options: `beginning`, `continue`, or `wait`. See Table 5–4 for detailed information about the valid values for *recovery-options*.

The fault recovery setting is entered in the print server's `/etc/lp/printers/`*printer-name*`/configuration` file.

3. **Check the information following the `After fault` heading in the output of the following command.**

```
$ lpstat -p printer-name -l
```

Example—Setting Printer Fault Recovery

In the following example, the command sets up the printer `luna` to continue printing at the top of the page where printing stopped.

```
# lpadmin -p luna -F continue
```

Limiting User Access to a Printer

You can control which users can access some or all of the available printers. For example, you can prevent some users from printing on a high-quality printer to minimize expense. To restrict user access to printers, you can create `allow` and `deny` lists using the `lpadmin -u` command on the print server. (Solaris Print Manager enables you to create only allow lists.) If you create neither, a printer is available to all users who can access the printer.

An allow list contains the names of users allowed access to the specified printer; a deny list contains the names of users denied access to the specified printer.

The rules for allow and deny lists are:

When You ...	Then ...
Do not create allow and deny lists, or if you leave both lists empty	All users can access the printer.
Specify `all` in the allow list	All users can access the printer.
Specify `all` in the deny list	All users, except root and lp (on the server), are denied access to the printer.
Make any entry in the allow list	The deny list is ignored. Only those users who are listed can access the printer.
Create a deny list, but you do not create an allow list or you leave the allow list empty	Users who are listed in the deny list are denied access to the printer.

Because the print server is actually controlling access to the printer, allow and deny lists can only be created on the print server itself. If you create allow and deny lists, the print server will exclusively control user access to printers.

The table below lists the values you can add to an allow or deny list to limit user access to a printer.

TABLE 5–5 Values for Allow and Deny Lists

Value for *user-list*	Description
user	*User* on any system
`all`	All users on all systems
none	No user on any system
system!*user*	*User* on *system* only
!*user*	*User* on local system only
`all`!*user*	*User* on any system
`all!all`	All users on all systems

TABLE 5–5 Values for Allow and Deny Lists *(continued)*

Value for *user-list*	Description
system`!all`	All users on *system*
`!all`	All users on local system

▼ How to Limit User Access to a Printer

1. **Log in as superuser or lp on the print server.**
2. **Allow or deny users access to a printer by using the `lpadmin` command.**

```
# lpadmin -p printer-name -u allow:user-list [ deny:user-list]
```

–p *printer-name*	Name of the printer to which the allow or deny user access list applies.
–u `allow:`*user-list*	User names to be added to the allow user access list. You can specify multiple user names with this command. Use a space or a comma to separate names. If you use spaces, enclose the list of names in quotes. Table 5–5 provides the valid values for *user-list*.
–u `deny:`*user-list*	User names to be added to the deny user access list. You can specify multiple user names with this command. Use a space or a comma to separate names. If you use spaces, enclose the list of names in quotes. Table 5–5 provides the valid values for *user-list*.

The specified users are added to the allow or deny user access list for the printer in one of the following files on the print server:

`/etc/lp/printers/`*printer-name*`/users.allow`

`/etc/lp/printers/`*printer-name*`/users.deny`

Note - If you specify none as the value for *user-list* in the allow user access list, the following files are not created for the print server:

`/etc/lp/printers/`*printer-name*`/alert.sh`

`/etc/lp/printers/`*printer-name*`/alert.var`

`/etc/lp/printers/`*printer-name*`/users.allow`

`/etc/lp/printers/`*printer-name*`/users.deny`

3. **Check the information following the** `Users allowed` **or** `Users denied` **heading in the output of the following command.**

```
$ lpstat -p printer-name -l
```

Examples—Limiting User Access to a Printer

In the following example, the command allows only the users `nathan` and `george` access to the printer `luna`.

```
# lpadmin -p luna -u allow:nathan,george
```

In the next example, the command denies the users `nathan` and `george` access to the printer `asteroid`.

```
# lpadmin -p asteroid -u deny:"nathan george"
```

Managing Print Requests

When a user submits a print request from a print client, the print request is added to a queue on the print server before it is sent to the printer. While a print request is in the queue, you can cancel or gain status information on the request from a client system. You must login to the print server to move, hold, resume, or change the priorities of print requests with LP commands. These actions can help you keep printing services operating smoothly.

The table below lists the values for changing the priority of a print request with the `lp -H` command.

TABLE 5–6 Values for Changing the Priority of a Print Request

Value for –H *change-priority*	Description
`hold`	Places the print request on hold until you cancel it or instruct the LP print service to resume printing the request.
`resume`	Places a print request that has been on hold back in the queue. It will be printed according to its priority and placement in the queue. If you put a hold on a print job that is already printing, `resume` puts the print request at the head of the queue so it becomes the next request printed.
`immediate`	Places a print request at the head of the queue. If a request is already printing, you can put it on hold to allow the next request to print immediately.

▼ How to Check the Status of Print Requests

1. **Log in on any system on the network.**

2. **Check the status of printers and print requests by using the `lpstat` command.**

   Only the most commonly used options are shown here. See `lpstat`(1) for other valid options.

```
$ lpstat -o [list] | -u [user-list]
```

–o *list*	Shows the status of print requests on a specific printer. *list* can be one or more printer names, printer class names, or print request IDs. You can specify multiple printer names, class names, and IDs for *list*. Use a space or a comma to separate values. If you use spaces, enclose the list of values in quotes. If you don't specify *list*, the status of print requests to all printers is displayed.
–u *user-list*	Shows the status of print requests for a specific user. *user-list* can be one or more user names. You can specify multiple users with this command. Use a space or a comma to separate user names. If you use spaces, enclose the list of names in quotes. If you don't specify *user-list*, the status of print requests for all users is displayed.

When used to check the status of print requests, the `lpstat` command displays one line for each print request. From left to right, the line shows the request ID, the user, the output size in bytes, the date and time of the request, and information about the request, such as "being filtered."

Examples—Checking the Status of Print Requests

In the following example, the command shows that user `fred` has one print request queued to the printer `luna`.

```
$ lpstat
luna-1    fred     1261    Jul 12 17:34
```

In the following example, the command shows that the user `paul` currently has no print requests in queue.

```
$ lpstat -u paul
```

In the following example, the command shows that there are two print requests on the printer `moon`.

```
$ lpstat -o moon
moon-78    root     1024   Jul 14 09:07
moon-79    root     1024   Jul 14 09:08
```

Processing or Stopping Printing

The `enable`(1) and `disable`(1) commands control whether a printer prints or stops printing requests that are in the print queue. When you disable a printer, the printer stops printing requests in queue; however, requests are still added to the queue. (You must set the printer to reject print requests so requests are not added to the queue. See “Accepting or Rejecting Print Requests” on page 111 for information about rejecting print requests.)

A printer is enabled to print and accepts print requests when it is added using Solaris Print Manager. Solaris Print Manager doesn’t provide any additional printer processing management.

You must enable the printer whenever it has been disabled, which can happen when a printer fault occurs. When you enable a printer, it prints requests from the print queue until the queue is empty, even if the print service rejects additional requests for the print queue.

The figure below shows the point at which processing of print requests is interrupted when a printer is disabled.

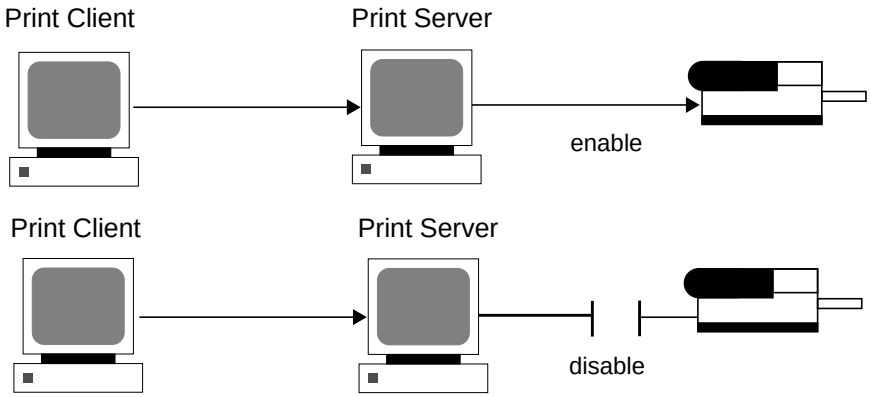


Figure 5–1 What Happens When a Printer Is Enabled or Disabled

▼ How to Accept or Reject Print Requests for a Printer

1. **Log in as superuser or lp on the print server.**

2. **Stop accepting print requests for the printer by using the `reject`(1M) command.**

```
# reject [-r "reason"] printer-name
```

`–r` *"reason"*	Provides users a reason why the printer is rejecting print requests. The reason is stored and displayed whenever a user checks on the status of the printer (`lpstat -p`).
printer-name	Name of the printer that will stop accepting print requests.

The queued requests will continue printing as long as the printer is enabled. For instructions on disabling a printer so it stops printing, see "How to Enable or Disable a Printer" on page 113.

3. **Start accepting print requests for the printer by using the `accept`(1M) command.**

```
# accept printer-name
```

4. **Check the status of the printer to see whether it is accepting or rejecting print requests by using the `lpstat` command.**

```
$ lpstat -p printer-name
```

Examples—Accepting or Rejecting Print Requests for a Printer

In the following example, the command stops the printer `luna` from accepting print requests.

```
# reject -r "luna is down for repairs" luna
destination "luna" will no longer accept requests
```

In the following example, the command sets the printer `luna` to accept print requests.

```
# accept luna
destination "luna" now accepting requests
```

Accepting or Rejecting Print Requests

The `accept` and `reject` commands enable you to turn on or turn off a print queue that stores requests to be printed.

When you use the `reject` command, the print queue for a specified printer is turned off—no new print requests can enter the queue on the print server. All print requests that are in the queue are still printed. You must disable the printer if you

want it to stop printing requests that are already in the queue. Table 5–7 compares the functions of the `accept`, `reject`, `enable`, and `disable` commands.

TABLE 5–7 Functions of `accept`/`reject` and `enable`/`disable` Commands

Command	Function
`accept`	Accept print requests that are sent to the print queue.
`enable`	Print the requests that are in the print queue.
`reject`	Reject print requests that are sent to the print queue.
`disable`	Stop printing requests that are currently in the print queue.

See "Processing or Stopping Printing" on page 110 for information about disabling a printer.

If a print request is rejected, the print service writes or mails a message to the user who submitted the request, saying that print requests are not being accepted for the specified printer.

You can also specify a reason for not accepting requests through the command line. The reason is displayed on users' systems when one tries to check the printer's queue. The figure below shows the point at which processing of print requests is interrupted when a print queue rejects print requests.

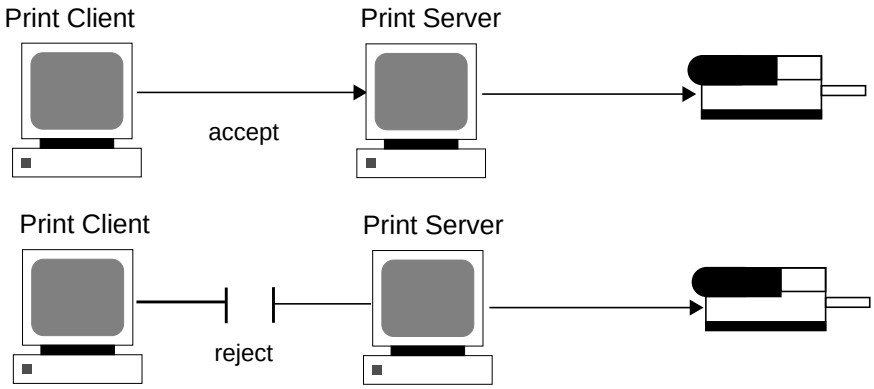


Figure 5–2 What Happens When a Print Queue Accepts or Rejects Requests

▼ How to Enable or Disable a Printer

1. **Log in as superuser or** `lp` **on the print server.**

2. **Stop printing print requests on the printer by using the** `disable` **command.**

```
# disable [-c | -W] [-r "reason"] printer-name
```

`disable`	Cancels the current job, then disables the printer. The current job is saved to reprint when the printer is enabled.
`–c`	Cancels the current job, then disables the printer. The current job is not printed later.
`–W`	Waits until the current job is finished before disabling the printer.
`–r` *"reason"*	Provides users with a reason why the printer is disabled. The reason is stored and displayed whenever a user checks on the status of the printer (`lpstat -p`).
printer-name	Name of the printer that will stop printing print requests.

Note - You cannot enable or disable classes of printers. Only individual printers can be enabled or disabled.

---

3. **Start printing print requests on the printer by using the** `enable` **command.**

```
# enable printer-name
```

4. **Verify the printer is enabled.**

```
$ lpstat -p printer-name
```

Examples—Enabling or Disabling a Printer

In the following example, the command stops the current job on the printer `luna`, saves it to print later, and provides a reason why the printer has stopped printing print requests.

```
# disable -r "changing the form" luna
```

In the following example, the command starts printing print requests on the printer luna.

```
# enable luna
printer "luna" enabled
```

Canceling a Print Request

You can use the `cancel`(1) to cancel print requests from printer queues or to cancel jobs that are printing. There are three ways to use the `cancel` command:

- Cancel requests by request identification number (request ID)
- Cancel requests from a specific user on all, or specified, printers
- Cancel the job currently printing

When you use `cancel`, a message tells you the request(s) are canceled, and the next request in queue is printed. You can cancel a print request only if you are:

- The user who submitted the request and you are logged in on the system from which you submitted the request
- The user who submitted the request on any client system and the print server has the "user-equivalence" option configured for the printer in it's `/etc/printers.conf` file.
- Logged in as superuser or `lp` on the print server.

To cancel a specific request, you need to know its request ID. The request ID is comprised of the name of the printer, a dash, and the number of the print request—for example, `luna-185`. When you submit the print request, the request ID is displayed. If you do not remember the print request ID, you can find it by using the `lpstat` command with the `-o printer` option.

▼ How to Cancel a Print Request

1. **If you are going to cancel print requests of other users, become superuser or `lp`.**

2. **Determine the request IDs of the print requests to cancel by using the `lpstat` command.**

   See "How to Check the Status of Print Requests" on page 108 for more details.

3. **Cancel a print request by using the `cancel` command.**

   ```
   $ cancel request-id | printer-name
   ```

request-id	Request ID of a print request to be canceled. You can specify multiple request IDs with this command. Use a space or a comma to separate request IDs. If you use spaces, enclose the list of request IDs in quotes.
printer-name	Specifies the printer for which you want to cancel the currently printing print request. You can specify multiple printer names with this command. Use a space or a comma to separate printer names. If you use spaces, enclose the list of printer names in quotes.

4. Verify the print requests are canceled.

```
$ lpstat -o printer-name
```

Examples—Canceling a Print Request

In the following example, the command cancels the `luna-3` and `luna-4` print requests.

```
$ cancel luna-3 luna-4
request "luna-3" cancelled
request "luna-4" cancelled
```

In the following example, the command cancels the print request that is currently printing on the printer `luna`.

```
# cancel luna
request "luna-9" cancelled
```

▼ How to Cancel a Print Request From a Specific User

1. (Optional) Become superuser or lp if you are going to cancel print requests of other users.

2. Cancel a print request from a specific user with the `cancel` command.

```
$ cancel -u user-list [printer-name]
```

–u *user-list*	Cancels the print request for a specified user. *user-list* can be one or more user names. Use a space or a comma to separate user names. If you use spaces, enclose the list of names in quotes.
printer-name	Specifies the printer for which you want to cancel the specified user's print requests. *printer-name* can be one or more printer names. Use a space or a comma to separate printer names. If you use spaces, enclose the list of printer names in quotes. If you don't specify *printer-name*, the user's print requests will be canceled on all printers.

Examples—Canceling a Print Request From a Specific User

In the following example, the command cancels all the print requests submitted by the user `george` on the printer `luna`.

```
# cancel -u george luna
request "luna-23" cancelled
```

In the following example, the command cancels all the print requests submitted by the user `george` on all printers.

```
# cancel -u george
request "asteroid-3" cancelled
request "luna-8" cancelled
```

Moving a Print Request

If you plan to change the way a printer is used or decide to take a printer out of service, you should set up the LP print service to reject additional print requests, and then move or cancel any requests that are currently queued to the printer. You can use the `lpmove`(1M) command to move individual or all print requests to another local printer.

Request IDs are not changed when you move print requests, so users can still find their requests. Print requests that have requirements (such as file content type or forms) that cannot be met by the newly specified printer cannot be moved; they must be canceled.

▼ How to Move Print Requests to Another Printer

To move all print requests from one printer to another, you do not need to know the request IDs; however, it is a good idea to see how many print requests are affected before you move them.

1. **Log in as superuser or lp on the print server.**

2. **(Optional) Check the request IDs of the print requests on the original printer.**

```
# lpstat -o printer-name1
```

3. **(Optional) Check if the destination printer is accepting print requests.**

```
# lpstat -p printer-name2
```

–p *printer-name2*	Name of the printer to which you are moving the print requests.

4. **Move all the print requests from the original printer to the destination printer.**

```
# lpmove printer-name1 printer-name2
```

printer-name1	Name of the printer from which all print requests will be moved.
printer-name2	Name of the printer to which all print requests will be moved.

If some requests cannot be printed on the destination printer, they are left in the original printer's queue. By using request IDs, you can also move specific print requests to another printer with the `lpmove` command.

5. **Start accepting print requests on the original printer.**

If you move all the print requests to another printer, the `lpmove` command automatically stops accepting print requests for the printer. This step is necessary if you want to begin accepting new print requests for the printer.

```
# accept printer-name1
```

6. **Check for any remaining print requests in the original printer's queue by using the following command.**

```
$ lpq -P printer-name1
```

Make sure all specified print requests were moved to the destination printer's queue by using the following command.

```
$ lpq -P printer-name2
```

Example—Moving Print Requests to Another Printer

In the following example, the `lpmove` command moves print requests from the printer `luna` to the printer `terra`, and the `accept` command tells the original printer `luna` to resume accepting print requests.

```
# lpmove luna terra
# accept luna
```

Changing the Priority of Print Requests

After a user has submitted a print request, you can change its priority in the print server's queue by:

- Putting any print request on hold if it has not finished printing. Putting a request on hold stops it, if it is currently printing, and keeps it from printing until you resume printing it. Other print requests go ahead of the on-hold request.
- Moving any print request to the head of the queue, where it will be the next job eligible for printing. If you want a job to start printing immediately, you can interrupt the job that is currently printing by putting it on hold.
- Changing the priority of a job still waiting to be printed, moving it in the queue so it is ahead of lower priority requests and behind requests at the same level or at a higher priority.

▼ How to Change the Priority of a Print Request

1. **Log in as superuser or lp on the print server that is holding the print request.**

2. **Determine the request IDs of the print requests whose priority you want to change by using the `lpstat` command.**

   See "How to Check the Status of Print Requests" on page 108 for more information.

3. Change the priority of a print request by using the `lp` command.

```
# lp -i request-id -H change-priority
```

–`i` *request-id*	Request ID of a print request you want to change. You can specify multiple request IDs with this command. Use a space or a comma to separate request IDs. If you use spaces, enclose the list of request IDs in quotes.
–`H` *change-priority*	One of the three ways to change the priority of a print request: `hold`, `resume`, `immediate`. See Table 5–6 for detailed information about valid values for *change-priority*.

You can also use the `lp -q` command to change the priority level of a specified print request. You can change the priority level from 0, the highest priority, to 39, the lowest priority.

Example—Changing the Priority of a Print Request

In the following example, the command changes a print request with the request ID `asteroid-79`, to priority level 1.

```
# lp -i asteroid-79 -q 1
```

CHAPTER 6

Managing Character Sets, Filters, Forms, and Fonts (Tasks)

This chapter provides background information and step-by-step instructions for setting up and administering character sets, print filters, forms, and fonts.

This is a list of the step-by-step instructions in this chapter.

- "How to Define a Print Wheel or Font Cartridge" on page 124
- "How to Unmount and Mount a Print Wheel or Font Cartridge" on page 125
- "How to Set an Alert to Mount a Print Wheel or Font Cartridge" on page 126
- "How to Set Up an Alias for a Selectable Character Set" on page 128
- "How to Add a Print Filter" on page 132
- "How to Delete a Print Filter" on page 133
- "How to View Information About a Print Filter" on page 133
- "How to Add a Form" on page 137
- "How to Delete a Form" on page 138
- "How to Unmount and Mount a Form" on page 138
- "How to Set an Alert to Mount a Form" on page 140
- "How to View Information About a Form" on page 142
- "How to View the Current Status of a Form" on page 143
- "How to Limit User Access to a Form" on page 143
- "How to Limit Printer Access to a Form" on page 144
- "How to Install Downloaded PostScript Fonts" on page 148
- "How to Install Host-Resident PostScript Fonts" on page 148

For overview information about printing, see Chapter 2.

Managing Character Sets

Printers differ in the method they use to print text in various font styles. For example, PostScript printers treat text as graphics. These printers can generate text in different fonts, and place the text in any position, size, or orientation on the page. Other types of printers support a more limited number of font styles and sizes, using either print wheels, font cartridges, or preprogrammed selectable character sets. Usually, only one of these printing methods applies to a given printer type.

Print wheels and font cartridges, from the perspective of the LP print service, are similar, because someone must intervene and mount the hardware on the printer, when needed. Character sets that require you to physically mount a wheel or cartridge are referred to as *hardware character sets*. Character sets that do not require hardware mounting, that come preprogrammed with the printer, and can be selected by a print request, are referred to as *software character sets*.

When you set up a non-PostScript printer, you need to tell the LP print service which print wheels or selectable character sets are available to users. When users submit print requests, the `lp -S` command enables them to specify a print wheel or selectable character set to use for the print job. Users do not have to know which type of character set applies; they just refer to the font style by the name you have defined. For example, you can define a print wheel as `gothic`. To request the `gothic` print wheel, the user would enter `lp -S gothic`.

Selectable Character Sets

The selectable character sets supported by a printer are listed in the `terminfo` entry for that printer. For example, the entry for the `ln03` printer is `/usr/share/lib/terminfo/l/ln03`. You can find the names of selectable character sets for any printer type in the `terminfo` database by using the `tput` command. The syntax for the `tput` command is:

```
tput -T printer-type csn
```

The `cs`*n* option is an abbreviation for character set number. The number starts with 0, which is always the default character set number after the printer is initialized. You can repeat the command, using –`1`, –`2`, –`3`, and so on in place of the –`0`, to display the names of the other character sets. For each selectable character set, a `terminfo` name (for example, `usascii`, `english`, `finnish`, and so forth) is returned.

In general, the `terminfo` character set names should closely match the character set names used in the manufacturer's documentation for the printer. Because manufacturers do not all use the same character set names, the `terminfo` character set names can differ from one printer type to the next.

You do not have to register the selectable character set names with the LP print service. However, you can give them more meaningful names or aliases.

Note - If you do not specify the selectable character sets that can be used with a printer, the LP print service assumes that the printer can accept any character set name (such as cs0, cs1, or cs2) or the `terminfo` name known for the printer.

Users can use the `lpstat -p -l` command to display the names of the selectable character sets that you have defined for each printer on a print server.

Note - Character sets for PostScript printers are not listed when you use the `lpstat -p -l` command because the PostScript fonts are controlled by PostScript filters, not by entries in the `terminfo` database. See “Managing Fonts” on page 145 for information about how to administer PostScript fonts.

Hardware-Mounted Character Sets

Another method to obtain alternative character sets is to use removable print wheels or font cartridges that you physically attach, or mount, in a printer.

To administer hardware-mounted character sets, you inform the LP print service of the names you want to use for the available print wheels, and how you want to be alerted when a printer needs a different print wheel. Then, when a user requests a particular character set with the `lp -S` command, the scheduler sends an alert to mount the print wheel, and the print request is placed in the print queue. When you mount the correct print wheel and tell the LP print service that the print wheel is mounted, the job is printed. See “How to Unmount and Mount a Print Wheel or Font Cartridge” on page 125 for more information.

If you do not specify multiple print wheels or cartridges for a printer, the LP print service assumes that the printer has a single, fixed print wheel or cartridge, and users cannot specify a special print wheel or cartridge when using the printer.

Unlike selectable character sets, the names you use for print wheels or cartridges are not tied to entries in the `terminfo` database. Print wheel or cartridge names are used only for the purpose of communicating with the LP print service and its users.

The names you choose for print wheels or cartridges, however, should have meaning to the users; the names should refer to font styles. In addition, the names should be the same across printers that have similar print wheels or cartridges, or selectable character sets. That way, users can ask for a font style (character set) without regard to which printer—or even whether a print wheel or cartridges—or selectable character set will be used.

Of course, you and the printer users should agree on the meanings of print wheel or cartridge names. Otherwise, what a user asks for and what you mount, might not be the same character set.

Tracking Print Wheels

The procedure for tracking print wheels is similar to the procedure for tracking forms. Some printers (usually letter-quality printers) have removable print heads, such as print wheels or print cartridges, that provide a particular font or character set. A user can request a named character set. If that character set is not available, the LP print service notifies root of the request. The job is stored in the print queue until the print wheel is changed.

Alerts for Mounting Print Wheels or Cartridges

You request alerts for mounting print wheels or cartridges in the same way you request other alerts from the LP print service. See "Setting Up Printer Fault Alerts" on page 100 for general information about alerts.

▼ How to Define a Print Wheel or Font Cartridge

1. **Log in as superuser or lp on the print server.**

2. **Define a print wheel or font cartridge that can be used with the printer.**

```
print-server# lpadmin -p printer-name -S hard-charset1[,hard-charset2...]
```

–p *printer-name*	Name of the printer for which you are defining a print wheel or font cartridge.
–s *hard-charset*	Hardware character set name of the print wheel or font cartridge. You can specify multiple hardware character sets with this command. Use commas or spaces to separate character set names. If you use spaces, enclose the list of character set names in quotes. Define names that are meaningful to users, and inform the users of the names.

   The print wheel or font cartridge definition is added in the print server's `/etc/lp/printers/`*printer-name*`/configuration` file.

3. **Log in as superuser or lp on a system that is a print client of the print server.**

4. **Define the same print wheel or font cartridge for the print client.**

```
print-client# lpadmin -p printer-name -S hard-charset1[,hard-charset2...]
```

In this command, the variables are the same as those in Step 2 on page 124.

The print wheel or font cartridge definition is added in the print client's `/etc/lp/printers/`*printer-name*`/configuration` file.

5. **Repeat Step 3 on page 124 and Step 4 on page 124 for each print client that might need to use the print wheel or font cartridge.**

6. **Verify the information following the `Character sets` heading in the following output is correct on both the print server and the print client.**

```
$ lpstat -p printer-name -l
```

Example—Defining a Print Wheel

In the following example, the command defines the `pica` print wheel on the printer `luna` for a print client named `asteroid`.

```
asteroid# lpadmin -p luna -S pica
```

▼ How to Unmount and Mount a Print Wheel or Font Cartridge

1. **Log in as superuser or lp on the print server.**

2. **Unmount the print wheel or font cartridge that is in the printer by using the `lpadmin` command.**

```
# lpadmin -p printer-name -M -S none
```

–p *printer-name*	Printer on which you are unmounting a print wheel or font cartridge.
`-M -S` none	Specifies unmounting the current print wheel or font cartridge.

The current print wheel or font cartridge is deleted from the print server's `/etc/lp/printers/`*printer-name*`/configuration` file.

3. **Remove the print wheel or font cartridge from the printer.**

4. **Put the new print wheel or font cartridge in the printer.**

5. **Mount the new print wheel or font cartridge by using the** `lpadmin` **command.**

```
# lpadmin -p printer-name -M -S hard-charset
```

–p *printer-name*	Printer on which you are mounting a print wheel or font cartridge.
–M –S *hard-charset*	Hardware character set name of the print wheel or font cartridge you want to mount.

The print wheel or font cartridge is added in the print server's `/etc/lp/printers/`*printer-name*`/configuration` file. The mounted print wheel or font cartridge remains active until it is unmounted or until a new print wheel or font cartridge is mounted.

6. **Check the information under the** `Print wheels` **or** `Character set` **heading in the output of the following command. You should see the name of the print wheel or character set and the notation** `(mounted)`

```
$ lpstat -p printer-name -l
```

Example—Unmounting and Mounting a Print Wheel

In the following example, the commands unmount the current print wheel on the printer `luna` and mount the `pica` print wheel.

```
# lpadmin -p luna -M -S none
# lpadmin -p luna -M -S pica
```

▼ How to Set an Alert to Mount a Print Wheel or Font Cartridge

1. **Log in as superuser or lp on the print server.**

2. **Set an alert to mount a print wheel or font cartridge by using the** `lpadmin`**(1M) command.**

```
# lpadmin -S hard-charset -A alert [-Q requests] [-W minutes]
```

–S *hard-charset*	Hardware character set name of the print wheel or font cartridge for which you want to set an alert.
–A *alert*	Specifies what kind of alert will occur when a print wheel or font cartridge is requested. See Table 5–3 for detailed information about the valid values for *alert*. Some valid values are `mail`, `write`, and `quiet`. If you specify `mail` or `write`, a predefined alert message says to mount the specified print wheel or font cartridge and includes the names of one or more printers that have been set up to use such a print wheel or cartridge.
–Q *requests*	Specifies the number of print requests that require the print wheel or font cartridge that must be in the queue before an alert occurs. If you don't specify this option, only one print request in the queue triggers an alert.
–W *minutes*	Specifies how often (in minutes) the alert will occur. If you don't specify this option, the alert is sent only once.

The alert is added in the print server's `/etc/lp/pwheels/`*charset-name*`/alert.sh` file.

3. **Verify that the alert has been added for the print wheel or font cartridge by checking the output of the following command.**

```
# lpadmin -S hard-charset -A list
```

Otherwise, if you have set a low number of print requests to trigger the alert, submit enough print requests to meet the minimum requirement and make sure you receive an alert to mount the print wheel or font cartridge.

Examples—Setting an Alert to Mount a Print Wheel or Font Cartridge

In the following example, the command sets email alerts to occur every five minutes for the `elite` print wheel when there are ten print requests for `elite` in the print queue.

```
# lpadmin -S elite -A mail -Q 10 -W 5
```

In the following example, the command sets email alerts to occur every minute for the `finnish` font cartridge when there are five print requests for `finnish` in the print queue.

```
# lpadmin -S finnish -A mail -Q 5 -W 1
```

In the following example, the command sets console-window alerts to occur every 10 minutes for the `elite` print wheel when there are five print requests for `elite` in the print queue.

```
# lpadmin -S elite -A write -Q 5 -W 10
```

In the following example, the command sets no alerts to occur for the `elite` print wheel.

```
# lpadmin -S elite -A none
```

▼ How to Set Up an Alias for a Selectable Character Set

Note - You do not need to perform this procedure if the `terminfo`(4) names for the selectable character sets are adequate. See "Adding a `terminfo` Entry for an Unsupported Printer" on page 154 for more information on using the `terminfo` database.

---

1. **Log in as superuser or lp on the print server.**

2. **Display the names of the selectable character sets for the specified printer type by using the `tput`(1) command.**

```
# tput -T printer-type csn
```

–T *printer-type*	Printer type found in the `terminfo` database. See "Printer Type" on page 57 for information on entries in the `terminfo` database.
n	Number (0, 1, 2, 3, 4, 5, and so on) that represents a selectable character set for the specified printer type. The system displays the selectable character set name followed by the prompt symbol. For example, `cs1` could cause the system to display `english#`.

3. **Set up an alias for a selectable character set.**

```
# lpadmin -p printer-name -S select-charset1=alias1[,select-charset2=alias2...]
```

–p *printer-name*	Printer on which you are setting up aliases for selectable character sets.
–S *select-charset*	Selectable character set name for which to set an alias. The name can be found in Step 2 on page 128.
alias	Alias for the specified selectable character set. This alias can be used in addition to the selectable character set name. You can set up more than one alias with this command. Use commas or spaces to separate the aliases. If you use spaces, enclose the list of aliases in quotes.

The alias is added in the print server's `/etc/lp/printers/`*printer-name*`/configuration` file.

4. **Log in as superuser or lp on a system that is a print client of the print server.**

5. **Set up an alias for the selectable character set.**

```
# lpadmin -p printer-name -S select-charset1=alias1[,select-charset2=alias2...]
```

In this command, the variables are the same as those in Step 3 on page 128.

The alias is added in the print client's `/etc/lp/printers/`*printer-name*`/configuration` file.

6. **Repeat Step 4 on page 129 and Step 5 on page 129 for each print client that might need to use the alias.**

7. **Verify that the selectable character set alias is listed in the output of the following command on the print server and print clients.**

```
$ lpstat -p printer-name -l
```

Otherwise, submit a print request that uses the alias for the selectable character set and check for output.

Example—Setting Up an Alias for a Selectable Character Set

In the following example, the commands display the names of selectable character sets and specify `text` as an alias for the `usascii` selectable character set on the printer `luna`, which is an `ln03` printer type.

```
# tput -T ln03 cs0
usascii# tput -T ln03 cs1
english# tput -T ln03 csn2
```

```
finnish# tput -T ln03 csn3
japanese# tput -T ln03 cs4
norwegian#
# lpadmin -p luna -S usascii=text
```

Managing Print Filters

Print filters are programs that convert the content type of a file to a content type that is acceptable to the destination printer. The LP print service uses filters to:

- Convert a file from one data format to another so it can be printed properly on a specific type of printer
- Handle the special modes of printing, like two-sided printing, landscape printing, or draft- and letter-quality printing
- Detect printer faults and notify the LP print service of them so the print service can alert users and system administrators

Not every print filter can perform all these tasks. Because each task is printer-specific, the tasks can be implemented separately.

The LP print service provides the PostScript filters listed in Table 6–1. The filter programs are located in the `/usr/lib/lp/postscript` directory. For PostScript printing, you usually do not need to do anything beyond installing the filter programs when setting up a print server. Solaris Print Manager automatically enables the supplied filters. However, if you administer other printers, you might need to administer print filters for them.

Creating Print Filters

To create a new print filter, you must write a print filter program and create a print filter definition. Filters contain input types, output types, and complex options that provide a language to process command-line arguments within the filter. See “Creating a New Print Filter” on page 163 for background information and step-by-step instructions.

Adding, Changing, Removing, and Restoring Print Filters

Print filters are added, changed, or removed on the print server only.

You use the `lpfilter`(1M) command to manage the list of available filters. System information about filters is stored in the `/etc/lp/filter.table` file. The `lpfilter` command gets the information about filters to write to the table from filter descriptor files. The filter descriptor files supplied (PostScript only) are located in the `/etc/lp/fd` directory. The actual filter programs are located under `/usr/lib/lp`.

The LP print service imposes no fixed limit on the number of print filters you can define. You can remove filters that are no longer used to avoid extra processing by the LP print service. (LP examines all filters to find one that works for a specific print request.) If in doubt, do not remove a filter.

As you add, change, or delete filters, you can overwrite or remove some of the original filters provided by the LP print service. You can restore the original set of filters, if necessary, and remove any filters you have added.

SunOS software provides a default set of PostScript filters, which Solaris Print Manager automatically adds to a print server. Some of the TranScript filters used with SunOS 4.1 have SunOS equivalents, but others do not. The table below lists the default PostScript filters and identifies the TranScript filters, where applicable.

TABLE 6–1 Default PostScript Filters

Filter	Action	TranScript Equivalent
`download`	Download fonts	
`dpost`	`ditroff` to PostScript	`psdit`
`postdaisy`	`daisy` to PostScript	
`postdmd`	`dmd` to PostScript	
`postio`	Serial interface for PostScript printer	`pscomm`
`postior`	Communicate with printer	
`postmd`	Matrix gray scales to PostScript	
`postplot`	`plot` to PostScript	`psplot`

TABLE 6–1 Default PostScript Filters *(continued)*

Filter	Action	TranScript Equivalent
`postprint`	`simple` to PostScript	`enscript`
`postreverse`	Reverse or select pages	`psrev`
`posttek`	`TEK4014` to PostScript	`ps4014`

The SunOS software does *not* provide the following filters:

- `TEX`
- `oscat` (NeWSprint `opost`)
- Enscript

The `postreverse`, `postprint`, `postio`, and `dpost` filters are provided in place of Enscript.

Solaris Print Manager adds the default PostScript filters to a print server. If you have printing needs that are not met by these filters, see "How to Create a New Print Filter" on page 172 for information about writing a custom print filter.

▼ How to Add a Print Filter

1. **Log in as superuser or lp on the print server.**

2. **Add a print filter that is based on a print filter definition by using the `lpfilter` command.**

```
# lpfilter -f filter-name -F filter-def
```

–f *filter-name*	Name you choose for the print filter.
–F *filter-def*	Name of the print filter definition.

The print filter is added in the print server's `/etc/lp/filter.table` file.

3. **Verify that the print filter was added by checking for information about the print filter in the output of the following command.**

```
# lpfilter -f filter-name -l
```

Example—Adding a Print Filter

In the following example, the command adds the `daisytroff` print filter that has the `daisytroff.fd` print filter definition.

```
# lpfilter -f daisytroff -F /etc/lp/fd/daisytroff.fd
```

▼ How to Delete a Print Filter

1. **Log in as superuser or lp on the print server.**

2. **Delete the print filter by using the `lpfilter` command.**

```
# lpfilter -f filter-name -x
```

–f *filter-name*	Name of the print filter to be deleted.
–x	Deletes the specified filter.

The print filter is deleted from the print server's `/etc/lp/filter.table` file.

3. **Verify that filter was deleted by using the following command. You should receive an error indicating that no filter by the specified name exists.**

```
# lpfilter -f filter-name -l
```

Example—Deleting a Print Filter

In the following example, the command deletes the `daisytroff` print filter.

```
# lpfilter -f daisytroff -x
```

▼ How to View Information About a Print Filter

1. **Log in as superuser or lp on the print server.**

2. **Request information about a print filter by using the `lpfilter` command.**

```
# lpfilter -f filter-name -l
```

–f *filter-name*	Print filter for which you want to view information. Specify `all` for *filter-name* to view information about all the available print filters.
–l	Displays information about the specified filter.

Information about the specified print filter(s) is displayed.

Examples—Viewing Information About a Print Filter

In the following example, the command requests information for the `postdaisy` print filter, and the information that is displayed in response.

```
# lpfilter -f postdaisy -l
Input types: daisy
Output types: postscript
Printer types: any
Printers: any
Filter type: slow
Command: /usr/lib/lp/postscript/postdaisy
Options: PAGES * = -o*
Options: COPIES * = -c*
Options: MODES group = -n2
Options: MODES group\=\([2-9]\) = -n\1
Options: MODES portrait = -pp
Options: MODES landscape = -pl
Options: MODES x\=\(\-*[\.0-9]*\) = -x\1
Options: MODES y\=\(\-*[\.0-9]*\) = -y\1
Options: MODES magnify\=\([\.0-9]*\) = -m\1
```

In the following example, the command redirects information about the `daisytroff` filter to a file (creates the filter definition for that filter). This is useful if a filter definition is removed unintentionally.

```
# lpfilter -f daisytroff -l > daisytroff.fd
```

In the following example, the command displays all the print filters that have been added to the system, and the information that is displayed in response.

```
# lpfilter -f all -l | grep Filter
(Filter "download")
Filter type: fast
(Filter "postio")
Filter type: fast
(Filter "postior")
Filter type: fast
```

(continued)

```
(Filter "postreverse")
Filter type: slow
```

Managing Forms

A *form* is a sheet of paper on which information is printed in a predetermined format. Unlike plain paper stock, forms usually have text or graphics preprinted on them. Common examples of forms are company letterhead, invoices, blank checks, receipts, and labels.

The term *form* has two meanings: the physical medium (the paper) and the software that defines a form to the LP print service.

The LP print service allows you to control the use of forms. This section provides information about adding, changing, removing, mounting, and controlling access to forms.

Adding, Changing, or Deleting Forms

When you add a form, you tell the LP print service to include the form in its list of available forms. You also have to supply the information required to describe or define the form. Although you can enter such definitions when you add the form, it helps to create the definitions first and save them in files. You can then change the form definition by editing the file. See the table below for information about how to create form definitions.

Note - No form definitions are supplied with the LP print service.

To change a form, you must re-add the form with a different definition.

The LP print service imposes no limit on the number of forms you can define. However, you should delete forms that are no longer appropriate. Obsolete forms can result in unnecessary processing by the print service.

Mounting Forms

To print a form, you must load the paper in the printer and use a command to *mount* the form, which notifies the LP print service that print requests submitted to the printer are to be printed using the form definition. If you use one printer for different types of printing, including forms, you should:

- Disable the printer before you load the paper and mount the form.
- Re-enable the printer when the form is ready; otherwise, the LP print service will continue to print files that do not need the form on the printer.

When you mount a form, make sure it is aligned properly. If an alignment pattern has been defined for the form, you can request that the pattern print repeatedly after you have mounted the form, until you have adjusted the printer so the alignment is correct.

When you want to change or discontinue using a form on a printer, you must notify the LP print service by unmounting the form.

Tracking Forms

The LP print service helps you track which forms are mounted on each printer and notifies you when it cannot find a description it needs to print a form. You are responsible for creating form descriptions and mounting and unmounting form paper in each printer, either as part of setting up a printer or in response to alerts from the LP print service.

Users can specify the form on which they want a job to print. As root, you can mount a specific form, then tell the LP print service that the form is available and on which printer it is mounted. Users can submit print requests specifying a particular form. When the LP print service receives the request, it sends an alert message to root requesting that you mount the form.

Defining Alerts for Mounting Forms

You request alerts for mounting forms in the same way you request other alerts from the LP print service. See “Setting Up Printer Fault Alerts” on page 100 for general information about alerts.

Checking Forms

When you have defined a form for the LP print service, you can check it with either of two commands, depending on the type of information you want to check.

- Show the attributes of the form by using the `lpforms`(1M) command. You can also redirect the output of the command into a file to save it for future reference.
- Display the current status of the form by using the `lpstat` command. To protect potentially sensitive content, the alignment pattern is not shown.

If you are not sure about the name of an existing form, you can list the contents of the `/etc/lp/forms` directory to see the names of the forms there.

Limiting Access to Forms

You can control which printers and users have access to some or all of the forms available on the network. For example, you might want only the people in the payroll or accounts payable department to be able to print check forms. In addition, you might want the check forms to be available only on certain printers.

To limit user access to forms, see "How to Limit User Access to a Form" on page 143. To limit printer access to a form, see "How to Limit Printer Access to a Form" on page 144.

▼ How to Add a Form

1. **Log in as superuser or lp on the print server.**

2. **Add a form that is based on a form definition by using the `lpforms` command.**

```
# lpforms -f form-name -F /etc/lp/forms/form
```

–f *form-name*	Name you choose for the form.
–F /etc/lp/forms/*form*	Name of the form definition.

   The form is added in the print server's `/etc/lp/forms/`*form-name*`/describe` file.

3. **Verify that the form was added by checking for a listing of information about the form in the output of the following command.**

```
# lpforms -f form-name -l
```

Example—Adding a Form

In the following example, the command adds the `medical` form that uses the `medical.fmd` form definition.

```
# lpforms -f medical -F /etc/lp/forms/medical.fmd
```

Note - Before the form can be used, one or more printers must be given access to the form. See "How to Limit Printer Access to a Form" on page 144.

▼ How to Delete a Form

1. **Log in as superuser or lp on the print server.**

2. **Delete the form by using the `lpforms` command.**

```
# lpforms -f form-name -x
```

–f *form-name*	Form to be deleted.
–x	Deletes the specified form.

The form is deleted from `/etc/lp/forms/`*form-name* file.

3. **Verify that form was deleted by using the following command. You should receive an error indicating that a form by the specified name does not exist.**

```
# lpforms -f form-name -l
```

Example—Deleting a Form

In the following example, the command deletes the `medical` form.

```
# lpforms -f medical -x
```

▼ How to Unmount and Mount a Form

1. **Log in as superuser or lp on the print server.**

2. **Stop accepting print requests on the printer on which you are unmounting the current form by using the `reject` command.**

```
# reject printer-name
```

printer-name	Name of the printer on which you are unmounting a form.

New print requests (which might not require the form) are not allowed to enter the printer's queue.

3. **Unmount the current form by using the `lpadmin` command.**

```
# lpadmin -p printer-name -M -f none
```

In this command, the variable *printer-name* is the same as in Step 2 on page 139.

The current form is deleted from the print server's `/etc/lp/printers/`*printer-name*`/configuration` file.

4. **Remove the form paper from the printer.**

5. **Load the form paper for the next print request.**

6. **Mount the form by using the `lpadmin` command.**

```
# lpadmin -p printer-name -M -f form-name[-a -o filebreak]
```

–p *printer-name*	Printer on which you are mounting a form.
–M –f *form-name*	Name of the form to be mounted.
–a –o `filebreak`	Optionally enables you to print a copy of the alignment pattern defined for the form, if it has one.

The specified form is added in the print server's `/etc/lp/printers/`*printer-name*`/configuration` file.

7. **Start accepting print requests on the printer.**

```
# accept printer-name
```

The printer is ready to print the form you just mounted.

8. **Verify that the form has been mounted by checking for the form name under the `Form mounted` heading in the output of the following command.**

```
$ lpstat -p printer-name -l
```

Otherwise, submit a print request that requires the new form and check the printer for output.

Examples—Unmounting and Mounting a Form

The following example shows the process of unmounting the currently mounted form on the printer `luna`.

```
# reject luna
destination "luna" will no longer accept requests
# lpadmin -p luna -M f none
# accept luna
destination "luna" now accepting requests
```

The following example shows the process of mounting the `medical` form on the printer `luna`.

```
# reject luna
destination "luna" will no longer accept requests
# lpadmin -p luna -M f medical -a -o filebreak
# accept luna
destination "luna" now accepting requests
```

▼ How to Set an Alert to Mount a Form

1. **Log in as superuser or lp on the print server.**

2. **Set a request alert for mounting a form by using the `lpadmin` command.**

```
# lpforms -f form-name -A alert [-Q requests] [-W minutes]
```

–f *form-name*	Form for which you want to set a request alert.
–A *alert*	Specifies what kind of alert will occur when a form is requested. See Table 5–3 for detailed information about the valid values for *alert*. Some valid values are `mail`, `write`, and `quiet`. If you choose `mail` or `write`, a predefined alert message says to mount the specified form and includes the names of one or more printers that have been set up to use the form.
–Q *requests*	Specifies how many print requests that require the form must be in the queue to trigger an alert. If you don't specify this option, an alert occurs with just one print request in the queue.
–W *minutes*	Specifies how often (in minutes) the alert will occur. If you don't specify this option, the alert is sent once.

The request alert is added in the print server's `/etc/lp/forms/`*form-name*`/alert.sh` file.

3. **Verify that the alert has been added for the form by checking the output of the following command.**

```
# lpforms -f form-name -A list
```

Otherwise, if you have set a low number of print requests to trigger the alert, submit print requests to meet the minimum requirement and make sure you receive an alert to mount the form.

Examples—Setting an Alert to Mount a Form

In the following example, the command sets email alerts to occur every five minutes for the `letterhead` form when there are 10 print requests for `letterhead` in the print queue.

```
# lpforms -f letterhead -A mail -Q 10 -W 5
```

In the following example, the command sets console window alerts to occur every 10 minutes for the `letterhead` form when there are five requests for `letterhead` in the print queue.

```
# lpforms -f letterhead -A write -Q 5 -W 10
```

In the following example, the command sets no request alerts for the `invoice` form.

```
# lpforms -f invoice -A none
```

▼ How to View Information About a Form

1. **Log in as superuser or lp on the print server.**

2. **Request information about a form by using the `lpforms` command.**

```
# lpforms -f form-name -l
```

–f *form-name*	Form for which you want to view information. Specify `all` for *form-name* to view information about all the available forms.
–l	Lists the specified form.

Information about the specified form(s) is displayed.

Examples—Viewing Information About a Form

In the following example, the command displays information about the `medical` form.

```
# lpforms -f medical -l
Page length: 62
Page width: 72
Number of pages: 2
Line pitch: 6
Character pitch: 12
Character set choice: pica
Ribbon color: black
Comment:
Medical claim form
```

In the following example, the command redirects the information about the `medical` form to a file. (This command creates the form definition for the form.) This is useful if a form definition gets removed unintentionally.

```
# lpforms -f medical -l > medical.fmd
```

▼ How to View the Current Status of a Form

1. **Log in on the print server.**

2. **Request information about the current status of a form by using the** `lpstat`**(1) command.**

```
$ lpstat -f form-name
```

–f *form-name*	Form for which you want to view the current status. Specify `all` for *form-name* to view the current status of all the forms.

Information about the current status of the specified form(s) is displayed.

Example—Viewing the Current Status of a Form

In the following example, the command displays the status of the `medical` form.

```
$ lpstat -f medical
form medical is available to you
```

▼ How to Limit User Access to a Form

1. **Log in as superuser or lp on the print server.**

2. **Allow or deny users access to a form by using the** `lpforms` **command.**

```
# lpforms -f form-name -u allow:user-list | deny:user-list
```

–f *form-name*	Name of the form for which the allow or deny user access list is being created.
–u allow:*user-list*	Represents users to be added to the allow access list. Use a comma or a space to separate users' login IDs. If you use spaces, enclose the list of IDs in quotes. Table 5–5 provides the valid values for *user-list*.
deny:*user-list*	Represents users to be added to the deny user access list. Use a comma or a space to separate users' login IDs. If you use spaces, enclose the list of IDs in quotes. Table 5–5 provides the valid values for *user-list*.

The specified user(s) are added to the allow or deny user access list for the specified form in one of the following files on the print server:

`/etc/lp/forms/`*form-name*`/allow` or

`/etc/lp/forms/`*form-name*`/deny`

3. Verify the allow and deny user access lists by using the `lpforms` command.

```
# lpforms -f form-name -l
```

Examples—Limiting User Access to a Form

In the following example, the command allows only the users `nathan` and `marcia` access to the `check` form.

```
# lpforms -f check -u allow:nathan,marcia
```

In the following example, the command denies users `jones` and `smith` access to the `dental` form.

```
# lpforms -f dental -u deny:"jones,smith"
```

▼ How to Limit Printer Access to a Form

1. Log in as superuser or lp on the print server.

2. Allow or deny use of forms on a printer by using the `lpadmin` command.

```
# lpadmin -p printer-name -f allow:form-list | deny:form-list
```

–p *printer-name*	Name of the printer for which the allow or deny forms list is being created.
–f allow:*form-list* | deny:*form-list*	Form names to be added to the allow or deny list. Use a space or a comma to separate multiple form names. If you use spaces to separate form names, enclose the list of form names in quotes.

The specified form(s) are added to the allow or deny forms list in one of the following files on the print server:

`/etc/lp/printers/`*printer-name*`/form.allow`

`/etc/lp/printers/`*printer-name*`/form.deny`

3. Verify the allow and deny forms lists by using the following command.

```
# lpstat -p printer-name -l
```

Examples—Limiting Printer Access to a Form

In the following example, the command allows the printer `luna` to access only the `medical`, `dental`, and `check` forms.

```
# lpadmin -p luna -f allow:medical,dental,check
```

In the following example, the command denies the printer `luna` from accessing the `medical`, `dental`, and `check` forms.

```
# lpadmin -p luna -f deny:"medical dental check"
```

Managing Fonts

If you have a laser printer, you might need to install and maintain PostScript fonts. You might also have to decide where to install PostScript fonts and how to manage them. For many printers, the fonts are set up as part of the printer installation process.

PostScript fonts are stored in outline form, either on the printer or on a system that communicates with the printer. When a document is printed, the PostScript interpreter generates each character as needed (in the appropriate size) from the outline description of it. If a font required for a document is not stored on the printer

being used, it must be transmitted to that printer before the document can be printed. This transmission process is called *downloading fonts.*

Fonts are stored and accessed in several ways:

- *Printer-resident fonts* are stored permanently on a printer. These fonts are installed in read-only memory (ROM) on the printer by the manufacturer. If the printer has a disk, you can install fonts on that disk. Most PostScript printers are shipped with 35 standard fonts.
- A *permanently downloaded font* is transmitted to a printer with a PostScript `exitserver` program. A permanently downloaded font remains in printer memory until the printer is turned off. Memory allocated to a downloaded font reduces the memory available on the server for PostScript print requests. Use of an `exitserver` program requires the printer system password and can be reserved for the printer administrator. You should permanently download a font if most print requests serviced by the printer use that font.
- Fonts that are used infrequently or for special purposes can be stored on a user's system. The user can specify these fonts when submitting the print request. The fonts are appended to the print request and transmitted to the printer. When the print request is processed, the space allocated for the font is freed for other print requests.
- Host-resident *fonts* are stored on a system shared by many users. The system that stores the fonts can be a print server or a print client. Each user can request fonts in the document to be printed. This method is useful when there are numerous available fonts, or when these fonts are not used by all print requests. If the fonts will be used only on printers attached to a print server, they should be stored on the print server. If the fonts are to be used by the users on one system and the users can submit requests to multiple printers on a network, the fonts should be stored on the users' system.

  The LP print service provides a special download filter to manage host-resident fonts. It also supplies `troff` width tables for the 35 standard PostScript fonts which reside on many PostScript printers, for use by the `troff`(1) program.

Managing Printer-Resident Fonts

Most PostScript printers come equipped with fonts resident in the printer ROM. Some printers have a disk on which additional fonts are stored. When a printer is installed, you should add the list of printer-resident fonts to the font list for that printer. By identifying printer-resident fonts, you prevent fonts from being transmitted unnecessarily across a network. Each printer has its own list of resident fonts, which is contained in the file:

```
/etc/lp/printers/printer-name/residentfonts
```

When the printer is attached to a print server, make sure the list in the `residentfonts` file includes fonts that are on the print server and which are available for downloading to the printer.

You must edit the files containing the list of printer-resident fonts by using a text editor such as `vi`.

Downloading Host-Resident Fonts

When a PostScript document contains a request for fonts not loaded on the printer, the download filter manages this request. The download filter uses PostScript document structuring conventions to determine which fonts to download.

LP print filters are either fast or slow. A fast filter quickly prepares a file for printing, and it must have access to the printer while the filter is processing. A slow filter takes longer to convert a file, and it does not need to access the printer while the filter is processing. An example of a slow filter is ASCII to PostScript.

The download filter is a fast filter; it downloads fonts automatically if the fonts are on the print server. The download filter can also be used to send fonts to a print server. To do this, you can create a new filter table entry that calls the download filter as a slow filter by using the `lp -y` command. Alternatively, you can force selection of this filter by changing the input type.

The download filter performs five tasks:

1. It searches the PostScript document to determine which fonts are requested. These requests are documented with the following PostScript structuring comments: `%%DocumentFonts:` *font1 font2* ... in the header comments.
2. It searches the list of printer-resident fonts to determine if the requested font must be downloaded.
3. If the font is not resident on the printer, the download filter searches the host-resident font directory (by getting the appropriate file name from the map table) to determine if the requested font is available.
4. If the font is available, the filter takes the file for that font and appends it to the file to be printed.
5. It sends the font definition file and the source file (the file to be printed) to the PostScript printer.

Installing and Maintaining Host-Resident Fonts

Some fonts reside on the host system and are transmitted to the printer as needed for particular print requests. As the administrator, you make PostScript fonts available to all users on a system. To do so, you must know how and where to install these fonts. Because fonts are requested by name and stored in files, the LP print service keeps a map file that shows the correspondence between the names of fonts and the names

of the files containing those fonts. Both the map and the font list must be updated when you install host-resident fonts.

The fonts available for use with PostScript printers are stored in directories you create called `/usr/share/lib/hostfontdir/`*typeface*`/`*font*, where *typeface* is replaced by a name like `palatino` or `helvetica`, and *font* is replaced by a name like `bold` or `italic`.

▼ How to Install Downloaded PostScript Fonts

1. **Log in as superuser or lp on the print server or print client.**

2. **Change directory to the `/etc/lp/printers/`*printer-name* directory.**

```
# cd /etc/lp/printers/printer-name
```

printer-name	Name of the printer on which you want to install downloaded PostScript fonts.

3. **Create the `residentfonts` file, if it does not already exist.**

```
# touch residentfonts
```

This file might not exist if this is the first time you are adding permanently downloaded fonts.

4. **Edit the `residentfonts` file and add all the printer-resident fonts and fonts to be permanently downloaded.**

▼ How to Install Host-Resident PostScript Fonts

1. **Log in as superuser or lp on the print server or print client.**

2. **Create the `hostfontdir` directory, if it does not already exist.**

```
# cd /usr/share/lib
# mkdir hostfontdir
```

(continued)

```
# chmod 775 hostfontdir
```

3. **Create a directory for a new typeface, if the directory does not already exist.**

```
# mkdir typeface
```

4. **Copy the font file to the appropriate directory.**

```
# cp filename /usr/share/lib/hostfontdir/typeface/font
```

5. **Add the name of the font and the name of the file in which it resides to the map table.**

   a. **Change to the** `/usr/share/lib/hostfontdir` **directory.**

   b. **Edit the** `map` **file using a text editor such as vi.**

   Add a one-line entry for each font you want to add to the table, with the font name first, followed by a space, followed by the name of the file where the font resides. For example:

```
Palatino-Bold /usr/share/lib/hostfontdir/palatino/bold
```

   c. **Save the file.**

   When an example entry exists in the map table on the appropriate system, users will be able to apply the font (for example, Palatino Bold) in their print jobs. When they submit a print request containing this font, the LP print service appends a copy of the file `/usr/share/lib/hostfontdir/palatino/bold` to that file before sending it to the printer.

6. **If you are using** `troff`**, you must create new width tables for this font in the standard** `troff` **font directory.**

CHAPTER 7

Customizing the LP Print Service (Tasks)

This chapter provides background information and procedures for customizing the LP print service.

This is a list of the step-by-step instructions in this chapter.

- "How to Adjust the Printer Port Characteristics" on page 153
- "How to Add a `terminfo` Entry for an Unsupported Printer" on page 157
- "How to Set Up a Custom Printer Interface Program" on page 161
- "How to Create a New Print Filter" on page 172
- "How to Create a New Form Definition" on page 177

For overview information about printers, see Chapter 2.

Adjusting Printer Port Characteristics

The printer port characteristics set by the LP print service must be compatible with the printer communication settings. If the default printer port settings provided by the LP print service do not work with a printer, refer to the printer manual from the manufacturer to find out what settings the printer requires from the LP print service. Use the `stty` command to set and display printer communication settings.

The table below shows the default `stty` settings used by the LP print service.

TABLE 7–1 `stty` Default Settings Used by the LP Print Service

Option	Meaning
`–9600`	Set baud to 9600
`–cs8`	Set 8-bit bytes
`–cstopb`	Send one stop bit per byte
`–parity`	Do not generate parity
`–ixon`	Enable XON/XOFF (also known as START/STOP or DC1/DC3)
`–opost`	Do “output post-processing” using all the settings that follow in this table
`–olcuc`	Do not map lowercase to uppercase
`–onlcr`	Change line feed to carriage return/line feed
`–ocrnl`	Do not change carriage returns into line feeds
`–onocr`	Output carriage returns even at column 0
`–n10`	No delay after line feeds
`–cr0`	No delay after carriage returns
`–tab0`	No delay after tabs
`–bs0`	No delay after backspaces
`–vt0`	No delay after vertical tabs
`–ff0`	No delay after form feeds

▼ How to Adjust the Printer Port Characteristics

1. **Log in as superuser or lp on the print server.**

2. **Adjust the printer port characteristics by using the `lpadmin` command.**

```
# lpadmin -p printer-name -o "stty=options"
```

–p *printer-name*	Name of the printer for which you are adjusting the port characteristics.
–o "`stty=`*options*"	Sets the port characteristic (`stty` option) specified by *options*.You can change more than one `stty` option setting with this command. Enclose each option in single quotation marks and use a space to separate the options. See `stty`(1) for a complete list of options. Table 7–1 shows the default `stty` settings used by the LP print service.

3. **Verify that the printer port characteristics have been changed by using the following command.**

```
# stty -a
```

Examples—Adjusting the Printer Port Characteristics

In the following example, the command sets the port characteristics for the printer `luna`. The `parenb` option enables parity checking/generation, `parodd` sets odd parity generation, and `cs7` sets the character size to 7 bits.

```
# lpadmin -p luna -o "stty='parenb parodd cs7'"
```

In the following example, the command sets the terminal baud rate to 19200 for the printer `venus`.

```
# lpadmin -p venus -o "stty=19200"
```

Adding a terminfo Entry for an Unsupported Printer

The LP print service uses an interface program and the terminfo database to initialize each printer and establish a selected page size, character pitch, line pitch, and character set.

Each printer is identified in the terminfo database with a short name. The name required by the terminfo database is identical to the name used to set the TERM shell variable. This name is also the printer type you specify when setting up a printer. For example, the entries for different types of PostScript printers are in /usr/share/lib/terminfo/P. The default entries provided with the SunOS release are PS (for PostScript) and PSR (for PostScript Reverse).

If you cannot find a terminfo entry for your printer, you still might be able to use the printer with the LP print service without the automatic selection of page size, pitch, and character sets. However, you might have trouble keeping the printer set in the correct modes for each print request.

If there is no terminfo entry for your type of printer and you want to keep the printer set in the correct modes, you can either customize the interface program used with the printer or add an entry to the terminfo database. A terminal or printer entry in the terminfo database contains and defines hundreds of items. The LP print service, however, uses fewer than 50 of these items. The table below lists the required terminfo items for a printer.

TABLE 7–2 Required terminfo Items for a Printer

Item		Meaning
Booleans:		
	cpix	Changing character pitch changes resolution
	daisy	Printer requires an operator to change character set
	lpix	Changing line pitch changes resolution
Numbers:		
	bufsx	Number of bytes buffered before printing
	cols	Number of columns in a line

TABLE 7–2 Required `terminfo` Items for a Printer *(continued)*

Item		Meaning
	`cps`	Average print rate in characters per second
	`it`	Tabs initially every *n* spaces
	`lines`	Number of lines on a page
	`orc`	Horizontal resolution, in units per character
	`orhi`	Horizontal resolution, in units per inch
	`orl`	Vertical resolution, in units per line
	`orvi`	Vertical resolution, in units per inch
Strings:		
	`chr`	Change horizontal resolution
	`cpi`	Change number of characters per inch
	`cr`	Carriage return
	`csnm`	List of character set names
	`cudl`	Down one line
	`cud`	Move carriage down *n* lines
	`cuf`	Move carriage right *n* columns
	`cvr`	Change vertical resolution
	`ff`	Page eject
	`hpa`	Horizontal position absolute
	`ht`	Tab to next 8-space tab stop

TABLE 7–2 Required `terminfo` Items for a Printer *(continued)*

Item		Meaning
	`if`	Name of initialization file
	`iprog`	Path name of initialization program
	`is1`	Printer initialization string
	`is2`	Printer initialization string
	`is3`	Printer initialization string
Strings:		
	`lpi`	Change number of lines per inch
	`mgc`	Clear all margins (top, bottom, and sides)
	`rep`	Repeat a character *n* times
	`rwidm`	Disable double-wide printing
	`scs`	Select character set
	`scsd`	Start definition of a character set
	`slines`	Set page length to *n* lines per page
	`smgl`	Set left margin at current column
	`smglp`	Set left margin
	`smgr`	Set right margin at current column
	`smgrp`	Set right margin
	`smglr`	Set both left and right margins
	`msgt`	Set top margin at current line

TABLE 7–2 Required `terminfo` Items for a Printer *(continued)*

Item	Meaning
`smgtp`	Set top margin
`smgb`	Set bottom margin at current line
`smgbp`	Set bottom margin
`smgtb`	Set both top and bottom margins
`swidm`	Enable double-wide printing
`vpa`	Vertical position absolute

▼ How to Add a `terminfo` Entry for an Unsupported Printer

Note - Before you create a `terminfo` entry for a printer, you should first make sure none of the existing `terminfo` entries will support the printer. To do so, try to set up the printer with an entry for a similar printer, if there is one.

1. **Log in as superuser or lp on the print server.**

2. **Determine a `terminfo` entry name for the printer.**

   The directories in the `/usr/share/lib/terminfo` directory contain all the valid `terminfo` entries. Use them as a guide for choosing a name for the printer.

3. **Create a `terminfo` entry file for the printer.**

   Table 7–2 shows the items you must define in the `terminfo` entry to add a new printer to the LP print service. For more details about the structure of the `terminfo` database, see `terminfo`(4).

   To help you start writing a new `terminfo` entry, use the `infocmp` command to save an existing `terminfo` entry to a file. This is helpful if there is a `terminfo` entry that is similar to one you want to create. For example, the following command saves the `ps` entry to the `ps_cust` file, which will become the new `terminfo` entry.

   ```
   infocmp ps > ps_cust
   ```

4. **Compile the** `terminfo` **entry file into the** `terminfo` **database.**

```
# tic terminfo_entry
```

terminfo_entry — The `terminfo` entry file you created.

5. **Check for the new** `terminfo` **entry file in the** `/usr/share/lib/terminfo` **directory.**

Customizing the Printer Interface Program

If you have a printer that is not supported by the standard printer interface program, you can furnish your own printer interface program. You can copy the standard program and then tell the LP print service to use it for a specified printer. But first you need to understand what is in the standard program. The following section describes the standard program.

A printer interface program should:

- Initialize the printer port, if necessary. The standard printer interface program uses the `stty` command to initialize the printer port.
- Initialize the printer hardware. The standard printer interface program gets the control sequences from the `terminfo` database and the `TERM` shell variable.
- Print a banner page, if necessary.
- Print the number of copies specified by the print request.

Caution - If you have a printer interface program from a release of UNIX System V prior to Release 3.2, it will probably work with the SunOS 5.8 or compatible LP print service. However, several –o options have been standardized in the SunOS 5.8 or compatible LP print service and will be passed to every printer interface program. These options might interfere with similarly named options used by the old interface.

The LP print service, not a printer interface program, is responsible for opening the printer port. The printer port is given to the printer interface program as standard output, and the printer is identified as the “controlling terminal” for the printer

interface program so that a “hang-up” of the port will cause a SIGHUP signal to be sent to the printer interface program.

The Standard Printer Interface Program

The standard (model) printer interface program, `/usr/lib/lp/model/standard`, is used by the LP print service to set the printing defaults shown in Table 7–3.

TABLE 7–3 Default Printer Port Characteristics

Characteristic	Default Setting
Default filter	`None`
Character pitch	`None`
Line pitch	`None`
Page width	`None`
Page length	`None`
Character set	`None`
`stty` options	`9600 cs8 -cstopb -parenb -parodd ixon -ixany opost -olcuc onlcr -ocrnl -onocr -onlret -ofill nl0 cr0 tab0 bs0 vt0 ff0`
Exit code	`0`

Customizing `stty` Modes

If you need to change the terminal characteristics, like baud rate or output options, look for the section of the standard printer interface program that begins with the following comment:

```
## Initialize the printer port
```

Exit Codes

When printing is complete, your interface program should exit with a code that shows the status of the print job. The exit code is the last entry in the printer interface program.

The table below shows the exit codes and how they are interpreted by the LP print service.

TABLE 7–4 Printer Interface Program Exit Codes

Code	Meaning to the LP Print Service
`0`	The print request has been successfully completed. If a printer fault occurred, it has been cleared.
`1` to `127`	A problem was encountered when printing a request (for example, too many nonprintable characters or the request exceeds the printer capabilities). The LP print service notifies the person who submitted the request that there was an error when printing it. This error will not affect future print requests. If a printer fault has occurred, it has been cleared.
`128`	This code is reserved for internal use by the LP print service. Interface programs must not exit with this code.
`129`	A printer fault was encountered when printing the request. This fault will affect future print requests. If the fault recovery for the printer directs the LP print service to wait for the administrator to correct the problem, the LP print service disables the printer. If the fault recovery is to continue printing, the LP print service will not disable the printer, but it will try printing again in a few minutes.
`>129`	These codes are reserved for internal use by the LP print service. Interface programs must not exit with codes in this range.

If the program exits with a code of 129, root is alerted of a printer fault. The LP print service must also reprint the request from the beginning, after the fault has been cleared. If you do not want the entire request to be reprinted, you can have the interface program send a fault message to the LP print service, but wait for the fault to be cleared. When the fault is cleared, the interface program can resume printing the file. When printing is finished, the printer interface program can give a zero exit code, just as if the fault had never occurred. An added advantage of this approach is that the interface program can detect when the fault is cleared automatically, so that the administrator does not need to re-enable the printer.

Fault Messages

You can use the `lp.tell` program to send fault messages to the LP print service. This program is referenced by the `LPTELL` shell variable in the standard printer interface code. The program takes standard input and sends it to the LP print service, where it is put into the message that alerts the administrator to the printer fault. If its standard input is empty, `lp.tell` does not initiate an alert. For an example of how the `lp.tell` program is used, examine the standard printer interface code immediately after the following comment:

```
# Set up the $LPTELL program to capture fault messages here
```

If you use the special exit code `129` or the `lp.tell` program, the printer interface program does not need to disable the printer itself. The interface program can disable the printer directly, but doing so will override the fault-alerting mechanism. Alerts are sent only if the LP print service detects that the printer has a fault, and the special exit code and the `lp.tell` program are its main detection tools.

If the LP print service has to interrupt printing of a file at any time, it kills the interface program with a signal TERM (trap number 15). (See `kill`(1) and `signal`(3C).) If the printer interface program dies from receipt of any other signal, the LP print service assumes that future print requests will not be affected, and continues to use the printer. The LP print service notifies the user who submitted the request that the request has not been finished successfully.

When the interface is first invoked, the signals HUP, INT, QUIT, and PIPE (trap numbers 1, 2, 3, and 13) are ignored. The standard interface changes this so the signals are trapped at appropriate times. The standard interface interprets receipt of these signals as warnings that the printer has a problem; when it receives a signal, it issues a fault alert.

Using a Customized Printer Interface Program

You can create a customized printer interface program and use it in place of the standard printer interface program on the print server. To do so, you use the `lpadmin` command to register the program with the LP print service for a specific printer.

▼ How to Set Up a Custom Printer Interface Program

1. **Log in as superuser or lp on the print server.**

2. **Determine your next step based on whether you have a custom printer interface program.**

If You ...	Then ...
Need to create a custom printer interface program	Go to Step 3 on page 162.
Already have a custom printer interface program	Go to Step 5 on page 162.

3. **Copy the standard printer interface program.**

```
# cp /var/spool/lp/model/standard custom-interface
```

4. **Change the copy of the standard printer interface program to meet your needs.**
   Refer to the description of the program in "The Standard Printer Interface Program" on page 159 to determine what you need to change.

5. **Set up the custom printer interface program for a specific printer.**

```
# lpadmin -p printer-name -i custom-interface
```

–p *printer-name*	The printer that will use the custom printer interface program.
–i *custom-interface*	Name of the custom printer interface program.

   The custom printer interface program is registered with the LP print service, and will be used by that printer when users submit print requests.

6. **Verify that the custom printer interface program has been added in the `/etc/lp/printers/`*printer-name*`/configuration` file.**

Examples—Setting Up a Custom Printer Interface Program

In the following example, the command sets up a custom printer interface program named `custom` for the printer `luna`.

```
# lpadmin -p luna -i custom
```

In the following example, the command sets up a custom printer interface program that the system `venus` is using on the printer `asteroid`.

```
# lpadmin -p asteroid -e venus
```

Creating a New Print Filter

A filter is used by the LP print service each time it has to print a type of file that the printer cannot interpret. Creating a new print filter is not easy; it usually requires extensive experimentation. The process of defining a new print filter consists of two steps:

- Writing a print filter program
- Creating a print filter definition

A print filter can be as simple or as complex as needed. Filters contain input types, output types, and complex options that provide a language to process command-line arguments within the filter.

If you have non-PostScript printers, you have to create and add print filters as required. First, you need to understand what print filters are and the requirements that must be met by a filter program.

Writing a Print Filter Program

The LP print service provides filter programs in the `/usr/lib/lp/postscript` directory. These filters cover most PostScript printing situations—where the destination printer requires the data to be in PostScript format. A print filter program must be a binary executable.

Types of Filters

There are two types of print filters: fast filters and slow filters.

Fast filters do not require much processing time to prepare a file for printing. They must have access to the printer when they run. To be capable of detecting printer faults, a print filter must be a fast filter. Any filter that uses the `PRINTER` keyword as a filter option must be installed as a fast filter.

Slow filters require a great deal of processing time to prepare a file for printing. They do not require access to the printer when they run. Slow filters are run in the background so they do not tie up the printer, allowing other files that do not need slow filtering to be printed.

Converting Files

The LP print service uses print filters to convert files from one content type to another. You can specify the accepted file content types for each printer. The user

specifies the file content type when submitting a print request, and the LP print service finds a printer that can print files of that content type. Because many applications can generate files for various printers, this is often sufficient. However, some applications can generate files that cannot be printed on any available printers.

Each time the LP print service receives a request to print a type of file that is in a format that cannot be accepted directly by a printer, the LP print service tries to match the content type of the print request with the content type of the available (or specified) printer. If there is a match, the file can be sent directly to the printer without filtering. If no match is found, or if the content type specifies that a filter be used, the LP print service tries to match the content type of the file with the input content type of available filters, and match the output type of the filter with the content type of the printer. When an appropriate filter is found, the print request is passed through the filter.

Handling Special Printing Modes

A print filter handles special modes and requests to print specific pages. A special printing mode is needed to print any characteristics of print requests that require a customized filter. Filters handle the following characteristics:

- Printer type
- Character pitch
- Line pitch
- Page length
- Page width
- Pages to print
- Character set
- Form name
- Number of copies

The LP print service provides default settings for these characteristics; however, a print filter can handle some characteristics more efficiently. For example, some printers can handle multiple copies more efficiently than the LP print service, and, in this case, you can provide a filter for multiple-copy page control.

Detecting Printer Faults

Each printer has its own way of detecting printer faults and transmitting fault signals to the LP print service. The LP print service only checks for hang-ups (loss of carrier) and excessive delays in printing.

Some printers provide good fault coverage and can send a message describing the reason for a fault. Other printers indicate a fault by using signals other than the

signals indicating loss of carrier signal or shut off of data flow. A filter is required to interpret this additional printer fault information.

A filter can also put a print request on hold, wait for a printer fault to clear, and then resume printing. With this capability, the print request that was interrupted does not need to be reprinted in its entirety. Only a filter that knows the control sequences used by a printer can determine where to break a file into pages. Consequently, only such a filter can find the place in the file where printing should start after a fault is cleared.

When a print filter generates messages, those messages are handled by the LP print service, and alerts are sent to the system administrator if alerts are enabled. For further information, see “Setting Up Printer Fault Alerts” on page 100.

Requirements for a Print Filter Program

A print filter can be simple or complex, but it has to meet the following requirements:

- The filter should get the contents of a file from its standard input and send the converted file to the standard output.
- A program cannot be used as a filter if it references external files. You might be tempted to use a program like `troff`, `nroff`, or a similar word processing program as a filter. The LP print service does not recognize references to other files, known as include files, from a filter program. Because `troff` and `nroff` allow include files, they can fail when used as filters. If the program needs other files to complete its processing, it should not be used as a filter.
- The filter should not depend on files that normally would not be accessible to a user. If a filter fails when run directly by a user, it will fail when run by the LP print service.
- A slow filter can send messages about errors in the file to standard error; a fast filter should not. Error messages from a slow filter are collected and sent to the user who submitted the print request.
- If a slow filter dies because it received a signal, the print request is stopped and the user who submitted the request is notified. Likewise, if a slow filter exits with a non-zero exit code, the print request is stopped and the user is notified. The exit codes from fast filters are treated differently.

If you want the filter to detect printer faults, it should also meet the following requirements:

- If possible, the filter should wait for a fault to be cleared before exiting. It should also continue to print at the top of the page where printing stopped after the fault is cleared. If you do not want use the continuation feature, the LP print service will stop the filter before alerting the administrator.
- The filter should send printer fault messages to its standard error as soon as the fault is recognized. It does not have to exit; it can wait for the fault to be cleared.

- The filter should not send messages about errors in the file to standard error. These messages should be included in the standard output, where they can be read by the user.
- The filter should exit with a zero exit code if the file is finished printing (even if errors in the file have prevented it from being printed correctly).
- The filter should exit with a non-zero exit code only if a printer fault has prevented it from finishing a print request.
- When added to the filter table, the filter must be added as a fast filter.

Creating a Print Filter Definition

A print filter definition tells the LP print service about the filter, what print filter program to run, what kind of conversion it does, and so on. A set of filter descriptor files are provided in the `/etc/lp/fd` directory. These files describe the characteristics of the filters (for example, fast or slow filter), and point to the filter programs (for example, `/usr/lib/lp/postscript/postdaisy`).

When defining a new print filter, in addition to writing a filter program, you must create a print filter definition. A print filter definition contains the following information used by the LP print service:

- Name of the filter program to run
- Input types it accepts
- Output types it produces
- Printer types to which it can send jobs
- Names of specific printers to which it can send jobs
- Filter types (either fast or slow)
- Options

You can type the characteristics as direct input to the `lpfilter` command. You also can create a file that specifies the filter's characteristics, and use the file name as input to the `lpfilter` command. Such a file is called a *filter descriptor file* and should be located in the `/etc/lp/fd` directory. These files are not the filters themselves, but rather point to the filters.

Whether you store the information in a file, or enter it directly on the command line, use the following format:

```
Command: command-pathname [options]
Input types: input-type-list
Output types: output-type-list
Printer types: printer-type-list
```

(continued)

```
Printers: printer-list
Filter type: fast or slow
Options: template-list
```

Note - If you provide more than one definition (that is, more than one line) for any filter characteristic other than `Options`, only the second definition will be used by the print service.

The information can be arranged in any order, and not all the information is required. When you do not specify values, those shown in the table below are assigned by default. They are not very useful, which is why you should specify explicit values.

TABLE 7–5 Default Values for `lpfilter` Arguments

Item	Default
Input types	`any`
Output type	`any`
Printer types	`any`
Printers	`any`
Filter type	`slow`

Command

Use the full path of the filter program. If there are any fixed options that the program always needs, include them here.

Input Types

Input types is a list of file content types that the print filter can process. The LP print service does limit the number of input types, but most filters can accept only one

type. Several file types can be similar enough that the filter can deal with them. You can use whatever names you like, with a maximum of 14 alphanumeric characters and dashes. Do not use underscores as part of the input type name.

The LP print service uses these names to match a filter to a file type, so follow a consistent naming convention. For example, if more than one filter can accept the same input type, use the same name for that input type when you specify it for each filter. Inform your users of these names so they know how to identify the file type when submitting a file for printing.

Output Types

Output types is list of file types that the filter can produce as output. For each input type, the filter produces a single output type. The output type can vary, however, from job to job. The name of the output type is restricted to 14 alphanumeric characters and dashes.

The output type names should either match the types of available (local or remote) printers, or match the input types handled by other filters. The LP print service groups filters in a shell pipeline if it finds that several passes by different filters are needed to convert a file. It is unlikely that you will need this level of sophistication, but the LP print service allows it. Try to find a set of filters that takes as input types all the different files the users might want printed, and that converts those files directly into file types the printer can handle.

Printer Types

Printer types is a list of the types of printers into which the print filter can convert files. For most printers and filters, you can leave this part of the filter definition blank, because it is identical to the list of output types. But it can be different. For example, you could have a printer with a single printer type for purposes of initialization, but which can recognize several different file content types. Essentially, this printer has an internal filter that converts the various file types into one that it can handle. Thus, a filter might produce one of several output types that match the file types that the printer can handle. The print filter should be marked as working with that printer type.

As another example, you might have two different models of printers that are listed as accepting the same file types. Due to slight differences in manufacture, however, one printer deviates in the results it produces. You label the printers as being of different printer types, say A and B, where B is the one that deviates. You create a filter that adjusts files to account for the deviation produced by printers of type B. Because this filter is needed only for those printer types, you would list it as working only on type B printers.

Printers

A print filter is normally able to work with all printers that accept its output, so you can usually skip this part of the filter definition.

You might, however, have some printers that are or inappropriate for the output that the filter produces. For example, you might want to dedicate one printer for fast turnaround, only sending files that require no filtering to that printer. Other printers of identical type can be used for files that need extensive filtering before they can be printed.

Filter Type

The LP print service recognizes fast and slow filters, as described in “Types of Filters” on page 163.

Slow filters that are invoked by printing modes (using the `lp -y` command) must be run on the system from which the print request originated. The LP print service cannot pass values for modes to print servers. It can, however, match a file content type (specified after the `-T` option of the `lp` command) to a content type on a print server. Therefore, if you want to activate special modes on a print server, you must specify content types that permit the LP print service to match input types and output types.

Options

Options specify how different types of information are converted into command-line arguments to the filter command. This information can include specifications from a user (with the print request), the printer definition, and the specifications implemented by any filters used to process the request.

Defining Print Filter Options With Templates

There are 13 sources of information for defining print filter options, each of which is represented by a *keyword*. Each option is defined in a *template*. A template is a statement in a filter definition that defines an option to be passed to the filter command, based on the value of one of the characteristics of the filter.

The options specified in a filter definition can include none, all, or any subset of the 13 keywords. In addition, a single keyword can be defined more than once, if multiple definitions are required for a complete filter definition. The table below contains descriptions of the 13 keywords available for defining `Options` in a print filter definition.

TABLE 7–6 Print Filter Options Keywords

Characteristic	Keyword	Possible Patterns	Example
Content type (input)	`INPUT`	*content-type*	`troff`
Content type (output)	`OUTPUT`	*content-type*	`postscript, impress`
Printer type	`TERM`	*printer-type*	`att495`
Printer name	`PRINTER`	*printer-name*	`lp1`
Character pitch	`CPI`	*scaled-decimal*	`10`
Line pitch	`LPI`	*scaled-decimal*	`6`
Page length	`LENGTH`	*scaled-decimal*	`66`
Page width	`WIDTH`	*scaled-decimal*	`80`
Pages to print	`PAGES`	*page-list*	`1-5,13-20`
Character set	`CHARSET`	*character-set*	`finnish`
Form name	`FORM`	*form-name*	`invoice2`
Number of copies	`COPIES`	*integer*	`3`
Special modes	`MODES`	*mode*	`landscape`

A print filter definition can include more than one template. Multiple templates are entered on a single line and separated with commas, or they are entered on separate lines, preceded by the Options: prefix.

The format of a template is as follows:

keywordpattern = replacement

The keyword identifies the type of option being registered for a particular characteristic of the filter.

The *pattern* is a specific option for the keyword.

The *replacement* is what happens when the keyword has the noted value.

For an example of how an option is defined for a particular filter, suppose you want to have the print service scheduler assign print requests to filters following this criteria:

- If the type of `OUTPUT` to be produced by the filter is `impress`, then pass the `-I` option to the filter.
- If the type of `OUTPUT` to be produced by the filter is postscript, then pass the `-P` option to the filter.

To specify these criteria, provide the following templates as options to the `lpfilter` command:

```
Options: OUTPUT impress=-I, OUTPUT postscript=-P
```

If the `Options` line becomes too long, put each template on a separate line, as follows:

```
Options: OUTPUT impress=-I
Options: OUTPUT postscript=-P
```

In both templates, the *keyword* is defined as `OUTPUT`. In the first template, the pattern is `impress` and the value of the *replacement* is `–-I`. In the second template, the value of *pattern* is `postscript` and the value of *replacement* is `–P`.

To find out which values to supply for each type of template (that is, for the *pattern* and *replacement* arguments for each keyword), consider the following:

- The values for the `INPUT` templates come from the file content type that needs to be converted by the filter.
- The values for the `OUTPUT` templates come from the output type that has to be produced by the filter.
- The value for the `TERM` template is the printer type.
- The value for the `PRINTER` template is the name of the printer that will print the final output.
- The values for the `CPI`, `LPI`, `LENGTH`, and `WIDTH` templates come from the user's print request, the form being used, or the default values for the printer.
- The value for the `PAGES` template is a list of pages that should be printed. Typically, it is a list of page ranges separated by commas. Each page range consists of a pair of numbers separated by a dash, or a single number. (For example, 1–5,6,8,10 indicates pages 1 through 5, plus pages 6, 8, and 10.) However, whatever value was given in the `–P` option to a print request is passed unchanged.
- The value for the `CHARSET` template is the name of the character set to be used.
- The value for the `FORM` template is the name of the form requested by the `lp -f` command (the command used to submit a print request).

- The value of the `COPIES` template is the number of copies of the file to print. If the filter uses this template, the LP print service will reduce to one the number of copies of the filtered file it prints, since this "single copy" includes the multiple copies produced by the filter.
- The value of the `MODES` template comes from the `lp -y` command. Because a user can specify several `–y` options, there might be several values for the `MODES` template. The values will be applied in the left-to-right order given by the user.

The *replacement* part of a template shows how the value of a template should be given to the filter program. It is typically a literal option, sometimes with the placeholder asterisk (*) included to show where the value goes. The *pattern* and *replacement* also can use the regular expression syntax of `ed(1)` for more complex conversion of user input options into filter options. All regular expression syntax of `ed`(1) is supported, including the `\( ... \)` and `\n` constructions, which can be used to extract portions of the *pattern* for copying into the *replacement*, and the `&`, which can be used to copy the entire *pattern* into the *replacement*.

Note - If a comma or an equal sign (=) is included in a *pattern* or a *replacement*, precede it with a backslash (\). A backslash in front of any of these characters is removed when the *pattern* or *replacement* is used.

▼ How to Create a New Print Filter

1. **Log in as superuser or lp on the print server.**
2. **Create a print filter program.**

   See "Writing a Print Filter Program" on page 163 for information on print filter programs. By convention, filter programs for PostScript printers are located in the `/usr/lib/lp/postscript` directory. You should put programs you create under `/usr/lib/lp` in a directory of your choosing.
3. **Create a print filter definition.**

   See "Creating a Print Filter Definition" on page 166 for information on print filter definitions. You should save the printer filter definition in a text file. By convention, filter definitions are located in the `/etc/lp/fd` directory and are identified with the `.fd` suffix.
4. **Add the print filter to a print server.**

   For instructions, see "How to Add a Print Filter" on page 132.

Examples—Creating a New Print Filter

The following example shows a print filter definition to convert `N37` or `Nlp` to `simple`.

```
Input types: N37, Nlp, simple
Output types: simple
Command: /usr/bin/col
Options: MODES expand = -x
Options: INPUT simple = -p -f
```

In this example, the print filter program is named `col`. Once you add the new print filter to a print server, a user's print requests will be handled as follows:

- When a user enters the following command:

  ```
  $ lp -y expand report.doc
  ```

  The print filter program is run with the following arguments to convert the file:

  ```
  /usr/bin/col -x -p -f
  ```

- When a user enters the following command:

  ```
  $ lp -T N37 -y expand report.doc
  ```

  The print filter program is run with the following arguments to convert the file:

  ```
  /usr/bin/col -x
  ```

The following example shows a print filter definition to convert from `troff` to PostScript.

```
Input types: troff

Output types: postscript

Printer types: PS

Filter type: slow

Command: /usr/lib/lp/postscript/dpost

Options: LENGTH * = -l*

Options: MODES port = -pp, MODES land = -pl
Options: MODES group \=\([1-9]\) = -n\l
```

In this example, the filter program is named `dpost`. It takes one input type, `troff`, produces a `postscript` output, and works with any printer of type `PS` (PostScript). Users need to give just the abbreviation `port` or `land` when they ask for the paper orientation to be in portrait mode or landscape mode. Because these options are not intrinsic to the LP print service, users must specify them using the `lp -y` command.

After you add the new print filter to a print server, print requests will be handled as follows:

- When a user enters the following command to submit a `troff` file type for printing on a PostScript printer (type `PS`), with requests for landscape orientation and a page length of 60 lines:

```
$ lp -T troff -o length=60 -y land -d luna ch1.doc
```

  The print filter program `dpost` is run with the following arguments to convert the file:

```
/usr/lib/lp/postscript/dpost -l60 -pl luna ch1.doc
```

- When a user enters the following command:

```
$ lp -T troff -y group=4 -d luna ch1.doc
```

  The print filter program `dpost` is run with the following arguments to convert the file:

```
/usr/lib/lp/postscript/dpost -n4
```

Creating a New Printer Form

When you want to provide a new form, you must define its characteristics by entering information about nine required characteristics (such as page length and page width) as input to the `lpforms` command. The LP print service uses this information to:

- Initialize the printer so that printing is done properly on the form
- Send reminders to the system administrator about how to handle the form

The form name can be anything you choose, as long as it does not contain more than 14 alphanumeric characters and underscores. The information must be in the following format:

```
Page length: scaled number
Page width: scaled number
Number of pages: integer
Line pitch: scaled number
Character pitch: scaled number
Character set choice: character-set-name [,mandatory]
Ribbon color: ribbon-color
```

(continued)

```
Comment:
informal notes about the form
Alignment pattern: [content-type] alignment pattern
```

The optional phrase `[,mandatory]` means that the user cannot override the character set choice in the form. The *content-type* can be given, although this is optional, with an alignment pattern. If this attribute is given, the print service uses it to determine, as necessary, how to filter and print the file.

With two exceptions, the information can appear in any order. The exceptions are the `Alignment pattern` (which must always be last), and the *comment* (which must always follow the line with the `Comment:` prompt). If the comment contains a line beginning with a key phrase (like `Page length`, `Page width` , and so on), precede that line with a > character so the key phrase is not at the beginning of the line. The initial > character is stripped from the comment and is not displayed.

Not all of the information must be given. When you do not specify values for the items listed in the table below the default values are assigned. Before running the `lpforms` command, gather the following information about the new form:

TABLE 7–7 Default Form Values

Item	Default	Description
Page length	66 lines	The length of the form, or the length of each page in a multipage form. This information can be the number of lines, or the size in inches or centimeters.
Page width	80 columns	The width of the form, in characters, inches, or centimeters.
Number of pages	1	The number of pages in a multipage form. The LP print service uses this number with a print filter (if available) to restrict the alignment pattern to a length of one form. See the description of alignment pattern below. If no filter is available, the LP print service does not truncate the output.

TABLE 7–7 Default Form Values *(continued)*

Item	Default	Description
Line pitch	6 lines per inch	A measurement of how close lines appear on the form. This is also called leading. It is the distance between two lines, from baseline to baseline, measured by either lines per inch or lines per centimeter.
Character pitch	10 characters per inch	A measurement of how close together characters appear on the form. It is the distance between characters, measured by either characters per inch or characters per centimeter.
Character set choice	Any	The character set, print wheel, or font cartridge that should be used when this form is used. Users can choose a different character set for their own print requests when using this form, or you can require that only one character set be used.
Ribbon color	Any	If the form should always be printed using a certain color ribbon, the LP print service can give a mount alert message indicating which color to use.
Comment	(No default)	Any remarks that might help users understand the form. For example, the remarks could indicate the name of the form, its revision, its purpose, or restrictions on its use.
Alignment pattern	(No default)	A sample file that the LP print service uses to fill one blank form. When mounting the form, you can print this pattern on the form to align it properly. You can also define a content type for this pattern so that the print service knows how to print it.

Note - The LP print service does not try to mask sensitive information in the alignment pattern. If you do not want sensitive information printed on sample forms—for example when you align checks—then you should mask the appropriate data. The LP print service keeps the alignment pattern stored in a safe place, where only those logged in as root or lp can read it.

When you have gathered the information for the form, you enter it as input to the `lpforms` command. You should record this information first in a separate file so you can edit it before entering it with `lpforms`. You can then use the file as input instead of typing each piece of information separately after a prompt.

▼ How to Create a New Form Definition

1. **Log in as superuser or lp on the print server.**

2. **Create a form definition file.**

   See "Creating a New Printer Form" on page 174 for a description on creating print forms. You should save the printer definition in a text file.

3. **Add the form to the LP print service by using the `lpadmin` command.**

```
# lpadmin -p printer-name -M -f form-name
```

4. **Add the form to a print server.**

   For instructions, see "How to Add a Form" on page 137.

CHAPTER 8

LP Print Service Reference Information

This chapter provides background information on the LP print service.

- "The Structure of the LP Print Service" on page 180
- "LP Print Service Commands" on page 189
- "Functions of the LP Print Service" on page 190
- "How LP Administers Files and Schedules Local Print Requests" on page 191
- "Scheduling Network Print Requests" on page 192
- "Filtering Print Files" on page 193
- "What the Printer Interface Program Does" on page 193
- "How the `lpsched` Daemon Tracks the Status of Print Requests" on page 194
- "Cleaning Out Log Files" on page 194

For step-by-step instructions on print management tasks, see:

- Chapter 4
- Chapter 5
- Chapter 6
- Chapter 7

The LP Print Service

The *LP print service* is a set of software utilities that allows users to print files while they continue to work. Originally, the print service was called the LP spooler. (LP stood for line printer, but its meaning now includes many other types of printers, such as laser printers. Spool is an acronym for system peripheral operation off-line.)

The print service consists of the LP print service software, any print filters you might provide, and the hardware (the printer, system, and network connections).

The Structure of the LP Print Service

This section describes the directory structure, files, logs, and commands of the LP print service.

LP Print Service Directories

The files of the LP print service are distributed among seven directories, as shown in the table below.

TABLE 8–1 Directories for the LP Print Service

Directory	Contents
`/usr/bin`	The LP print service user commands
`/etc/lp`	A hierarchy of LP server configuration files
`/usr/share/lib`	The `terminfo` database directory
`/usr/sbin`	The LP print service administrative commands
`/usr/lib/lp`	The LP daemons; directories for binary files and PostScript filters; and the `model` directory (which contains the standard printer interface program)
`/var/lp/logs`	The logs for LP activities: `lpsched.`*n* – Messages from `lpsched` and `requests.`*n* – Information about completed print requests
`/var/spool/lp`	The spooling directory where files are queued for printing
`/var/spool/print`	The LP print service client-side request staging area

LP Print Service Configuration Files

The scheduler stores configuration information in LP configuration files located in the `/etc/lp` directory, as described in the table below.

Caution - The configuration files listed in the table below are private interfaces, and are subject to change in future releases. You should not build software that relies on these files being in their current locations or that relies on the data being in the format currently used.

TABLE 8–2 Contents of the `/etc/lp` Directory

File	Type	Description
`classes`	Directory	Files identifying classes provided by the `lpadmin -c` command.
`fd`	Directory	Description of existing filters.
`filter.table`	File	Print filter lookup table.
`forms`	Directory	Location to put files for each form. Initially, this directory is empty.
`interfaces`	Directory	Printer interface program files.
`logs`	Link to `/var/lp/logs`	Log files of printing activities.
`model`	Link to `/usr/lib/lp/model`	The standard printer interface program.
`printers`	Directory	Directories for each local printer. Each directory contains configuration information and alert files for an individual printer.
`pwheels`	Directory	Print wheel or cartridge files.

These configuration files serve a similar function to the `/etc/printcap` file in the SunOS 4.1 release.

Note - You can check the contents of the configuration files, but you should not edit them directly. Instead, use the `lpadmin`(1M) command to make configuration changes. Your changes will be written to the configuration files in the `/etc/lp` directory. The `lpsched` daemon administers and updates the configuration files.

The `/etc/lp/printers` directory has a subdirectory for each local printer known to the system. The following example shows the `/etc/lp/printers` subdirectories of printers `sparc1` and `luna`.

```
$ ls -l /etc/lp/printers
drwxrwxr-x 2 lp lp 512 Jan 23 23:53 luna
drwxrwxr-x 2 lp lp 512 Jan 11 17:50 sparc1
```

The following table describes the files within each of the printer-specific directories.

File Name	Description
`alert.sh`	Shell to execute in response to alerts
`alert.vars`	Alert variables
`configuration`	Configuration file
`users.deny`	List of users to whom printer access is denied
`comment`	Printer description

The configuration file for the printer `luna`, `/etc/lp/printers/luna/configuration`, would typically appear as follows:

```
Banner: on: Always
Content types: PS
Device: /dev/term/b
Interface: /usr/lib/lp/model/standard
Printer type: PS
Modules: default
```

The `terminfo` Database

The `/usr/share/lib` directory contains the `terminfo` database directory, which contains definitions for many types of terminals and printers. The LP print service uses information in the `terminfo` database to initialize a printer, to establish a

selected page size, character pitch, line pitch, and character set, as well as to communicate the sequence of codes to a printer.

Each printer is identified in the `terminfo` database with a short name. See “Printer Type” on page 57 for a description of the structure of the `terminfo` database. If necessary, you can add entries to the `terminfo` database, but it is a tedious and time-consuming process. See “Adding a `terminfo` Entry for an Unsupported Printer” on page 154.

Daemons and LP Internal Files

The `/usr/lib/lp` directory contains daemons and files used by the LP print service, as described in the table below.

TABLE 8–3 Contents of the `/usr/lib/lp` Directory

File	Type	Description
`bin`	Directory	Contains files for generating printing alerts, slow filters, and queue management programs.
`lpsched`	Daemon	Manages scheduling of LP print requests.
`model`	Directory	Contains the standard printer interface program.
`postscript`	Directory	Contains all PostScript filter programs provided by the LP print service. These filters come with descriptor files in the `/etc/lp/fd` directory that tell the LP print service the characteristics of the filters and where to locate them.

LP Print Service Log Files

The LP print service maintains two sets of log files described in the following table.

Log File Name	Description
`syslogd`(1M)	Set `lpr.debug` in `/etc/syslog.conf` to enable LP print service logging
`/var/spool/lp`	A list of current requests that are in the print queue
`/var/lp/logs/requests`	An ongoing history of print requests

Print Queue Logs

The scheduler for each system keeps a log of print requests in the directories `/var/spool/lp/tmp/`*system* and `/var/spool/lp/requests/`*system*. Each print request has two files (one in each directory) that contain information about the request. The information in the `/var/spool/lp/requests/`*system* directory can be accessed only by root or lp. The information in the `/var/spool/lp/tmp/`*system* can be accessed only by the user who submitted the request, root, or `lp`.

The following example shows the contents of the `/var/spool/lp/tmp/terra` directory:

```
$ ls /var/spool/lp/tmp/terra
20-0 21-0
terra$ cat 21-0
C 1
D slw2
F /etc/default/login
P 20
t simple
U tamiro
s 0x1000
```

These files remain in their directories only as long as the print request is in the queue. Once the request is finished, the information in the files is combined and appended to the file `/var/lp/logs/requests`, which is described in the next section.

Use the information in the `/var/spool/lp logs` if you need to track the status of a print request that is currently in the queue.

History Logs

The LP print service records a history of printing services in two log files: `lpsched` and `requests`. These log files are located in the `/var/lp/logs` directory. You can use the information in these logs to diagnose and troubleshoot printing problems. This is an example of the contents of the `/var/lp/logs` directory:

```
# cd /var/lp/logs
# ls
lpsched.1    requests    requests.2
lpsched      lpsched.2   requests.1
#
```

The files with the `.1` and `.2` suffixes are copies of the previous day's logs. Each day, the `lp cron` job cleans out the `lpsched` and `requests` log files and keeps copies for two days. See "Creating and Editing `crontab` Files" on page 501 for suggestions on modifying the `cron` job for cleaning out the `requests` log.

The two most important log files for troubleshooting is the `lpsched` log, which contains information about local printing requests

The `requests` log contains information about print requests that are completed and no longer in the print queue. Once a request is finished printing, the information in the `/var/spool/lp` log files is combined and appended to the `/var/lp/logs/requests` log.

The `requests` log has a simple structure, so that you can extract data using common UNIX shell commands. Requests are listed in the order they are printed, and are separated by lines showing their request IDs. Each line below the separator line is marked with a single letter that identifies the kind of information contained in that line. Each letter is separated from the data by a single space.

The following example shows the contents of a `requests` log:

```
# pwd
/var/lp/logs
# tail requests.2
= slw2-20, uid 200, gid 200, size 5123, Tue Jun 17 10:16:10 MDT
1998
z slw2
C 1
D slw2
F /etc/motd
P 20
t simple
U irving
s 0x0100
#
```

The table below shows the letter codes and the content of their corresponding lines in the LP `requests` log.

TABLE 8–4 Letter Codes in the LP `requests` Log

Letter	Content of Line
`=`	The separator line. It contains the following items: request ID, user ID (UID), and group IDs (GIDs) of the user, the total number of bytes in the original (unfiltered) file size, and the time when the request was queued.
`C`	The number of copies printed.
`D`	The printer or class destination or the word `any`.
`F`	The name of the file printed. The line is repeated for each file printed; files were printed in the order shown.
`f`	The name of the form used.
`H`	One of three types of special handling: resume, hold, and immediate.
`N`	The type of alert used when the print request was successfully completed. The type is the letter `M` if the user was notified by email or `W` if the user was notified by a message to the terminal.
`O`	The printer-dependent –`o` options (for example, nobanner).
`P`	The priority of the print request.
`p`	The list of pages printed.
`r`	A single-letter line that is included if the user asked for “raw” processing of the files (the `lp -r` command).
`S`	The character set, print wheel, or cartridge used.
`s`	The outcome of the request, shown as a combination of individual bits expressed in hexadecimal form. Several bits are used internally by the print service. The bits and what they mean are describe in the table below.
`T`	The title placed on the banner page.
`t`	The type of content found in the files.
`U`	The name of the user who submitted the print request.
`x`	The slow filter used for the print request.

TABLE 8–4 Letter Codes in the LP `requests` Log *(continued)*

Letter	Content of Line
`Y`	The list of special modes for the print filters used to print the request.
`z`	The printer used for the request. This printer differs from the destination (the `D` line) if the request was queued for any printer or a class of printers, or if the request was moved to another destination.

The table below shows the outcome codes in the LP `requests` log and their descriptions.

TABLE 8–5 Outcome Codes in the LP requests Log

Outcome Code	Description
`0x0001`	The request was held pending resume.
`0x0002`	Slow filtering is running.
`0x0004`	Slow filtering finished successfully.
`0x0008`	The request is on the printer.
`0x0010`	Printing finished successfully.
`0x0020`	The request was held pending user change.
`0x0040`	The request was canceled.
`0x0080`	The request will print next.
`0x0100`	The request failed filtering or printing.
`0x0200`	The request is in transit to a remote printer. (obsolete)
`0x0400`	The user will be notified.
`0x0800`	A notification is running.

TABLE 8–5 Outcome Codes in the LP requests Log *(continued)*

Outcome Code	Description
0x1000	A remote system has accepted the request. (obsolete)
0x2000	The administrator placed a hold on the request.
0x4000	The printer had to change filters.
0x8000	The request is temporarily stopped.

Spooling Directories

Files queued for printing are stored in the `/var/spool/lp` directory until they are printed, which might be only seconds. The table below shows the contents of the `/var/spool/lp` directory.

TABLE 8–6 Contents of the `/var/spool/lp` Directory

File	Type	Description
SCHEDLOCK	File	Lock file for the scheduler. Check for this file if the scheduler dies and will not restart.
admins	Directory	Link to `/etc/lp`.
bin	Directory	Link to `/usr/lib/lp/bin`.
logs	Link	Link to `../lp/logs` where completed print requests are logged.
model	Link	Link to `/usr/lib/lp/model`.
requests	Directory	Directory that contains subdirectories for each configured printer where print requests are logged until printed. Users cannot access this log.
system	Directory	A print status file for the system.

TABLE 8–6 Contents of the `/var/spool/lp` Directory *(continued)*

File	Type	Description
`temp`	Link	Link to `/var/spool/lp/tmp/`*hostname*, which contains the spooled requests.
`tmp`	Directory	Directory for each configured printer where print requests are logged until printed. Changes to existing print requests are also recorded in this log.

LP Print Service Commands

The table below lists frequently used LP print service commands. You must be root or `lp` to use the `1M` commands.

TABLE 8–7 Quick Reference to LP Print Service Commands

Command	Task
`enable`(1)	Activate a printer
`cancel`(1)	Cancel a print request
`lp`(1)	Send one or more file(s) to a printer
`lpstat`(1)	Report the status of the LP print service
`disable`(1)	Deactivate one or more printers
`accept`(1M)	Permit print requests to be queued for a specific destination
`reject`(1M)	Prevent print requests from being queued for a specific destination

TABLE 8–7 Quick Reference to LP Print Service Commands *(continued)*

Command	Task
`lpadmin`(1M)	Set up or change printer configuration
`lpfilter`(1M)	Set up or change filter definitions
`lpforms`(1M)	Set up or change preprinted forms
`lpadmin`(1M)	Mount a form
`lpmove`(1M)	Move output requests from one destination to another
`lpsched`(1M)	Start the LP print service scheduler
`lpshut`(1M)	Stop the LP print service scheduler
`lpusers`(1M)	Set or change the default priority and priority limits that can be requested by users of the LP print service

Functions of the LP Print Service

The LP print service performs the following functions:

- Administers files and schedules local print requests
- Receives and schedules network requests
- Filters files (if necessary) so they print properly
- Starts programs that interface with the printers
- Tracks the status of jobs
- Tracks forms mounted on the printer

- Tracks print wheels currently mounted
- Delivers alerts to mount new forms or different print wheels
- Delivers alerts about printing problems

The table below describes the directory structure and commands.

How LP Administers Files and Schedules Local Print Requests

The LP print service has a scheduler daemon called `lpsched`. The scheduler daemon updates the LP system files with information about printer setup and configuration.

The lpsched daemon schedules all local print requests on a print server, as shown in the figure below, whether users issue the requests from an application or from the command line. Also, the scheduler tracks the status of printers and filters on the print server. When a printer finishes a request, the scheduler schedules the next request, if there is one, in the queue on the print server.

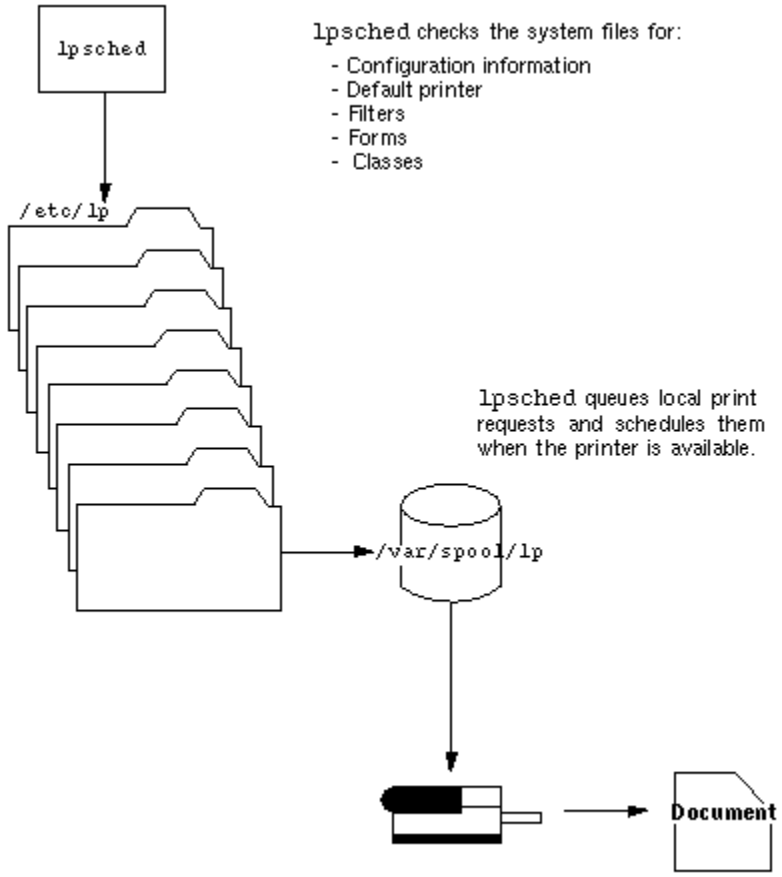


Figure 8–1 The `lpsched` Daemon Schedules Local Print Requests

Each print server must have *only* one LP scheduler running. The scheduler is started when a system is booted (or enters run level 2) by the control script `/etc/rc2.d/S80lp`. Without rebooting the systems, you can stop the scheduler with the `/usr/lib/lp/lpshut` command and restart the scheduler with the `lpsched` command. The scheduler for each system manages requests issued to the system by the `lp` commands.

Scheduling Network Print Requests

Each print client communicates directly with a print sever over the network. The communication is done between the requesting command (`lp`, `lpstat`, `cancel`, `lpr`, `lpq`, or `lprm`) and the print service on the print server. Doing so, reduces the

print system overhead on client only systems, improving scalability, performance and accuracy of data.

Print servers now listen for print request with the Internet services daemon (inetd). Upon hearing a request for print service from the network, `inetd` starts a program called the “protocol adaptor” (`in.lpd`). The protocol adaptor translates the print request and communicates it to the print spooler, returning the results to the requester. It starts on demand and exits when it has serviced the network request. This eliminates idle system overhead for printing. It also eliminates any additional system configuration for networked printing support as was the case in previous versions of Solaris printing.

Filtering Print Files

Print filters are programs on the print server that convert the content of a queued file from one format to another.

A print filter can be as simple or as complex as needed. The SunOS release provides print filters in the `/usr/lib/lp/postscript` directory that cover most situations where the destination printer requires the data to be in PostScript format. If you need filters for non-PostScript printers, you have to create the filters and add them to the systems that need them.

A set of *print filter descriptor files* are provided in the `/etc/lp/fd` directory. These descriptor files describe the characteristics of the filter (for example, fast or slow filter), and point to the filter program (for example, `/usr/lib/lp/postscript/postdaisy`).

What the Printer Interface Program Does

The LP print service interacts with other parts of the operating system. It uses a standard printer interface program to:

- Initialize the printer port, if necessary. The standard printer interface program uses the `stty` command to initialize the printer port.
- Initialize the printer. The standard printer interface program uses the `terminfo` database and the `TERM` shell variable to find the appropriate control sequences.
- Print a banner page, if necessary.
- Print the correct number of copies specified by the print request.

The LP print service uses the standard interface program (found in the `/usr/lib/lp/model` directory) unless you specify a different one. You can create custom interface programs, but you must make sure that the custom program does not terminate the connection to the printer or interfere with proper printer initialization.

How the lpsched Daemon Tracks the Status of Print Requests

The lpsched daemon on both the print server and print client keeps a log of each print request that it processes and notes any errors that occur during the printing process. This log is kept in the /var/lp/logs/lpsched file. Every night, the lp cron job renames /var/lp/logs/lpsched to a new lpsched.*n* file and starts a new log file. If errors occur or jobs disappear from the print queue, you can use the log files to determine what lpsched has done with a printing job.

Cleaning Out Log Files

The lpsched and requests log files in the /var/lp/logs directory grow as information is appended. The LP print service uses a default cron job to clean out the log files. The lp cron job is located in the /var/spool/cron/crontabs/lp file. It periodically moves the contents of the log files. The contents of *log* are moved to *log*.1, and the contents of *log*.1 are moved to *log*.2. The contents of *log*.2 are lost (that is, replaced by the former contents of *log*.1) when *log*.2 gets overwritten.

```
# pwd
/var/lp/logs
# tail requests
s 0x1010
= slw2-20, uid 200, gid 200, size 5123, Mon Jun 16 12:27:33 MDT
1997
z slw2
C 1
D slw2
F /etc/motd
P 20
t simple
U irving
s 0x1010
#
```

▼ How to Change Frequency of Printer Request Log Rotation

Starting with the Solaris 2.6 release, the requests log file on the printer server is rotated weekly rather than daily. You can change the rotation interval back to daily if the printer server is busy.

1. **Become superuser or** lp **on the printer server.**

2. **Set the** EDITOR **environment variable.**

```
# EDITOR=vi
# export EDITOR
```

3. **Edit the** `lp crontab` **file.**

```
# crontab -e lp
```

4. **Change the first line of the file which rotates the** `requests` **log files every Sunday (**`0`**) to an asterisk (**`*`**) for daily rotation:**

```
13 3 * * * cd /var/lp/logs; if [ -f requests ]; then if
[ -f requests.1 ]; then /bin/mv requests.1 requests.2; fi; /usr/bin/cp
requests requests.1; >requests; fi
```

5. **Save the file and exit.**

How Local Printing Works

The figure below shows what happens when a user submits a request to print a PostScript file on a *local* printer, which is a printer connected to the user's system. The local system does all processing; however, the print request follows the same path it would if the client and server were separate systems. Requests always flow from client to server following the same path.

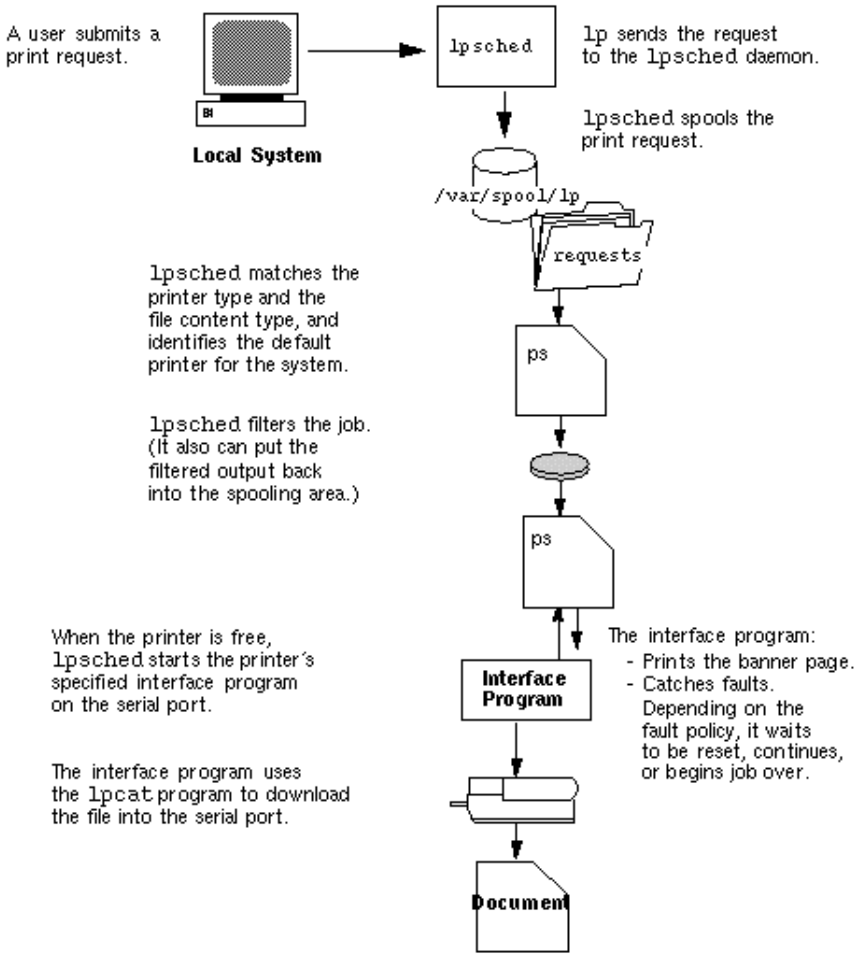


Figure 8–2 The Local Printing Process

How Remote Printing Works

The figure below shows what happens when a user on a SunOS 5.8 print client submits a print request to a SunOS 4.1 print server. The command opens a connection and handles it's own communications with the print server directly.

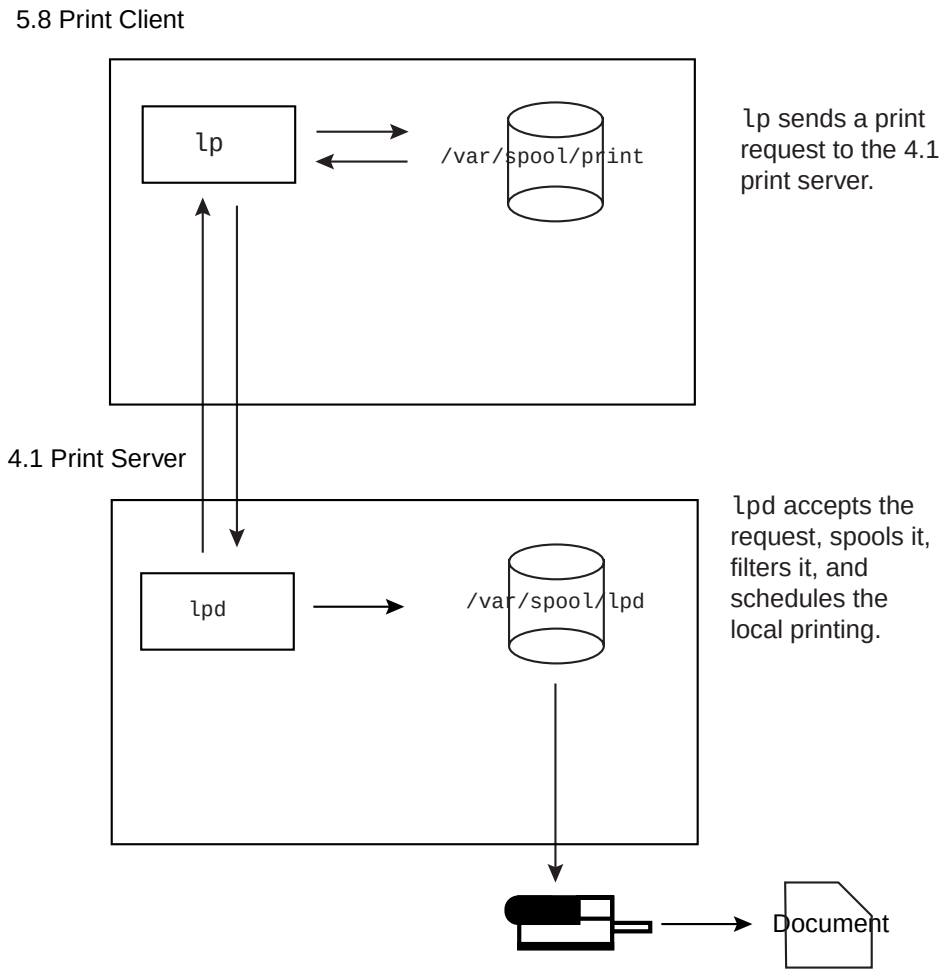


Figure 8–3 Network Printing Between a SunOS 5.8 Print Client and a SunOS 4.1 Print Server

The figure below shows a SunOS 4.1 print client submitting a print request to a SunOS 5.8 print server. The `lpd` daemon handles the local part of the print request and the connection to the print server. On the print server, the network listen process, `inetd`, waits for network printing requests and starts a protocol adaptor to service the request. The protocol adaptor communicates with the `lpsched` daemon, which processes the request on the print server.

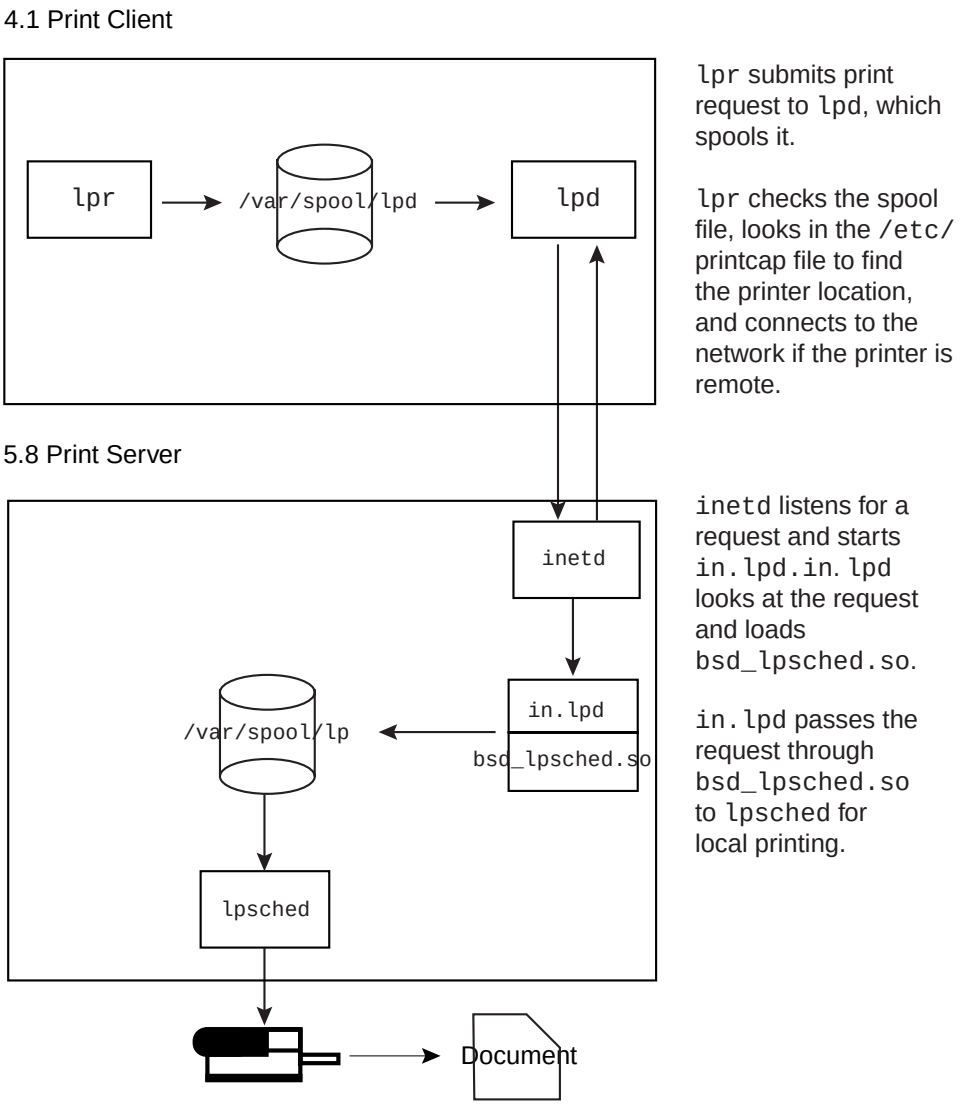


Figure 8–4 Network Printing Between a SunOS 4.1 Print Client and a SunOS 5.8 Print Server

The figure below shows what happens when a user on a SunOS 5.8 print client submits a print request to a SunOS 5.8 print server. The print command on the print client handles the local part of each print request by communicating directly with the print server.

The `inetd` process on the print server monitors network printing requests and starts a protocol adaptor to communicate with the `lpsched` daemon on the print server, which processes the print request.

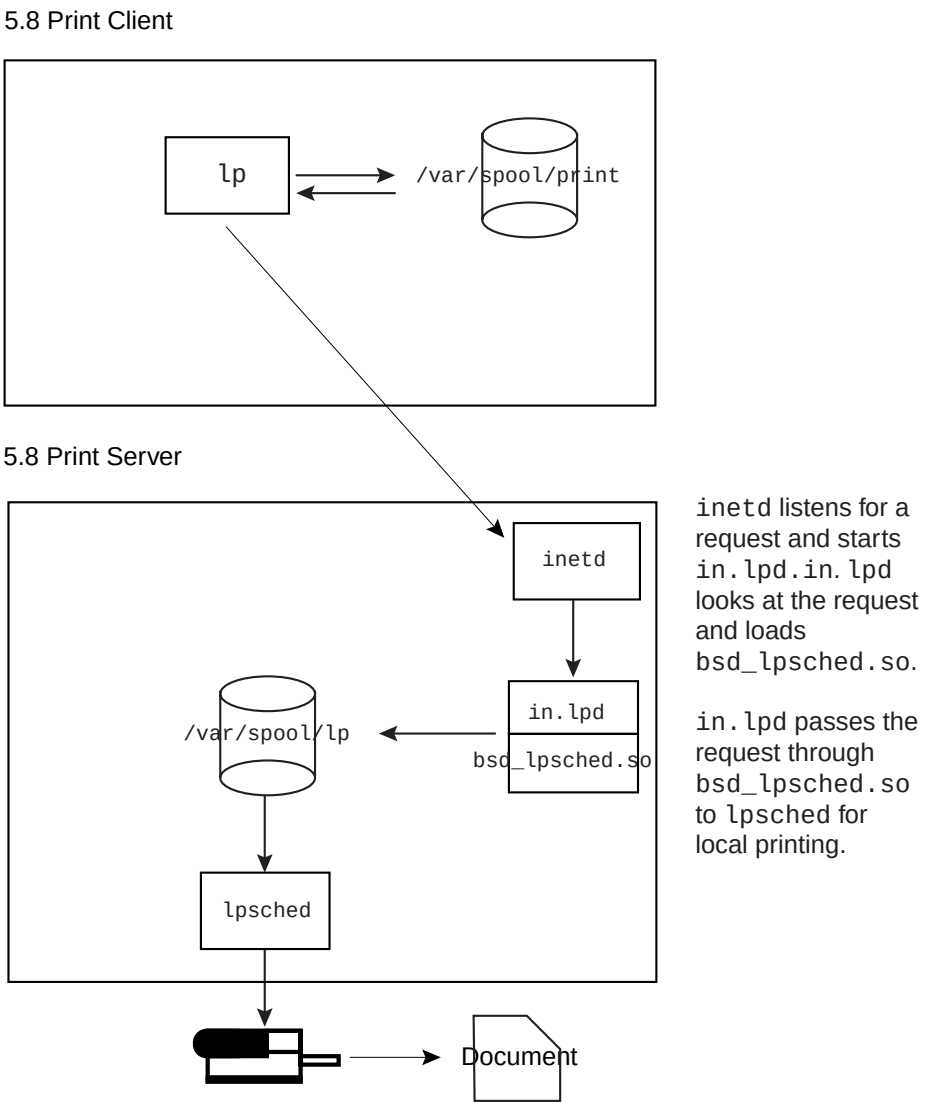


Figure 8–5 Network Printing Between a SunOS 5.8 Print Client and a SunOS 5.8 Print Server

CHAPTER 9

Working With Remote Systems Topics

This section provides instructions for working with remote systems in the Solaris environment. This section contains this chapter.

Chapter 10	Step-by-step instructions for working with remote systems using the `rlogin`, `ftp`, and `rcp` commands, and remote authorization and authentication.

CHAPTER 10

Working With Remote Systems (Tasks)

This chapter describes all the tasks required to log in to remote systems and work with their files. This is a list of the step-by-step instructions in this chapter.

- "How to Search for and Remove `.rhosts` Files" on page 211
- "How to Find Out If a Remote System Is Operating" on page 212
- "How to Find Who Is Logged In to a Remote System" on page 212
- "How to Log In to a Remote System (`rlogin`) " on page 213
- "How to Log Out From a Remote System (`exit`)" on page 214
- "How to Open an `ftp` Connection to a Remote System" on page 216
- "How to Close an `ftp` Connection to a Remote System" on page 217
- "How to Copy Files From a Remote System (`ftp`)" on page 218
- "How to Copy Files to a Remote System (`ftp`)" on page 220
- "How to Copy Files Between a Local and a Remote System (`rcp`)" on page 225

What is a Remote System?

For the purpose of this chapter, a remote system is a workstation or server that is connected to the local system with any type of physical network and configured for TCP/IP communication, shown in the figure below:

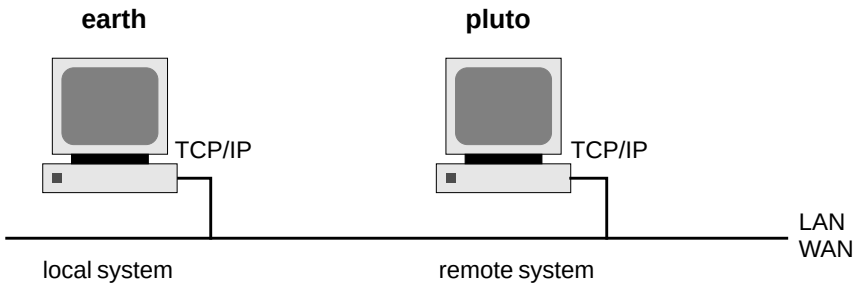


Figure 10–1 A Remote System

On systems running the Solaris release, TCP/IP configuration is established automatically during start-up. For more information, see *System Administration Guide, Volume 3*.

Logging In to a Remote System (`rlogin`)

The `rlogin` command enables you to log in to a remote system. Once logged in, you can navigate through the remote file system and manipulate its contents (subject to authorization), copy files, or execute remote commands.

If the system you are logging into is in a remote domain, be sure to append the domain name to the system name. In this example, `SOLAR` is the name of the remote domain:

```
rlogin pluto.SOLAR
```

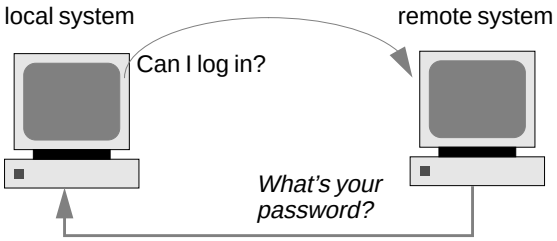
Also, you can interrupt a remote login operation at any time by typing Control-d.

Authentication for Remote Logins (`rlogin`)

Authentication (establishing who you are) for `rlogin` operations can be performed either by the remote system or by the network environment.

The main difference between these forms of authentication lies in the type of interaction they require from you and the way they are established. If a remote system tries to authenticate you, you will be prompted for a password, unless you set up the `/etc/hosts.equiv` or `.rhosts` file. If the network tries to authenticate you, you won't be asked for a password, since the network already knows who you are. The figure below shows a simplified illustration to describe authentication for remote logins.

Authentication by the Remote System



Authentication by the Network

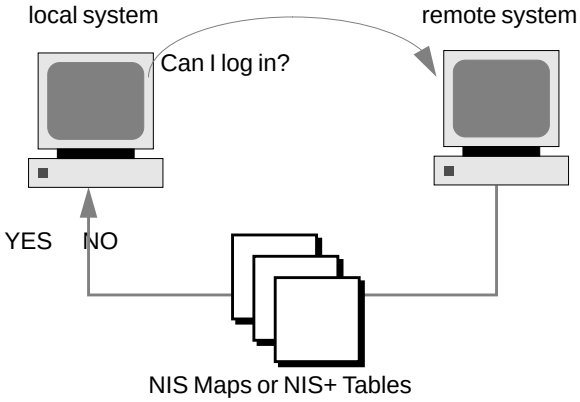


Figure 10–2 Authentication for Remote Logins (`rlogin`)

When the remote system attempts to authenticate you, it relies on information in its local files; specifically if:

- Your system name and user name appears in the remote system's `/etc/hosts.equiv` file, or
- Your system name and user name appears in the remote user's `.rhosts` file, under the remote user's home directory

Network authentication relies on one of these two methods:

- A "trusting network environment" that has been set up with your local network information service and the automounter
- One of the network information services pointed to by the remote system's `/etc/nsswitch.conf` file contains information about you

---

Note - Network authentication generally supersedes system authentication.

The /etc/hosts.equiv File

The /etc/hosts.equiv file contains a list of trusted hosts for a remote system, one per line. If a user attempts to log in remotely (using rlogin) from one of the hosts listed in this file, and if the remote system can access the user's password entry, the remote system allows the user to log in without a password.

A typical hosts.equiv file has the following structure:

```
host1
host2 user_a
+@group1
-@group2
```

When a simple entry for a host is made in hosts.equiv, such as the entry above for host1, it means that the host is trusted, and so is any user at that machine.

If the user name is also mentioned, as in the second entry in the example, then the host is trusted only if the specified user is attempting access.

A group name preceded by a plus sign (+) means that all the machines in that netgroup are considered trusted.

A group name preceded by a minus sign (–) means that none of the machines in that netgroup are considered trusted.

Security Risks When Using the /etc/hosts.equiv File

The /etc/hosts.equiv file presents a security risk. If you maintain a /etc/hosts.equiv file on your system, you should include only trusted hosts in your network. The file should not include any host that belongs to a different network, or any machines that are in public areas. (For example, do not include a host that is located in a terminal room.)

This can create a serious security problem. Either replace the /etc/hosts.equiv file with a correctly configured one, or remove the file altogether.

A single line of + in the /etc/hosts.equiv file indicates that every known host is trusted.

The .rhosts File

The .rhosts file is the user equivalent of the /etc/hosts.equiv file. It contains a list of host-user combinations, rather than hosts in general. If a host-user combination is listed in this file, the specified user is granted permission to log in remotely from the specified host without having to supply a password.

Note that a `.rhosts` file must reside at the top level of a user's home directory. `.rhost` files located in subdirectories are not consulted.

Users can create `.rhosts` files in their home directories. Using the `.rhosts` file is another way to allow trusted access between their own accounts on different systems without using the `/etc/hosts.equiv` file.

Security Risks When Using the `.rhosts` *File*

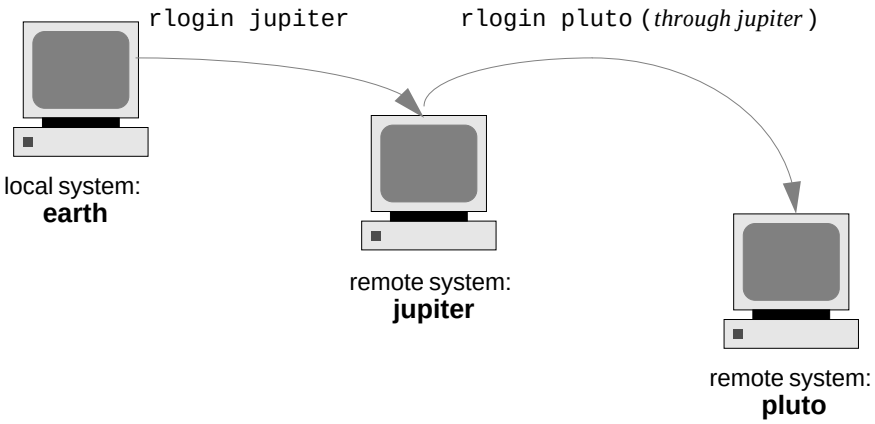
Unfortunately, the `.rhosts` file presents a major security problem. While the `/etc/hosts.equiv` file is under the system administrator's control and can be managed effectively, any user can create a `.rhosts` file granting access to whomever the user chooses without the system administrator's knowledge.

In a situation in which all of the users' home directories are on a single server and only certain people have superuser access on that server, a good way to prevent a user from using a `.rhosts` file is to create an empty file as superuser in their home directory. You would then change the permissions in this file to 000 so that it would be difficult to change it, even as superuser. This would effectively prevent a user from risking system security by using a `.rhosts` file irresponsibly. It would not, however, solve anything if the user is able to change the effective path to his or her home directory.

The only secure way to manage `.rhosts` files is to completely disallow them. See "How to Search for and Remove `.rhosts` Files" on page 211 for detailed instructions. As system administrator, you can check the system often for violations of this policy. One possible exception to this policy is for the root account—you might need to have a `.rhosts` file to perform network backups and other remote services.

Linking Remote Logins

Provided your system is configured properly, you can link remote logins. In this example, a user on `earth` logs in to `jupiter`, and from there decides to log in to `pluto`:



Of course, the user could have logged out of `jupiter` and then logged in directly to `pluto`, but this type of linking can be more convenient.

To link remote logins without having to supply a password, you must have the `/etc/hosts.equiv` or `.rhosts` file set up correctly.

Direct vs. Indirect Remote Logins

The `rlogin` command allows you to log in to a remote system directly or indirectly, as shown in the figure below.

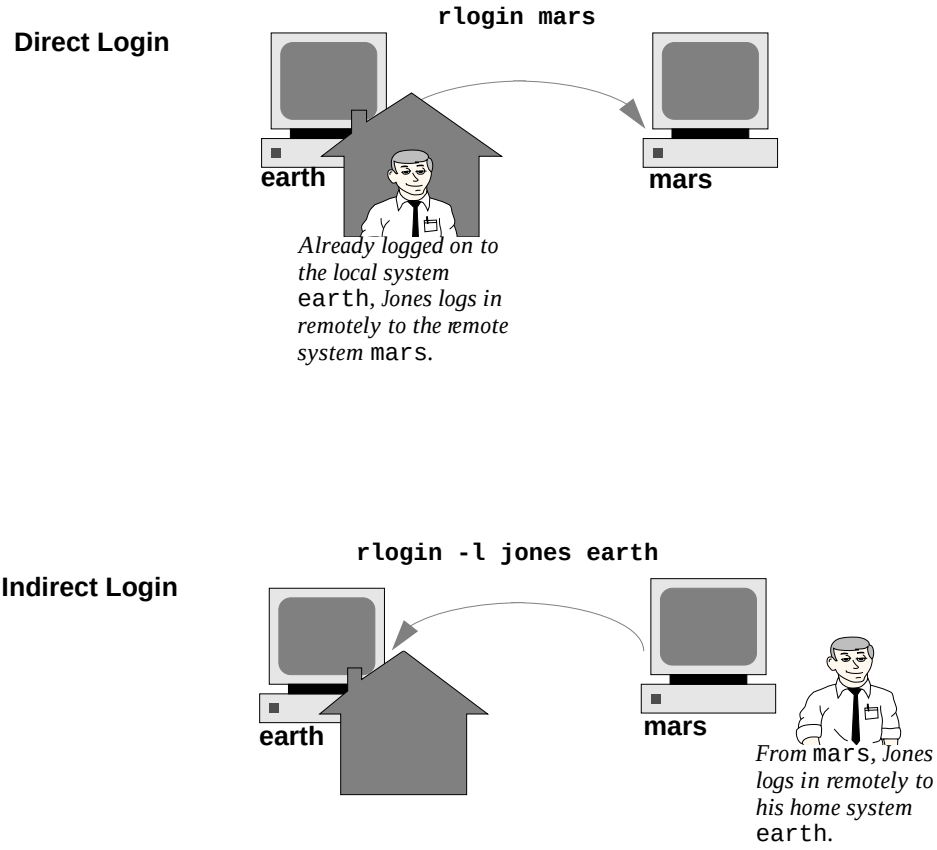


Figure 10–3 Direct and Indirect Logins

A direct remote login is attempted with the default user name; that is, the user name of the individual currently logged in to the local system. This is the most common form of remote login.

An indirect remote login is attempted with a different user name, which is supplied during the remote login operation. This is the type of remote login you might attempt from a workstation that you borrowed temporarily. For instance, if you were in a coworker's office and needed to examine files in your home directory, you might log in to your system remotely, from your coworker's system, but you would perform an indirect remote login, supplying your own user name.

The dependencies between direct and indirect logins and authentication methods are summarized in the table below.

TABLE 10–1 Dependencies Between Login Method and Authentication Method (`rlogin`)

Type of Login	User Name Supplied By	Authentication	Password
Direct	System	Network	None
		System	Required
Indirect	User	Network	None
		System	Required

What Happens After You Log In Remotely

When you log in to a remote system, the `rlogin` command attempts to find your home directory. If the `rlogin` command can't find your home directory, it will assign you to the remote system's root (`/`) directory. For example:

```
Unable to find home directory, logging in with /
```

However, if the `rlogin` command finds your home directory, it sources both your `.cshrc` and `.login` files. Therefore, after a remote login, your prompt is your standard login prompt, and the current directory is the same as when you log in locally.

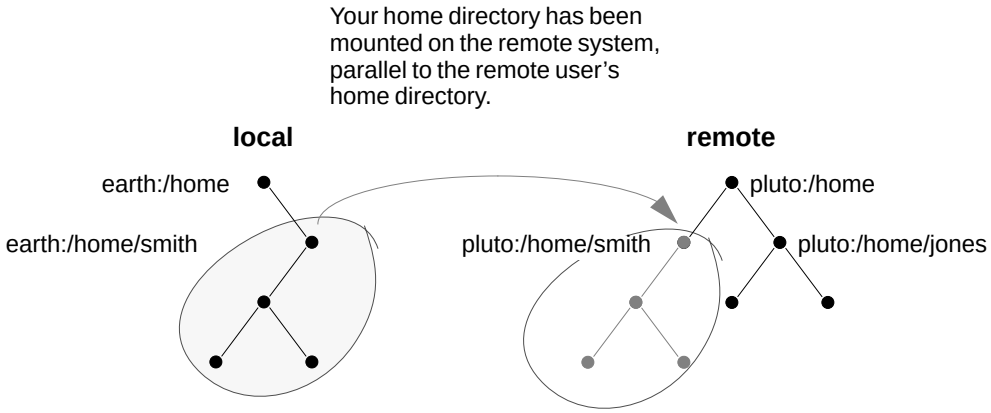
For example, if your usual prompt displays your system name and working directory, and when you log in, your working directory is your home directory, your login prompt looks like this:

```
earth(/home/smith):
```

Then when you log in to a remote system, you will see a similar prompt and your working directory will be your home directory, regardless of the directory from which you entered the `rlogin` command:

```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/smith):
```

The only difference is that the name of the remote system would take the place of your local system at the beginning of the prompt. Where, then, is the remote file system? It is parallel to your home directory, as shown below:



In other words, if you `cd` to `/home` and then run `ls`, this is what you'll see:

```
earth(home/smith): cd ..
earth(/home): ls
smith  jones
```

▼ How to Search for and Remove `.rhosts` Files

1. **Become superuser.**

2. **Search for and remove `.rhosts` files by using the `find`(1) command.**

```
# find home-directories -name .rhosts -print -exec rm{}
```

home-directories	Identifies the path to a directory where users' home directories are located. Note that you can enter multiple paths to search more than one home directory at a time.
`–name .rhosts`	Identifies the filename.
`–print`	Prints the current pathname.
`–exec rm {} \;`	Tells the `find` command to apply the `rm` command to all files identified using the matching filename.

The `find` command starts at the designated directory and searches for any file named `.rhosts`. If it finds any, it prints the path on the screen and removes it.

Example—Searching For and Removing `.rhosts` Files

The following example searches and removes `.rhosts` files in all the user's home directories located in the `/export/home` directory.

```
# find /export/home -name .rhosts -print | xargs -i -t rm{}
```

▼ How to Find Out If a Remote System Is Operating

Find out if a remote system is operating by using the `ping` command.

```
$ ping system-name | ip-address
```

system-name	The name of the remote system.
ip-address	The IP address of the remote system.

The `ping` command returns one of three messages:

Status Message	Explanation
system-name `is alive`	The system can be accessed over the network.
`ping:unknown host` *system-name*	The system name is unknown.
`ping:no answer from` *system-name*	The system is known, but is not currently operating.

If the system you "ping" is located in a different domain, the return message can also contain routing information, which you can ignore.

The `ping` command has a time-out of 20 seconds. In other words, if it does not get a response within 20 seconds, it returns the third message. You can force `ping` to wait longer (or less) by entering a *time-out* value, in seconds:

```
$ ping system-name | ip-address time-out
```

For more information, see `ping`(1M).

▼ How to Find Who Is Logged In to a Remote System

Find who is logged in to a remote system by using the `rusers`(1) command.

```
$ rusers [-l] remote-system-name
```

`rusers`	(No options) Displays the name of the system followed by the name of users currently logged in to it, including root.
`–l`	Displays additional information about each user: the user's login window, login time and date, amount of time logged in, and the name of the remote system from which the user logged on.

Example—Finding Who Is Logged In to a Remote System

The following example shows the short output of `rusers`.

```
$ rusers pluto
pluto    smith  jones
```

In the following example, the long version of `rusers` show that two users are logged in to the remote system `starbug`. The first user logged in from the system console on September 10 and has been logged on for 137 hours and 15 minutes. The second user logged in from a remote system, `mars`, on September 14.

```
$ rusers -l starbug
root         starbug:console           Sep 10 16:13  137:15
rimmer       starbug:pts/0             Sep 14 14:37         (mars)
```

▼ How to Log In to a Remote System (`rlogin`)

Log in to a remote system using the `rlogin`(1) command.

```
$ rlogin [-l user-name] system-name
```

`rlogin`	(No options) Logs you in to the remote system *directly*; in other words, with your current user name.
`–l` *user-name*	Logs you into the remote system *indirectly;* in other words, with the user name you supply.

If the network attempts to authenticate you, you won't be prompted for a password. If the remote system attempts to authenticate you, you will be asked to provide a password.

If the operation succeeds, the `rlogin` command displays brief information about your latest remote login to that system, the version of the operating system running on the remote system, and whether you have mail waiting for you in your home directory.

Example—Logging In to a Remote System (`rlogin`)

The following example shows the output of a direct remote login to `pluto`. The user has been authenticated by the network.

```
$ rlogin starbug
Last login: Mon Jul 12 09:28:39 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

The following example shows the output of an indirect remote login to `pluto`, with the user being authenticated by the remote system.

```
$ rlogin -l smith pluto
password: user-password
Last login: Mon Jul 12 11:51:58 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

▼ How to Log Out From a Remote System (`exit`)

Log out from a remote system by using the `exit`(1) command.

```
$ exit
```

Example—Logging Out From a Remote System (`exit`)

This example shows the user `smith` logging out from the system `pluto`.

```
$ exit
pluto% logout
Connection closed.
earth%
```

Logging In to a Remote System (ftp)

The ftp command opens the user interface to the Internet's File Transfer Protocol. This user interface, called the command interpreter, enables you to log in to a remote system and perform a variety of operations with its file system. The principal operations are summarized in the table below.

The main benefit of ftp over rlogin and rcp is that ftp does not require the remote system to be running UNIX. (The remote system does, however, need to be configured for TCP/IP communications.) On the other hand, rlogin provides access to a richer set of file manipulation commands than ftp does.

Authentication for Remote Logins (ftp)

Authentication for ftp remote login operations can be established either by:

- Including your password entry in the remote system's /etc/passwd file or equivalent network information service map or table.
- Establishing an anonymous ftp account on the remote system.

Essential ftp Commands

TABLE 10–2 Essential ftp Commands

Command	Description
ftp	Accesses the ftp command interpreter
ftp *remote-system*	Establishes an ftp connection to a remote system. For instructions, see "How to Open an ftp Connection to a Remote System" on page 216
open	Logs in to the remote system from the command interpreter
close	Logs out of the remote system and returns to the command interpreter

TABLE 10–2 Essential `ftp` Commands *(continued)*

Command	Description
`bye`	Quits the `ftp` command interpreter
`help`	Lists all `ftp` commands or, if a command name is supplied, briefly describes what the command does
`reset`	Re-synchronizes the command-reply sequencing with the remote `ftp` server
`ls`	Lists the contents of the remote working directory
`pwd`	Displays the name of the remote working directory
`cd`	Changes the remote working directory
`lcd`	Changes the local working directory
`mkdir`	Creates a directory on the remote system
`rmdir`	Deletes a directory on the remote system
`get, mget`	Copies a file (or multiple files) from the remote working directory to the local working directory
`put, mput`	Copies a file (or multiple files) from the local working directory to the remote working directory
`delete, mdelete`	Deletes a file (or multiple files) from the remote working directory

For more information, see `ftp`(1).

▼ How to Open an `ftp` Connection to a Remote System

1. **Make sure you have `ftp` authentication.**

   You must have `ftp` authentication, as described in "Authentication for Remote Logins (`ftp`)" on page 215.

2. **Open a connection to a remote system by using the `ftp` command.**

```
$ ftp remote-system
```

If the connection succeeds, a confirmation message and prompt is displayed.

3. **Enter your user name.**

```
Name (remote-system:user-name): user-name
```

4. **If prompted, enter your password.**

```
331 Password required for user-name:
Password: password
```

If the system you are accessing has established an anonymous `ftp` account, you will not be prompted for a password. If the `ftp` interface accepts your password, it displays a confirmation message and the (`ftp>`) prompt.

You can now use any of the commands supplied by the `ftp` interface, including help. The principal commands are summarized in Table 10–2.

Example—Opening an `ftp` Connection to a Remote System

This `ftp` session was established by the user `smith` on the remote system `pluto`:

```
$ ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:smith): smith
331 Password required for smith:
Password: password
230 User smith logged in.
ftp>
```

▼ How to Close an `ftp` Connection to a Remote System

Close an `ftp` connection to a remote system by using the `bye` command.

```
ftp> bye
221 Goodbye.
earth%
```

A good-bye message appears, followed by your usual shell prompt.

▼ How to Copy Files From a Remote System (ftp)

1. **Change to a directory on the local system where you want the files from the remote system to be copied.**

   ```
   $ cd target-directory
   ```

2. **Establish an ftp connection.**

   See "How to Open an ftp Connection to a Remote System" on page 216.

3. **Change to the source directory.**

   ```
   ftp> cd source-directory
   ```

   If your system is using the automounter, the home directory of the remote system's user appears parallel to yours, under /home.

4. **Make sure you have read permission for the source files.**

   ```
   ftp> ls -l
   ```

5. **To copy a single file, use the get command.**

   ```
   ftp> get filename
   ```

6. **To copy multiple files at once, use the mget command.**

   ```
   ftp> mget filename [filename ...]
   ```

   You can supply a series of individual file names and you can use wildcard characters. The mget command will copy each file individually, asking you for confirmation each time.

7. **Close the ftp connections.**

   ```
   ftp> bye
   ```

Examples—Copying Files From a Remote System (ftp)

In this example, the user kryten opens an ftp connection to the system pluto, and uses the get command to copy a single file from the /tmp directory:

```
$ cd $HOME
ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34344)
(0 bytes).
dtdbcache_:0
filea
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
53 bytes received in 0.022 seconds (2.39 Kbytes/s)
ftp> get filea
200 PORT command successful.
150 ASCII data connection for filea (129.152.221.238,34331)
(0 bytes).
226 ASCII Transfer complete.
ftp> bye
221 Goodbye.
```

In this example, the same user kryten uses the mget command to copy a set of files from the /tmp directory to his home directory. Note that kryten can accept or reject individual files in the set.

```
$ ftp> cd /tmp
250 CWD command successful.
ftp> ls files
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34345)
(0 bytes).
fileb
filec
filed
226 ASCII Transfer complete.
remote: files
21 bytes received in 0.015 seconds (1.36 Kbytes/s)
ftp> cd files
250 CWD command successful.
ftp> mget file*
```

(continued)

```
mget fileb? y
200 PORT command successful.
150 ASCII data connection for fileb (129.152.221.238,34347)
(0 bytes).
226 ASCII Transfer complete.
mget filec? y
200 PORT command successful.
150 ASCII data connection for filec (129.152.221.238,34348)
(0 bytes).
226 ASCII Transfer complete.
mget filed? y
200 PORT command successful.
150 ASCII data connection for filed (129.152.221.238,34351)
(0 bytes).
226 ASCII Transfer complete.200 PORT command successful.
ftp> bye
221 Goodbye.
```

▼ How to Copy Files to a Remote System (`ftp`)

1. **Change to the source directory on the local system.**

   The directory from which you enter the `ftp` command will be the local working directory, and thus the source directory for this operation.

2. **Establish an `ftp` connection.**

   See "How to Open an `ftp` Connection to a Remote System" on page 216.

3. **Change to the target directory.**

   ```
   ftp> cd target-directory
   ```

   Remember, if your system is using the automounter, the home directory of the remote system's user appears parallel to yours, under `/home`.

4. **Make sure you have write permission to the target directory.**

   ```
   ftp> ls -l target-directory
   ```

5. **To copy a single file, use the `put` command.**

   ```
   ftp> put filename
   ```

6. **To copy multiple files at once, use the** `mput` **command.**

```
ftp> mput filename [filename ...]
```

You can supply a series of individual file names and you can use wildcard characters. The `mput` command will copy each file individually, asking you for confirmation each time.

7. **To close the** `ftp` **connection, type** `bye`**.**

```
ftp> bye
```

Examples—Copying Files to a Remote System (`ftp`)

In this example, the user `kryten` opens an `ftp` connection to the system `pluto`, and uses the `put` command to copy a file from his system to the `/tmp` directory on system `pluto`:

```
$ cd /tmp
ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> put filef
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
226 Transfer complete.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34357) (0 bytes).
dtdbcache_:0
filea
filef
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
60 bytes received in 0.058 seconds (1.01 Kbytes/s)
ftp> bye
221 Goodbye.
```

In this example, the same user `kryten` uses the `mput` command to copy a set of files from his home directory to `pluto`'s `/tmp` directory. Note that `kryten` can accept or reject individual files in the set.

```
$ cd $HOME/testdir
$ ls
test1   test2   test3
$ ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> mput test*
mput test1? y
200 PORT command successful.
150 ASCII data connection for test1 (129.152.221.238,34365).
226 Transfer complete.
mput test2? y
200 PORT command successful.
150 ASCII data connection for test2 (129.152.221.238,34366).
226 Transfer complete.
mput test3? y
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
226 Transfer complete.
ftp> bye
221 Goodbye.
```

Remote Copying With `rcp`

The `rcp` command copies files or directories between a local and a remote system or between two remote systems. You can use it from a remote system (after logging in with the `rlogin` command) or from the local system (without logging in to a remote system).

With `rcp`, you can perform the following remote copy operations:

- Copy a file or directory from your system to a remote system
- Copy a file or directory from a remote system to your local system
- Copy a file or directory between remote systems from your local system

If you have the automounter running, you can perform these remote operations with the `cp` command. However, the range of `cp` is constrained to the virtual file system created by the automounter and to operations relative to a user's home directory and, since `rcp` performs the same operations without these constraints, this section will describe only the `rcp` versions of these tasks.

Security Considerations for Copy Operations

To copy files or directories between systems, you must have permission to log in and copy files.

Caution - Both the `cp` and `rcp` commands can overwrite files without warning. Make sure file names are correct before executing the command.

Specifying Source and Target

With the `rcp` command in the C-shell, you can specify source (the file or directory you want to copy) and target (the location into which you will copy the file or directory) with either absolute or abbreviated pathnames.

	Absolute Pathnames	**Abbreviated Pathnames**
From Local System	`mars:/home/jones/myfile.txt`	`~jones/myfile.txt`
After Remote Login	`/home/jones/myfile.txt`	`~jones/myfile.txt`

Absolute pathnames identify files or directories mounted on a particular system. In the example above, the first absolute pathname identifies a file (`MyFile.txt`) on the `mars` system. Abbreviated pathnames identify files or directories relative to a user's home directory, wherever that might reside. In the first example above, the abbreviated pathname identifies the same file, `MyFile.txt`, but uses "~" symbol to indicate the `jones` home directory. In effect . . .

```
~ = mars:/home/jones
```

The examples on the second line, above, demonstrate the user of absolute and abbreviated pathnames after a remote login. There is no difference for the abbreviated pathname, but because the remote login operation mounted the `jones` home directory onto the local system (parallel to the local user's home directory), the absolute pathname no longer requires the system name `mars`. For more information about how a remote login operation mounts another user's home directory, see "What Happens After You Log In Remotely" on page 210.

The table below provides a sample of absolute and abbreviated pathnames recognized by the C shell. It uses the following terminology:

- Working directory - The directory from which the `rcp` command is entered. Can be remote or local.
- Current user - The user name under which the `rcp` command is entered.

TABLE 10–3 Allowed Syntaxes for Directory and File Names

Logged in to	Syntax	Description
Local system	.	The local working directory
	path/filename	The *path* and *filename* in the local working directory
	~	The current user's home directory
	~/path/filename	The *path* and *filename* beneath the current user's home directory
	~user	The home directory of *user*
	~user/path/filename	The *path* and *filename* beneath the home directory of *user*
	remote-system:path/filename	The *path* and *filename* in the remote working directory
Remote system	.	The remote working directory
	filename	The *filename* in the remote working directory
	path/filename	The *path* and *filename* in the remote working directory
	~	The current user's home directory
	~/path/filename	The *path* and *filename* in the current user's home directory
	~user	The home directory of *user*
	~/user/path/filename	The *path* and *filename* beneath the home directory of *user*
	local-system:path/filename	The *path* and *filename* in the local working directory

▼ How to Copy Files Between a Local and a Remote System (rcp)

1. **Be sure you have permission to copy.**

   You should at least have read permission on the source system and write permission on the target system.

2. **Determine the location of the source and target.**

   If you don't know the path of the source or target, you can first log into the remote system with the `rlogin` command, as described in "How to Log In to a Remote System (`rlogin`) " on page 213. Then, navigate through the remote system until you find the location. You can then perform the next step without logging out.

3. **Copy the file or directory.**

```
$ rcp [-r] source-file | directory      target-file | directory
```

`rcp`	(No options) Copies a single file from the source to the target.
`–r`	Copies a directory from the source to the target.

   This syntax applies whether you are logged in to the remote system or in to the local system. Only the pathname of the file or directory changes, as described in Table 10–3 and as illustrated in the examples below.

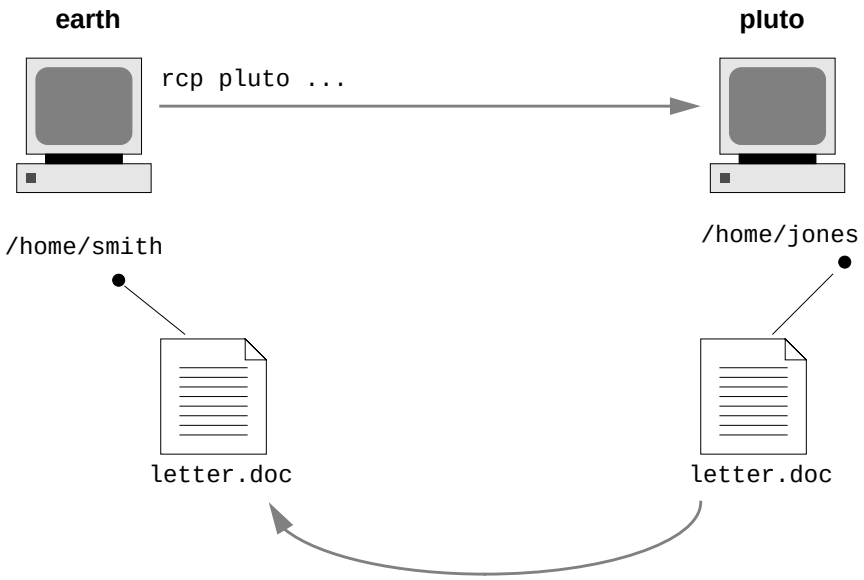
   You can use the "`~`" and "`.`" characters to specify the path portions of the local file or directory names. Note, however, that "`~`" applies to the current user, not the remote system, and that "`.`" applies to system you are logged into. For explanations of these symbols, see Table 10–3.

Examples—Copying Files Between a Local and a Remote System (rcp)

Here are a few examples. In the first two, the source is remote; in the last two, the source is local.

In this example, `rcp` copies the file `letter.doc` from the `/home/jones` directory of the remote system `pluto` to the working directory (`/home/smith`) on the local system, `earth`:

```
earth(/home/smith): rcp pluto:/home/jones/letter.doc .
```



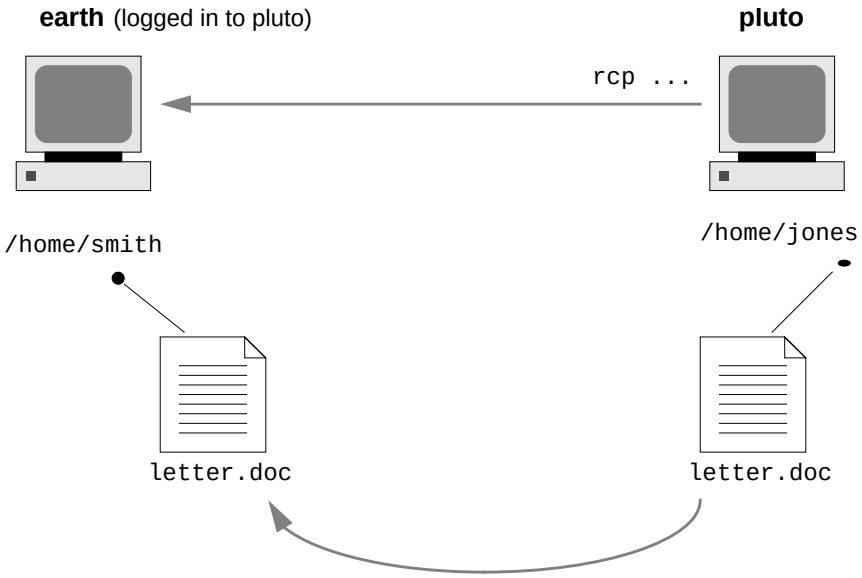
Since the `rcp` operation is performed without a remote login, the "`.`" symbol applies to the local system, not the remote system.

The working directory happens to be the local user's home directory, so it could have been specified with the "`~`" symbol as well:

```
earth(home/smith): rcp pluto:/home/jones/letter.doc ~
```

In the following example, `rcp` is used —while logged in to the remote system— to perform the same operation. Although the flow of the operation is the same, the paths change to take into account the remote login:

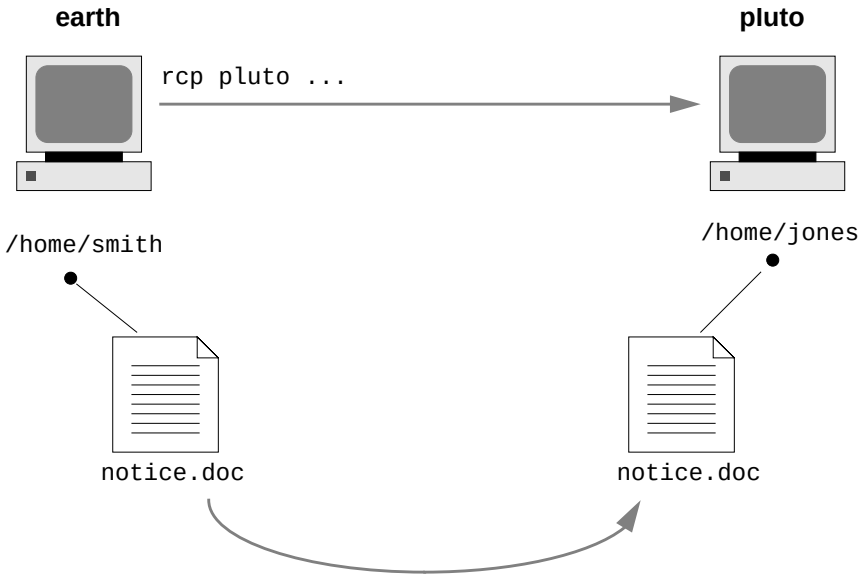
```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/jones): rcp letter.doc ~
```



Use of the "." symbol would be inappropriate in this instance because of the remote login; it would simply apply to the remote system, essentially directing `rcp` to create a duplicate file. The "~" symbol, however, refers to the current user's home directory, even when logged in to a remote system.

In the following example, `rcp` copies the file `notice.doc` from the home directory (`/home/smith`) of the local system `earth` to the `/home/jones` directory of the remote system, `pluto`:

```
earth(/home/smith): rcp notice.doc pluto:/home/jones
```



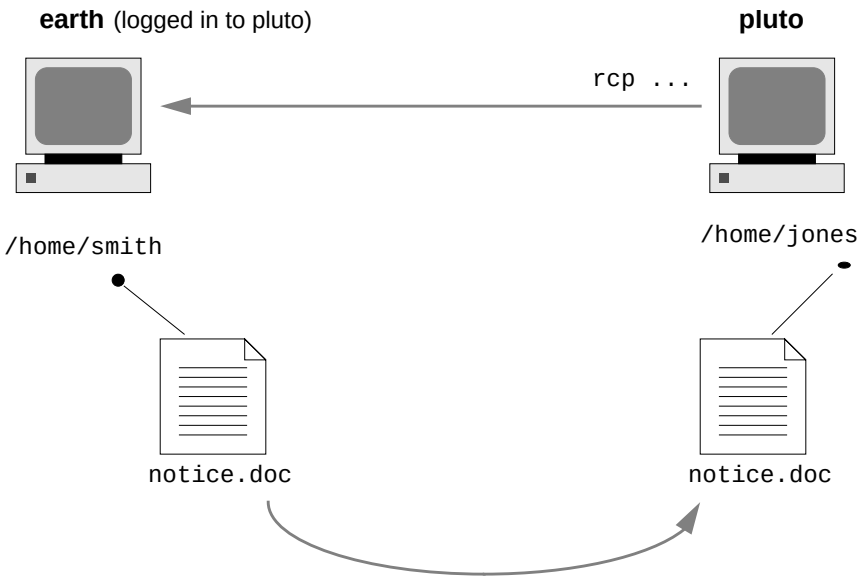
Because no remote filename is provided, the file `notice.doc` is copied into the `/home/jones` directory with the same name.

In this example, the operation is repeated, but `rcp` is entered from a different working directory on the local system (`/tmp`). Note the use of the "~" symbol to refer to the current user's home directory:

```
earth(/tmp): rcp ~/notice.doc pluto:/home/jones
```

In this example, `rcp` is used —while logged in to the remote system— to perform the same operation as in the previous example. Although the flow of the operation is the same, the paths change the take into account the remote login:

```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/jones): rcp ~/notice.doc .
```



In this instance, the "~" symbol can be used to denote the current user's home directory, even though it is on the local system. The "." symbol refers to the working directory on the remote system because the user is logged in to the remote system. Here is an alternative syntax that performs the same operation:

```
pluto(/home/jones): rcp earth:/home/smith/notice.doc /home/jones
```

CHAPTER 11

Managing Terminals and Modems Topics

This section provides instructions for managing terminals and modems. This section contains these chapters.

Chapter 12	Provides overview information about terminals and modems.
Chapter 13	Provides step-by-step instructions for setting up terminals and modems.
Chapter 14	Provides step-by-step instructions for using SAF commands to set up terminals and modems.

CHAPTER 12

Managing Terminals and Modems (Overview)

This chapter provides the overview information for managing terminals and modems. This is a list of the overview information in this chapter.

- "Terminals, Modems, Ports, and Services" on page 233
- "Tools for Managing Terminals and Modems" on page 236
- "Admintool" on page 237
- "Service Access Facility" on page 238

For step-by-step instructions about how to set up terminals and modems with Admintool, see Chapter 13.

For step-by-step instructions about how to set up terminals and modems with the Service Access Facility (SAF), see Chapter 14.

Terminals, Modems, Ports, and Services

Terminals and modems provide both local and remote access to system and network resources. Setting up terminals and modem access is an important responsibility of a system administrator. This section explains some of the concepts behind modem and terminal management in the Solaris environment.

Terminals

Your system's bit-mapped graphics display is not the same as an alphanumeric terminal, which connects to a serial port and displays only text. You don't have to perform any special steps to administer the graphics display.

Modems

Modems can be set up in three basic configurations:

- Dial-out
- Dial-in
- Bidirectional

A modem connected to your home computer might be set up to provide *dial-out* service, meaning you can access other computers from your own home, but nobody outside can gain access to your machine.

Dial-in service is just the opposite. It allows people to access a system from remote sites, but it does not permit calls to the outside world.

Bidirectional access, as the name implies, provides both dial-in and dial-out capabilities.

Ports

A *port* is a channel through which a device communicates with the operating system. From a hardware perspective, a port is a "receptacle" into which a terminal or modem cable might be plugged.

However, a port is not strictly a physical receptacle, but an entity with hardware (pins and connectors) and software (a device driver) components. A single physical receptacle often provides multiple ports, allowing connection of two or more devices.

Common types of ports include serial, parallel, small computer systems interface (SCSI), and Ethernet.

A *serial* port, using a standard communications protocol, transmits a byte of information bit-by-bit over a single line.

Devices that have been designed according to RS-232-C or RS-423 standards (this includes most modems, alphanumeric terminals, plotters, and some printers) can be plugged interchangeably (using standard cables) into serial ports of computers that have been similarly designed.

When many serial port devices must be connected to a single computer, it might be necessary to add an *adapter board* to the system. The adapter board, with its driver

software, provides additional serial ports for connecting more devices than could otherwise be accommodated.

Services

Modems and terminals gain access to computing resources via the serial port software. The serial port software must be set up to provide a particular “service” for the device attached to the port. For example, you can set up a serial port to provide bidirectional service for a modem.

Port Monitors

The main mechanism for gaining access to a service is through a *port monitor*. A port monitor is a program that continuously monitors for requests to log in or access printers or files.

When a port monitor detects a request, it sets whatever parameters are required to establish communication between the operating system and the device requesting service. Then the port monitor transfers control to other processes that provide the services needed.

The table below describes the two types of port monitors included in the Solaris environment.

TABLE 12–1 Port Monitor Types

Port Monitor	Description
`listen`(1M)	Controls access to network services, such as handling remote print requests prior to the Solaris 2.6 release. The default Solaris operating environment no longer uses this port monitor type.
`ttymon`(1M)	Provides access to the login services needed by modems and alphanumeric terminals. Admintool automatically sets up a `ttymon` port monitor to process login requests from these devices.

You might be familiar with an older port monitor called `getty`(1M). The new `ttymon` is more powerful; a single `ttymon` can replace multiple occurrences of `getty`. Otherwise, these two programs serve the same function.

Tools for Managing Terminals and Modems

The table below lists the recommended tools for managing terminals and modems. Table 12–3 lists specific differences in functionality between the Service Access Facility (SAF) and Admintool: Serial Ports.

TABLE 12–2 Recommended Tools For Managing Terminals and Modems

If You Want The Tool That Is ...	Then Use ...	To Start This Tool See ...
The most comprehensive	Service Access Facility (SAF) commands	"Service Access Facility" on page 238
The quickest setup	Admintool graphical user interface (for local systems only)	Chapter 13

TABLE 12–3 Functionality Differences Between Admintool and Service Access Facility

If You Need To ...	Then Use ...	Comment
Inform users that a port is disabled	Service Access Facility `ttyadm -i`	`ttyadmin -i` specifies the inactive (disabled) response message. The message is sent to a terminal or modem when a user attempts to log in when the port is disabled. This functionality is not provided when a port is disabled using Admintool.
Keep the modem connection when a user logs off a host	Service Access Facility `ttyadm -h`	`ttyadm -h` specifies that the system will not hang up on a modem before setting or resetting to the default or specified value. If `ttyadm -h` is not used, when the user logs out of a host, the host will hang up the modem.
Require the user to type a character before the system displays a prompt	Service Access Facility `ttyadm -r`	`ttyadm -r` specifies that `ttymon` should require the user to type a character or press Return a specified number of times before the login prompt appears. When `-r` is not specified, pressing Return one or more times will print the prompt anyway. This option prevents a terminal server from issuing a welcome message that the Solaris host might misinterpret to be a user trying to log in. Without the `-r` option, the host and terminal server might begin looping and printing prompts to each other.

Admintool

Admintool: Serial Ports sets up the serial port software to work with terminals and modems by calling the `pmadm` command with the appropriate information. It also provides:

- Templates for common terminal and modem configurations
- Multiple port setup, modification, or deletion
- Quick visual status of each port

Service Access Facility

The SAF is the tool used for administering terminals, modems, and other network devices. In particular, SAF enables you to set up:

- `ttymon` and `listen` port monitors (using the `sacadm` command)
- `ttymon` port monitor services (using the `pmadm` and `ttyadm` commands)
- `listen` port monitor services (using the `pmadm` and `nlsadmin` commands)
- And troubleshoot `tty` devices
- And troubleshoot incoming network requests for printing service
- And troubleshoot the Service Access Controller (using the `sacadm` command)

The SAF is an open-systems solution that controls access to system and network resources through `tty` devices and local-area networks (LANs). SAF is not a program. It is a hierarchy of background processes and administrative commands.

CHAPTER 13

Setting Up Terminals and Modems (Tasks)

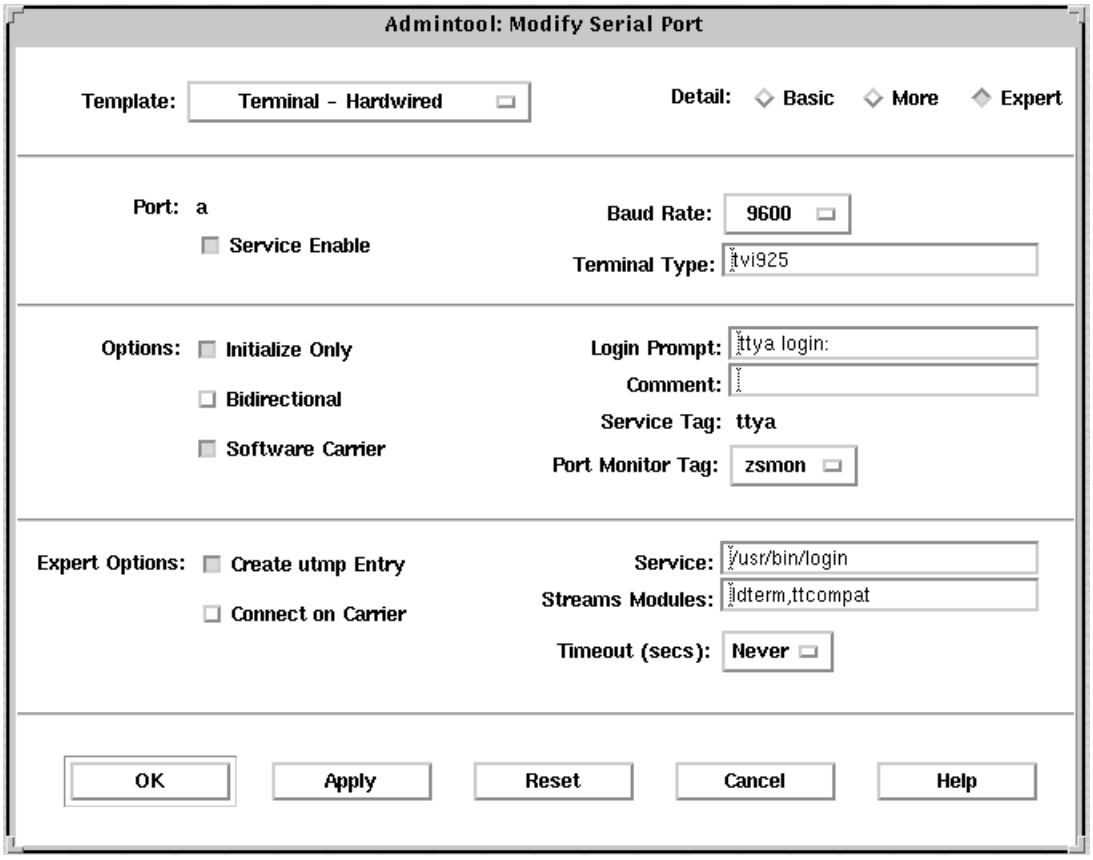
This chapter provides step-by-step instructions for setting up terminals and modems using Admintool. This is a list of the step-by-step instructions in this chapter.

- "How to Start Admintool" on page 245
- "How to Set Up a Terminal" on page 245
- "How to Set Up a Modem" on page 247
- "How to Set Up a Modem for Use With UUCP" on page 249
- "How to Initialize a Port" on page 250
- "How to Disable a Port" on page 251
- " How to Remove a Port Service" on page 252

For overview information about terminals and modems, see Chapter 12.

Setting Up Terminals and Modems

When setting up serial port information, start Admintool, and select Serial Ports from the Browse menu. Select a serial port from the Serial Ports window and then choose Modify from the Edit menu to bring up the Modify Serial Port window. This window provides access to the port templates and provides information on the port in three levels of detail—Basic, More, and Expert.



Note - The Modify Serial Port window appears in the Basic detail mode. To view More or Expert details, select the More or Expert option from the Detail menu.

The descriptions of each item in the Modify Serial window are listed in the table below.

TABLE 13–1 Modify Serial Port Window Items

Detail	Item	Description
Basic	Port	Lists the port or ports you selected from Serial Ports main window.
	Service	Specifies that the service for the port is turned on (enabled).

TABLE 13–1 Modify Serial Port Window Items *(continued)*

Detail	Item	Description
	Baud Rate	Specifies the line speed used to communicate with the terminal. The line speed represents an entry in the `/etc/ttydefs` file.
	Terminal Type	Shows the abbreviation for the type of terminal, for example, ansi or vt100. Similar abbreviations are found in `/etc/termcap`. This value is set in the `$TERM` environment variable.
More	Option: Initialize Only	Specifies that the port software is initialized but not configured.
	Option: Bidirectional	Specifies that the port line is used in both directions.
	Option: Software Carrier	Specifies that the software carrier detection feature is used. If the option is *not* checked, the *hardware* carrier detection signal is used.
	Login Prompt	Shows the prompt displayed to a user after a connection is made.
	Comment	Shows the comment field for the service.
	Service Tag	Lists the service tag associated with this port—typically an entry in the `/dev/term` directory.
	Port Monitor Tag	Specifies the name of the port monitor to be used for this port. Note: The default monitor is typically correct.
Expert	Create `utmpx` Entry	Specifies that a `utmpx` entry is created in the accounting files upon login. Note: This item must be selected if a login service is used. See the Service item.
	Connect on Carrier	Specifies that a port's associated service is invoked immediately when a connect indication is received.
	Service	Shows the program that is run upon connection.
	Streams Modules	Shows the STREAMS modules that are pushed before the service is invoked.
	Timeout (secs)	Specifies the number of seconds before a port is closed if the open process on the port succeeds and no input data is received.

Setting Up Terminals

The table below describes the menu items (and their default values) when setting up a terminal using Serial Ports.

TABLE 13–2 Terminal - Hardwired Default Values

Detail	Item	Default Value
Basic	Port	—
	Service	Enabled
	Baud Rate	`9600`
	Terminal Type	—
More	Option: Initialize Only	no
	Option: Bidirectional	no
	Option: Software Carrier	yes
	Login Prompt	`login:`
	Comment	Terminal - Hardwired
	Service Tag	—
	Port Monitor Tag	`zsmon`
Expert	Create `utmpx` Entry	yes
	Connect on Carrier	no
	Service	`/usr/bin/login`
	Streams Modules	`ldterm,ttcompat`
	Timeout (secs)	Never

Setting Up Modems

The table below describes the three modem templates available when setting up a modem using Serial Ports.

TABLE 13–3 Modem Templates

Modem Configuration	Description
Dial-In Only	Users can dial in to the modem but cannot dial out.
Dial-Out Only	Users can dial out from the modem but cannot dial in.
Bidirectional	Users can either dial in or out from the modem.

The table below describes the default values of each template.

TABLE 13–4 Modem Template Default Values

Detail	Item	Modem - Dial-In Only	Modem - Dial-Out Only	Modem - Bidirectional
Basic	Port	—	—	—
	Service	Enabled	Enabled	Enabled
	Baud Rate	9600	`9600`	`9600`
	Terminal Type	—	—	—
More	Option: Initialize Only	yes	no	no
	Option: Bidirectional	no	no	yes
	Option: Software Carrier	no	no	no
	Login Prompt	`login:`	`login:`	`login:`
	Comment	Modem - Dial-In Only	Modem - Dial-Out Only	Modem - Bidirectional

TABLE 13–4 Modem Template Default Values *(continued)*

Detail	Item	Modem - Dial-In Only	Modem - Dial-Out Only	Modem - Bidirectional
	Service Tag	—	—	—
	Port Monitor Tag	`zsmon`	`zsmon`	`zsmon`
Expert	Create `utmpx` Entry	yes	yes	yes
	Connect on Carrier	no	no	no
	Service	`/usr/bin/login`	`/usr/bin/login`	`/usr/sbin/login`
	Streams Modules	`ldterm,ttcompat`	`ldterm,ttcompat`	`ldterm,ttcompat`
	Timeout (secs)	Never	Never	Never

The table below describes the default values for the Initialize Only template.

TABLE 13–5 Initialize Only - No Connection Default Values

Detail	Item	Default Value
Basic	Port	—
	Service	Enabled
	Baud Rate	9600
	Terminal Type	—
More	Option: Initialize Only	yes
	Option: Bidirectional	no
	Option: Software Carrier	no
	Login Prompt	`login:`
	Comment	Initialize Only - No Connection

TABLE 13–5 Initialize Only - No Connection Default Values *(continued)*

Detail	Item	Default Value
	Service Tag	—
	Port Monitor Tag	`zsmon`
Expert	Create `utmpx` Entry	yes
	Connect on Carrier	no
	Service	`/usr/bin/login`
	Streams Modules	`ldterm`,`ttcompat`
	Timeout (secs)	Never

▼ How to Start Admintool

1. **Verify that the following prerequisites are met. To use Admintool, you must:**
   - Have a bit-mapped display monitor. The Admintool software can be used only on a system with a console that is a bit-mapped screen such as a standard display monitor that comes with a Sun workstation.
   - Be running an X Window System, such as the CDE environment.
   - Be a member of the `sysadmin` group (group 14).

   If you want to perform administration tasks on a system with an ASCII terminal as the console, use Solaris commands instead.

2. **Start Admintool.**

```
$ admintool &
```

   The Users main window is displayed.

▼ How to Set Up a Terminal

1. **Start Admintool, if it's not already running.**

   See "How to Start Admintool" on page 245 for more information on starting Admintool.

2. **Select Serial Ports from the Browse menu.**

   The Serial Ports menu is displayed.

3. **Select the port or ports that will be used with a terminal.**

4. **Choose Modify from the Edit menu.**

   The Modify Serial Port window appears in the Basic Detail mode. To enter additional details, select either the More or Expert Detail modes.

5. **Choose Terminal-Hardwired from the Use Template menu.**

   See Table 13–2 for a description of the Terminal–Hardware menu items.

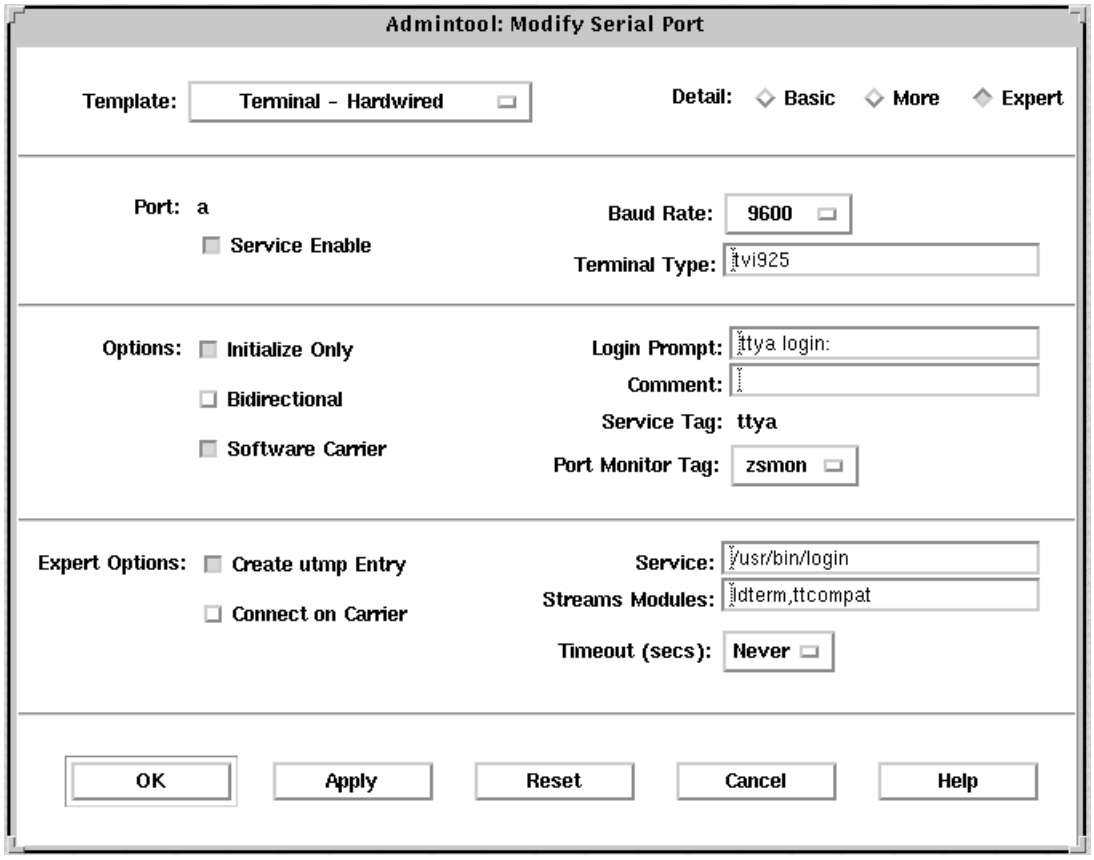
6. **Change values of template entries if desired.**

7. **Click on OK to configure the port.**

8. **Use the `pmadm` command to verify the terminal service has been added.**

```
$ pmadm -l -s ttya
```

Example—Completed Modify Window to Set Up a Terminal



▼ How to Set Up a Modem

1. **Start Admintool, if it's not already running.**

   See "How to Start Admintool" on page 245 for more information on starting Admintool.

2. **Select Serial Ports from the Browse menu.**

   The Serial Ports menu is displayed.

3. **Select the port or ports that will be used with a modem.**

4. **Choose Modify from the Edit menu.**

   The Modify Serial Port window appears in the Basic Detail mode. To enter additional details, select either the More or Expert Detail modes.

5. **Choose the modem configuration template from the Use Template menu that meets or most closely matches your modem service.**

   See Table 13–3 for a description of each template.

   See Table 13–4 for the default values of each template. If a UUCP service will be used to dial in to your modem on a Solaris system, see “How to Set Up a Modem for Use With UUCP” on page 249 for the rest of the procedure.

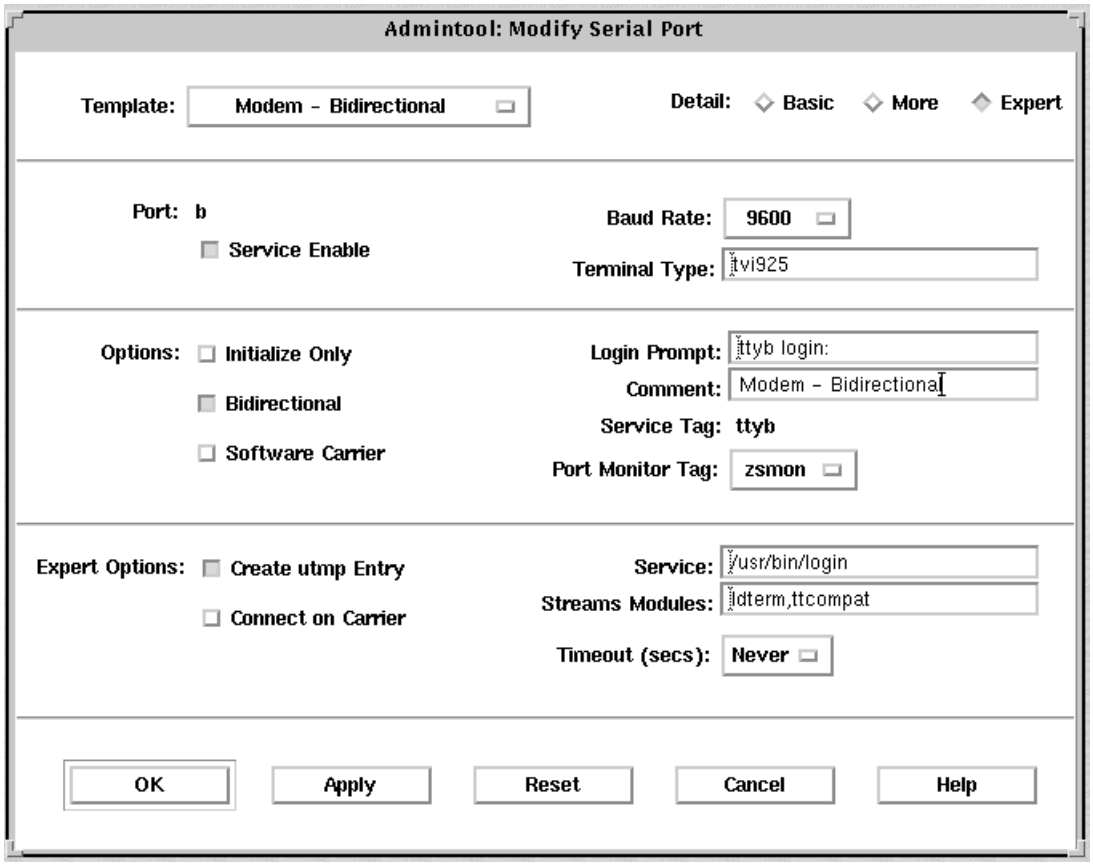
6. **Change values of template entries if desired.**

7. **Click on OK to configure the port.**

8. **Use the** `pmadm` **command to verify the modem service has been configured for use with UUCP.**

```
$ pmadm -l -s ttyb
```

Example—Completed Modify Window to Set Up a Modem



▼ How to Set Up a Modem for Use With UUCP

UUCP sends information to a service using seven bits and even parity. Solaris modem configurations use eight bits and no parity for internationalization purposes. To set up your modem service to work with UUCP, follow these instructions.

1. **Start Admintool, if it's not already running.**

   See "How to Start Admintool" on page 245 for more information on starting Admintool.

2. **Select Serial Ports from the Browse menu.**

   The Serial Ports menu is displayed.

3. **Select the port or ports that will be used with a modem.**

4. **Choose Modify from the Edit menu.**

The Modify Serial Port window appears in the Basic Detail mode. For additional details, select either the More or Expert Detail modes.

5. **Select Other from the Baud Rate menu.**

   A window listing baud rates from the `/etc/ttydefs` file is displayed.

6. **Enter a baud rate that provides seven bit, even parity service. Click on OK.**

7. **Change values of other template entries if desired.**

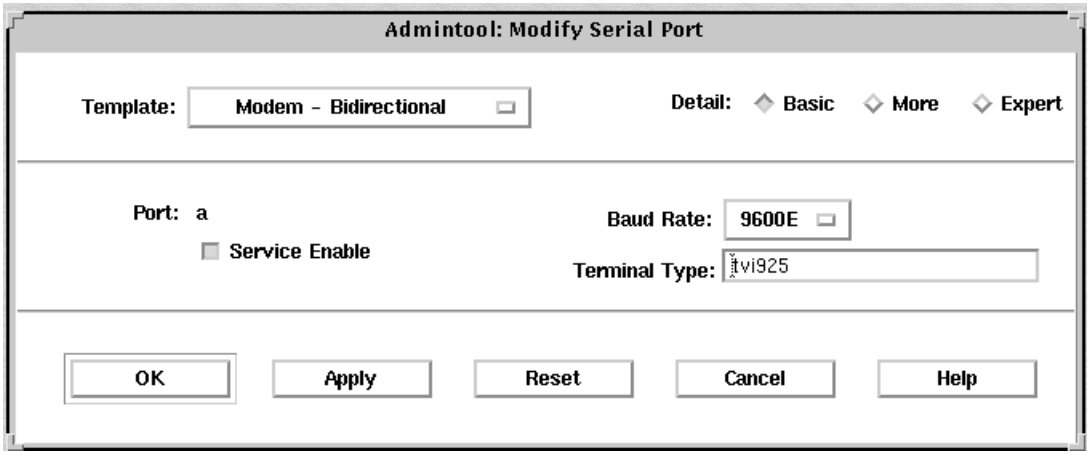
8. **Click on OK to configure the port.**

9. **Use the `pmadm` command to verify the modem service has been configured for use with UUCP.**

```
$ pmadm -l -s ttya
```

Example—Completed Modify Window to Set Up a Modem for Use With UUCP

In this example, the 9600E baud rate was selected. This provides a service with a 9600 baud rate, seven bits, and even parity.



▼ How to Initialize a Port

1. **Start Admintool, if it's not already running.**

   See "How to Start Admintool" on page 245 for more information on starting Admintool.

2. **Select Serial Ports from the Browse menu.**

   The Serial Ports menu is displayed.

3. **Select the port or ports that you want to initialize.**

4. **Choose Modify from the Edit menu.**

   The Modify Serial Port window appears in the Basic Detail mode. To enter additional details, select either the More or Expert Detail modes.

5. **Choose Initialize Only - No Connection from the Use Template menu.**

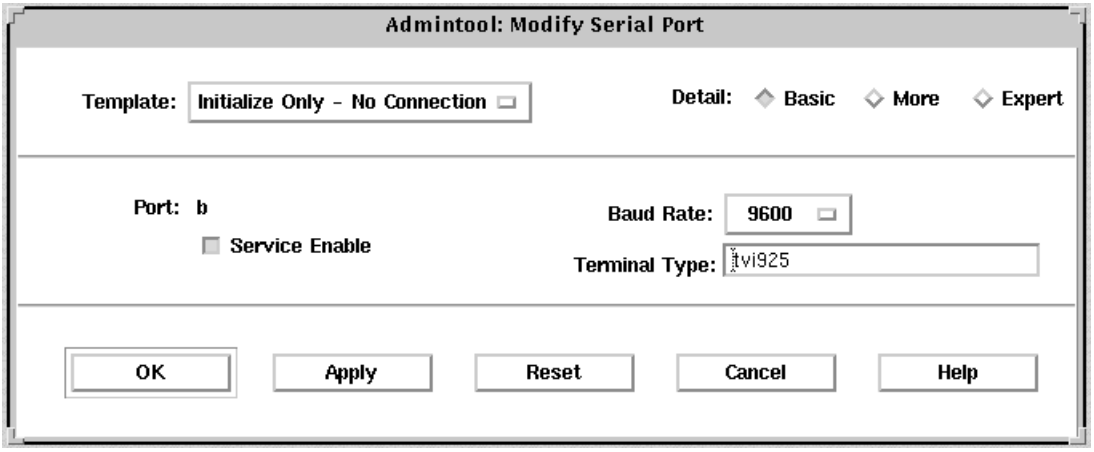
   See Table 13–5 for a description of the Initialize Only - No Connection template.

6. **Click on OK to initialize the port.**

7. **Use the** `pmadm` **command to verify the port has been disabled.**

```
$ pmadm -l -s ttyb
```

Example—Completed Modify Window to Initialize a Port



▼ How to Disable a Port

1. **Start Admintool, if it's not already running.**

   See "How to Start Admintool" on page 245 for more information on starting Admintool.

2. **Select Serial Ports from the Browse menu.**

The Serial Ports menu is displayed.

3. **Select the port or ports that you want to disable.**

4. **Choose Modify from the Edit menu.**

5. **Click on the Service Enable button to disable the port service in the Modify window.**

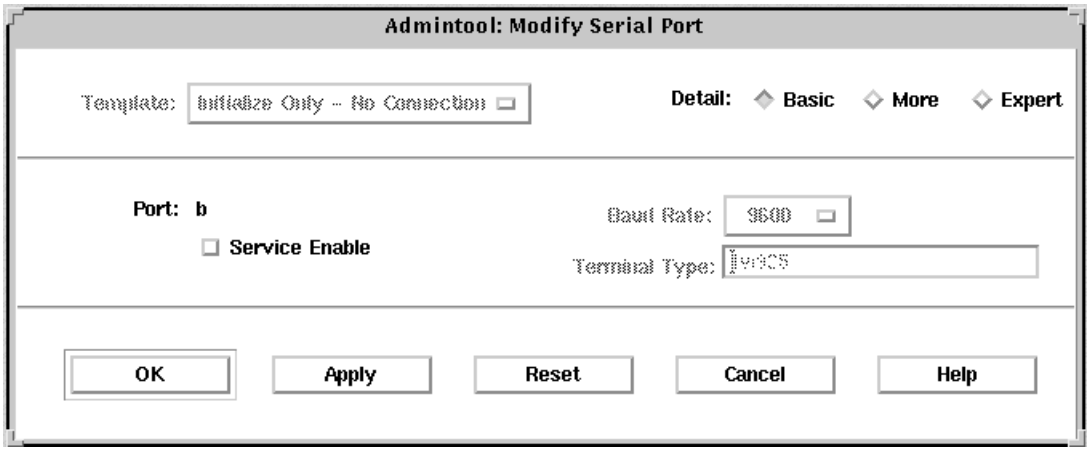
   This button acts as a toggle switch to enable or disable a port service.

6. **Click on OK to disable the port.**

7. **Use the** `pmadm` **command to verify the port service has been disabled.**

```
$ pmadm -l -s ttya
```

Example—Completed Modify Window to Disable a Port



▼ How to Remove a Port Service

1. **Start Admintool, if it's not already running.**

   See "How to Start Admintool" on page 245 for more information on starting Admintool.

2. **Select the port or ports that has a service you want to delete.**

3. **Choose Delete from the Edit menu.**

You are asked if you really want to delete the service for the specified port or ports. You can cancel the delete operation or continue with it.

4. **Use the** `pmadm` **command to verify the port service has been deleted.**

```
$ pmadm -l -s ttya
```

Troubleshooting Terminal and Modem Problems

If users are unable to log in over serial port lines after you have added a terminal or modem and set up the proper services, consider the following possible causes of failure.

- Check with the user.

  Malfunctions in terminals and modem use are typically reported by a user who has failed to log in or dial in. For this reason, it is best to begin troubleshooting by checking for a problem on the desktop.

  Some common reasons for login failure include:

  - Login ID or password is incorrect.
  - Terminal is waiting for X-ON flow control key (Control-q).
  - Serial cable is loose or unplugged.
  - Terminal configuration is incorrect.
  - Terminal is shut off or otherwise has no power.

- Check the terminal.

  Continue to troubleshoot by checking the configuration of the terminal or modem. Determine the proper *ttylabel* for communicating with the terminal or modem. Verify that the terminal or modem settings match those of the *ttylabel*.

- Check the terminal server.

  If the terminal checks out, continue to search for the source of the problem on the terminal or modem server. Use the `pmadm` command to verify that a port monitor has been configured to service the terminal or modem and that it has the correct *ttylabel* associated with it.

```
$ pmadm -l -t ttymon
```

Examine `/etc/ttydefs` and double check the label definition against the terminal configuration. Use `sacadm` to check the port monitor's status. Use `pmadm` to check the service associated with the port the terminal uses.

- Check the serial connection.

  If the Service Access Controller is *starting* the TTY port monitor and `pmadm` reports that the service for the terminal's port is *enabled*, and if the terminal's configuration matches the port monitor's, then continue to search for the problem by checking the serial connection. A serial connection comprises serial ports, cables, and terminals. Test each of these parts by using it with two other parts that are known to be reliable.

  Test all of the following:

  - Serial ports
  - Modems
  - Cables
  - Connectors

- Do not use Admintool to modify serial port settings if the serial port is being used as a console. The correct procedure for changing console settings is by modifying the following line in the `/etc/inittab` file:

```
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "‘uname -n‘ console
login: " -T terminal_type -d /dev/console -l console -m
ldterm,ttcompat
```

- If you are connecting a modem to an IA based system, verify the modem is supported by viewing the *Solaris 8 (Intel Platform Edition) Hardware Compatibility List*.

CHAPTER 14

Setting Up Terminals and Modems With the Service Access Facility (Tasks)

This chapter explains in detail what a system or network administrator needs to know about the Service Access Facility (SAF) in the Solaris environment.

If you want to see examples of specific SAF commands, skip the first section, "Using the Service Access Facility" on page 255, and use the following list to find the instructions you need.

- "Using the Service Access Facility" on page 255
- "Overall Administration: `sacadm` Command" on page 257
- "Port Monitor Service Administrator: `pmadm` Command" on page 258
- "Port Monitors: TTY Monitor and Network Listener" on page 260
- "Administering `ttymon` Port Monitors" on page 263
- "Administering `ttymon` Services" on page 267
- "Reference Material for Service Access Facility Administration" on page 272

For overview information about terminals and modems, see Chapter 12.

Using the Service Access Facility

The SAF is the tool used for administering terminals, modems, and other network devices. The top-level SAF program is the Service Access Controller (SAC). The SAC controls port monitors which you administer through the `sacadm` command. Each port monitor can manage one or more ports.

You administer the services associated with ports through the `pmadm` command. While services provided through SAC can differ from network to network, SAC and the administrative programs `sacadm` and `pmadm` are network independent.

The table below describes the SAF control hierarchy. The `sacadm` command is used to administer the SAC which controls the `ttymon` and `listen` port monitors.

The services of `ttymon` and `listen` are in turn controlled by `pmadm`. One instance of `ttymon` can service multiple ports and one instance of `listen` can provide multiple services on a network interface.

TABLE 14–1 SAF Control Hierarchy

Function	Program	Description
Overall Administration	`sacadm`	Command for adding and removing port monitors
Service Access Controller	`sac`	SAF's master program
Port Monitors	`ttymon`	Monitors serial port login requests
	`listen`	Monitors requests for network services
Port Monitor Service Administrator	`pmadm`	Command for controlling port monitors services
Services	logins; remote procedure calls; other	Services to which SAF provides access
Console Administration	console login	The console is automatically set up via an entry in the `/etc/inittab` file using `ttymon`-express mode. Do not use the `pmadm` or `sacadm` to manage the console directly. See "`ttymon` and the Console Port" on page 261 for more information.

Overall Administration: sacadm Command

The sacadm command is the top level of the SAF. The sacadm command primarily is used to add and remove port monitors such as ttymon and listen. Other sacadm functions include listing the current status of port monitors and administering port monitor configuration scripts.

Service Access Controller: SAC Program

The Service Access Controller program (SAC) oversees all port monitors. A system automatically starts SAC upon entering multiuser mode.

When SAC is invoked, it first looks for, and interprets, each system's configuration script, by which SAC customizes its environment. The modifications made to the SAC environment are inherited by all the "children" of the SAC. This inherited environment might be modified by the children.

After it has interpreted the per-system configuration script, the SAC program reads its administrative file and starts the specified port monitors. For each port monitor, SAC runs a copy of itself (SAC forks a child process). Each child then interprets its per-port monitor configuration script, if such a script exists.

Any modifications to the environment specified in the per-port monitor configuration script affect the port monitor and will be inherited by all its children. Finally, the child process runs the port monitor program using the command found in the SAC administrative file.

SAC Initialization Process

The following steps summarize what happens when SAC is first started:

1. The SAC program is spawned by init at run level two.
2. The SAC program reads /etc/saf/_safconfig, the per-system configuration script.
3. The SAC program reads /etc/saf/_sactab, the SAC administrative file.
4. The SAC program forks a child process for each port monitor it starts.
5. Each port monitor reads /etc/saf/pmtag/_config, the per-port monitor configuration script.

Port Monitor Service Administrator: `pmadm` Command

The `pmadm` command enables you to administer port monitors' services. In particular, you use the `pmadm` command to add or remove a service and to enable or disable a service. You can also install or replace per-service configuration scripts, or print information about a service.

Each instance of a service must be uniquely identified by a port monitor and a port. When you use the `pmadm` command to administer a service, you specify a particular port monitor via the *pmtag* argument, and a particular port via the *svctag* argument.

For each port monitor type, the SAF requires a specialized command to format port monitor-specific configuration data. This data is used by the `pmadm` command. For `ttymon` and `listen` type port monitors, these specialized commands are `ttyadm` and `nlsadmin`, respectively.

A Port Monitor at Work: `ttymon`

Whenever you attempt to log in via a directly connected modem or alphanumeric terminal, `ttymon` goes to work, as follows.

As shown in the figure below, the `init` program is the first process to be started at boot time. Consulting its administrative file (`/etc/inittab`), `init` starts other processes as they are needed. Listed among those processes is the SAC.

SAC, in turn, automatically starts up the port monitors designated in its administrative file (`/etc/saf/_sactab`). The figure below shows only a single `ttymon` port monitor.

After `ttymon` has been started, it monitors the serial port lines for service requests.

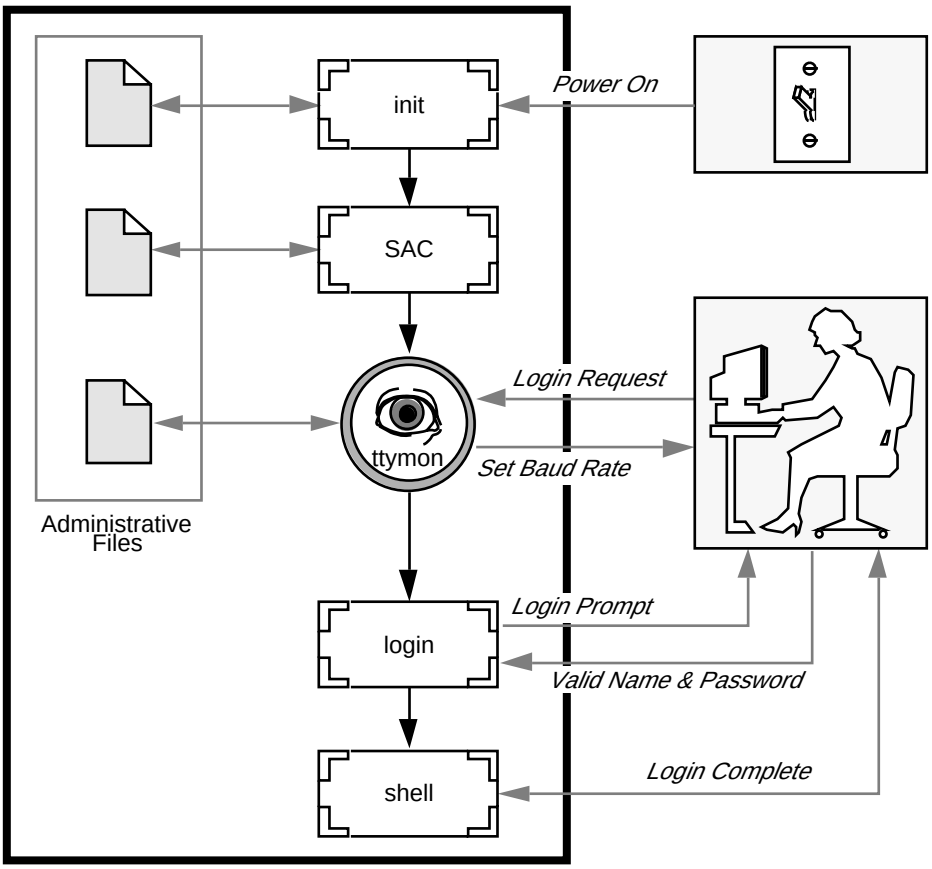


Figure 14–1 How `ttymon` Helps Process a Login Request

When someone attempts to log in via an alphanumeric terminal or a modem, the serial port driver passes the activity to the operating system. The `ttymon` port monitor notes the serial port activity, and attempts to establish a communications link. `ttymon` determines what data transfer rate, line discipline, and handshaking protocol are required to communicate with the device.

Having established the proper parameters for communication with the modem or terminal, `ttymon` passes these parameters to the login program and transfers control to it.

Port Initialization Process

When an instance of `ttymon` is invoked by SAC, `ttymon` starts to monitor its ports. For each port, `ttymon` first initializes the line disciplines, if they are specified, and the speed and terminal settings. The values used for initialization are taken from the appropriate entry in `/etc/ttydefs`.

The ttymon port monitor then writes the prompt and waits for user input. If the user indicates that the speed is inappropriate by pressing the Break key, ttymon tries the next speed and writes the prompt again.

If *autobaud* is enabled for a port, ttymon will try to determine the baud rate on the port automatically. Users must press Return before ttymon can recognize the baud rate and print the prompt.

When valid input is received, ttymon interprets the per-service configuration file for the port, creates a /etc/utmpx entry if required, establishes the service environment, and invokes the service associated with the port.

After the service terminates, ttymon cleans up the /etc/utmpx entry, if one exists, and returns the port to its initial state.

Bidirectional Service

If a port is configured for bidirectional service, ttymon will:

- Allow users to connect to a service
- Allow uucico, cu, or ct to use the port for dialing out (if the port's free)
- Wait to read a character before printing a prompt
- Invoke the port's associated service—without sending the prompt message—when a connection is requested (if the connect-on-carrier flag is set)

Port Monitors: TTY Monitor and Network Listener

Though SAF provides a generic means for administering any future or third-party port monitors, only two are implemented in the Solaris environment—ttymon and listen.

TTY Port Monitor: ttymon

The ttymon port monitor is STREAMS-based. It monitors ports; sets terminal modes, baud rates, and line disciplines; and invokes the login process. (It provides Solaris users the same services that getty did under previous versions of SunOS 4.1 software.)

The ttymon port monitor runs under the SAC program. It is configured using the sacadm command. Each instance of ttymon can monitor multiple ports. These ports

are specified in the port monitor's administrative file. The administrative file is configured using the `pmadm` and `ttyadm` commands.

`ttymon` and the Console Port

Console services are not managed by the Service Access Controller nor any explicit `ttymon` administration file. An entry in the `/etc/inittab` file is used to manage the console port using `ttymon` in *express* mode, which is a special `ttymon` mode that is invoked directly by a command that requires login service.

The default console entry in the `/etc/inittab` file looks like this:

```
co:234:respawn:/usr/lib/saf/ttymon -g -h -p "‘uname -n‘ console login: "
 -T terminal_type -d /dev/console -l console -m ldterm,ttcompat
```

`co:234:respawn:`	`co` identifies the entry as the console; `234` identifies the run levels for the action, `respawn`, which means the console entry should be restarted if it fails or doesn't exist at run levels 2, 3, and 4.
`/usr/lib/saf/ttymon -g -h`	The `–g` option is used so the correct baud rate and terminal setting can be set on a port and connect to a login service without being preconfigured by the SAC. The `–h` option forces a line hangup by setting the line speed to zero before setting the default or specified speed.
`–p "‘uname -n‘ console login:`	Identifies the prompt string for the console port.
`–t` *terminal_type*	Identifies the terminal type of the console.
`–d /dev/console -l console -m ldterm,ttcompat`	The `–d` option identifies the console device; the `–l` option identifies the `ttylabel` in the `/etc/ttydefs` file; and the `–m` option identifies the STREAMS modules to be pushed.

Special `ttymon`-Specific Administrative Command: `ttyadm`

The `ttymon` administrative file is updated by `sacadm` and `pmadm`, as well as by the `ttyadm` command. The `ttyadm` command formats `ttymon`-specific information and writes it to the standard output, providing a means for presenting formatted `ttymon`-specific data to the `sacadm` and `pmadm` commands.

Thus, `ttyadm` does not administer `ttymon` directly; rather, it complements the generic administrative commands, `sacadm` and `pmadm`. See `ttyadm`(1M) for more details.

Network Listener Service: `listen`

The `listen` port monitor runs under SAC. It monitors the network for service requests, accepts requests when they arrive, and invokes servers in response to those service requests.

The `listen` port monitor is configured using the `sacadm` command. Each instance of `listen` can provide multiple services. These services are specified in the port monitor's administrative file. This administrative file is configured using the `pmadm` and `nlsadmin` commands.

The network listener process can be used with any connection-oriented transport provider that conforms to the Transport Layer Interface (TLI) specification. In the Solaris environment, `listen` port monitors can provide additional network services not provided by `inetd`.

Special `listen`-Specific Administrative Command: `nlsadmin`

The `listen` port monitor's administrative file is updated by `sacadm` and `pmadm`, as well as by the `nlsadmin` command. The `nlsadmin` command formats `listen`-specific information and writes it to the standard output, providing a means of presenting formatted `listen`-specific data to the `sacadm` and `pmadm` commands.

Thus, `nlsadmin` does not administer `listen` directly; rather, it complements the generic administrative commands, `sacadm` and `pmadm`.

Each network can have at least one instance of the network listener process associated with it. Each network is configured separately. The `nlsadmin` command controls the operational states of `listen` port monitors.

The `nlsadmin` command can establish a `listen` port monitor for a given network, configure the specific attributes of that port monitor, and *start* and *kill* the monitor.

The `nlsadmin` command can also report on the `listen` port monitors on a machine.

See `nlsadmin`(1M) for more details.

Administering `ttymon` Port Monitors

Use the `sacadm` command to add, list, remove, kill, start, enable, disable, enable, and remove a `ttymon` port monitor.

Note - You must be superuser to perform the following procedures.

▼ How to Add a `ttymon` Port Monitor

To add a `ttymon` port monitor, type:

```
# sacadm -a -p mbmon -t ttymon -c /usr/lib/saf/ttymon -v ‘ttyadm
-V‘ -y "TTY Ports a & b"
```

–a	The *add* port monitor flag
–p	Specifies the *pmtag* `mbmon` as the port monitor tag
–t	Specifies the port monitor *type* as `ttymon`
–c	Defines the *command* string used to start the port monitor
–v	Specifies the *version* number of the port monitor
–y	Defines a comment to describe this instance of the port monitor

▼ How to View `ttymon` Port Monitor Status

To see the status of a `ttymon` port monitor, type:

```
# sacadm -l -p mbmon
```

`–l` The *list* port monitor status flag

`–p` Specifies the *pmtag* `mbmon` as the port monitor tag

Example—Viewing `ttymon` Port Monitor Status

```
# sacadm -l -p mbmon
PMTAG  PMTYPE  FLGS RCNT STATUS    COMMAND
mbmon  ttymon  -    0    STARTING  /usr/lib/saf/ttymon #TTY Ports a & b
```

`PMTAG` `mbmon`	Identifies the port monitor name, `mbmon`.
`PMTYPE` `ttymon`	Identifies the port monitor type, `ttymon`.
`FLGS` `-`	Indicates whether the following two flags are set: `d`, do not enable the new port monitor, or `x`, do not start the new port monitor. There are no flags set in this example.
`RCNT` `0`	Indicates the return count value. A return count of `0` indicates that the port monitor is not to be restarted if it fails.
`STATUS` `STARTING`	Indicates the current status of the port monitor.

`COMMAND`	Identifies the command used to start the port monitor.
`/usr/lib/saf ...`	
`#TTY Ports a & b`	Identifies any comment used to describe the port monitor.

▼ How to Stop a `ttymon` Port Monitor

To kill a `ttymon` port monitor, type:

```
# sacadm -k -p mbmon
```

`–k`	The *kill* port monitor status flag
`–p`	Specifies the *pmtag* `mbmon` as the port monitor tag

▼ How to Start a `ttymon` Port Monitor

To start a killed `ttymon` port monitor, type:

```
# sacadm -s -p mbmon
```

`–s`	The *start* port monitor status flag
`–p`	Specifies the *pmtag* `mbmon` as the port monitor tag

▼ How to Disable a `ttymon` Port Monitor

Disabling a port monitor prevents new services from starting, without affecting existing services.

To disable a `ttymon` port monitor, type:

```
# sacadm -d -p mbmon
```

–d The *disable* port monitor status flag

–p Specifies the *pmtag* `mbmon` as the port monitor tag

▼ How to Enable a `ttymon` Port Monitor

Enabling a `ttymon` port monitor allows it to service new requests.

To enable a `ttymon` port monitor, type:

```
# sacadm -e -p mbmon
```

–e The *enable* port monitor status flag

–p Specifies the *pmtag* `mbmon` as the port monitor tag

▼ How to Remove a `ttymon` Port Monitor

To remove a `ttymon` port monitor, type:

```
# sacadm -r -p mbmon
```

–r The *remove* port monitor status flag

–p Specifies the *pmtag* `mbmon` as the port monitor tag

Note - Removing a port monitor deletes all the configuration files associated with it. Port monitor configuration files cannot be updated or changed using `sacadm`. To reconfigure a port monitor, *remove* it and *add* a new one.

Administering ttymon Services

Use pmadm to add services, list the services of one or more ports associated with a port monitor, and enable or disable a service.

Note - You must be superuser to perform the following procedures.

▼ How to Add a Service

To add a standard terminal service to the mbmon port monitor, type:

```
# pmadm -a -p mbmon -s a -i root -v 'ttyadm -V' -m "'ttyadm -i 'Terminal disabled'
 -l contty -m ldterm,ttcompat -S y -d /dev/term/a -s /usr/bin/login'"
```

Note - In this example, the input wraps to the next line. Do not put a Return or line feed after contty.

–a	The *add* port monitor status flag
–p	Specifies the *pmtag* mbmon as the port monitor tag
–s	Specifies the *svctag* a as the port monitor *service* tag
–i	Specifies the *identity* to be assigned to *svctag* when it runs
–v	Specifies the *version* number of the port monitor
–m	Specifies the ttymon-specific configuration data formatted by ttyadm

The above pmadm command contains an embedded ttyadm command. The options in this embedded command are as follows:

`-b`	The *bidirectional* port flag
`-i`	Specifies the *inactive* (disabled) response message
`-l`	Specifies which TTY *label* in `/etc/ttydefs` to use
`-m`	Specifies the STREAMS *modules* to push before invoking this service
`-d`	Specifies the full path name to the *device* to use for the TTY port
`-s`	Specifies the full path name of the *service* to invoke when a connection request is received; if arguments are required, enclose the command and its arguments in quotation marks (")

▼ How to View the Status of a TTY Port Service

Use the `pmadm` command as shown to list the status of a TTY port, or all the ports associated with a port monitor.

Listing One Service

To list one service of a port monitor, type:

```
# pmadm -l -p mbmon -s a
```

`-l`	Lists service information
`-p`	Specifies the *pmtag* `mbmon` as the port monitor tag
`-s`	Specifies the *svctag* a as the port monitor *service* tag

Listing All Services of All Port Monitors

To list all services of all port monitors, type:

```
# pmadm -l
```

`–l` Lists service information

Listing All Services of a Port Monitor

To list all services of a port monitor, type:

```
# pmadm -l -p mbmon
```

`–l` Lists service information

`–p` Specifies the *pmtag* `mbmon` as the port monitor tag

Example—Viewing the Status of a TTY Port Monitor Service

```
# pmadm -l -p mbmon
PMTAG  PMTYPE  SVCTAG FLAGS ID    <PMSPECIFIC>
mbmon  ttymon  a      -     root  /dev/term/a - - /usr/bin/login - contty
ldterm,ttcompat login:  Terminal disabled - y  #
```

`mbmon` Identifies the port monitor name, `mbmon`, set by using the `pmadm -p` command.

`ttymon` Identifies the port monitor type, `ttymon`.

`a` Indicates the service tag value set by using the `pmadm -s` command.

`-` Identifies whether the following flags are set by using the `pmadm -f` command:

`x`, which means do not enable the service;

`u`, which means create a `utmpx` entry for the service. No flags are set in this example.

`root` Identifies the ID assigned to the service when its started. This value is set by using the `pmadm -i` command.

`<PMSPECIFIC>` *Information*

`/dev/term/a`	Indicates the TTY port pathname set by using the `ttyadm -d` command.
-	Indicates whether the following flags are set by using the `ttadm -c -b -h -I -r` command: `c`, sets the connect on carrier flag for the port `b`, sets the port as bidirectional, allowing both incoming and outgoing traffic `h`, suppresses an automatic hangup immediately after an incoming call is received `I`, initializes the port `r`, forces `ttymon` to wait until it receives a character from the port before it prints the `login:` message.
-	Indicates a value set by using the `ttyadm -r` option. This option determines when `ttymon` displays a prompt after receiving data from a port. If count is 0, `ttymon` will wait until it receives any character. If count is greater than 0, `ttymon` will wait until count new lines have been received. No value is set in this example.
`/usr/bin/login`	Identifies the full pathname of the service to be invoked when a connected is received. This value is set by using `ttyadm -s` command.
-	Identifies the `ttyadm -t` command's (timeout) value. This option specifies that `ttymon` should close a port if the open on the port succeeds, and no input data is received in timeout seconds. There is no timeout value in this example.
`contty`	Identifies the TTY label in the `/etc/ttydefs` file. This value is set by using the `ttyadm -l` command.
`ldterm,ttcompat`	Identifies the STREAMS modules to be pushed. These modules are set by using the `ttyadmin -m` command.
`login: Terminal disabled`	Identifies an inactive message to be displayed when the port is disabled. This message is set by using the `ttyadm -i` command.
`tvi925`	Identifies the terminal type, if set, by using the `ttyadm -T`command. The terminal type is `tvi925` in this example .

y	Identifies the software carrier value set by using the `ttyadm -S` command; `n` will turn software carrier off, `y` will turn software carrier on. Software carrier is turned on in this example.
#	Identifies any comment specified with the `pmadm -y` command. (There is no comment in this example).

▼ How to Enable a Port Monitor Service

To enable a disabled port monitor service, type:

```
# pmadm -e -p mbmon -s a
```

–e	The *enable* flag
–p	Specifies the *pmtag* `mbmon` as the port monitor tag
–s	Specifies the *svctag* `a` as the port monitor *service* tag

▼ How to Disable a Port Monitor Service

To disable a port monitor service, type:

```
# pmadm -d -p mbmon -s a
```

–d	The *disable* flag
–p	Specifies the *pmtag* `mbmon` as the port monitor tag
–s	Specifies the *svctag* `a` as the port monitor *service* tag

Reference Material for Service Access Facility Administration

Files Associated With SAF

SAF uses configuration files which can be modified by using the `sacadm` and `pmadm` commands. You should not need to edit them manually.

File Name	Description
`/etc/saf/_sysconfig`	Per-system configuration script
`/etc/saf/_sactab`	SAC's administrative file; contains configuration data for the port monitors that the SAC controls
`/etc/saf/`*`pmtag`*	Home directory for port monitor *pmtag*
`/etc/saf/`*`pmtag`*`/_config`	Per-port monitor configuration script for port monitor pmtag if it exists
`/etc/saf/`*`pmtag`*`/_pmtab`	Port monitor *pmtag*'s administrative file; contains port monitor-specific configuration data for the services *pmtag* provides
`/etc/saf/`*`pmtag`*`/`*`svctag`*	Per-service configuration script for service *svctag*
`/var/saf/log`	SAC's log file
`/var/saf/`*`pmtag`*	Directory for files created by *pmtag*, for example, log files

The `/etc/saf/_sactab` File

The `/etc/saf/_sactab` looks like this:

```
# VERSION=1
zsmon:ttymon::0:/usr/lib/saf/ttymon     #
```

`# VERSION=1`	Indicates the Service Access Facility version number.
`zsmon`	Is the name of the port monitor.
`ttymon`	Is the type of port monitor.
`::`	Indicates whether the following two flags are set: `d`, do not enable the port monitor `x`, do not start the port monitor. No flags are set in this example.
`0`	Indicates the return code value. A return count of `0` indicates that the port monitor is not be restarted if it fails.
`/usr/lib/saf/ttymon`	Indicates the port monitor pathname

The `/etc/saf/`*pmtab/_pmtab* File

The `/etc/saf/`*pmtab/_pmtab* file, such as `/etc/saf/zsmon/_pmtab`, looks like this:

```
# VERSION=1
ttya:u:root:reserved:reserved:reserved:/dev/term/a:I::/usr/bin/login::9600:ldterm,
ttcompat:ttya login\: ::tvi925:y:#
```

`# VERSION=1`	Indicates the Service Access Facility version number.
`ttya`	Indicates the service tag.
`x,u`	Identifies whether the following flags are set: `x`, which means do not enable the service `u`, which means create a `utmpx` entry for the service
`root`	Indicates the identity assigned to the service tag.
`reserved`	This field is reserved.
`reserved`	This field is reserved.

`reserved`	This field is reserved.
`/dev/term/a`	Indicates the TTY port pathname.
`/usr/bin/login`	Identifies the full pathname of the service to be invoked when a connection is received.
`:c,b,h,I,r:`	Indicates whether the following flags are set `c`, sets the connect on carrier flag for the port `b`, sets the port as bidirectional, allowing both incoming and outgoing traffic `h`, suppresses an automatic hangup immediately after an incoming call is received `I`, initializes the port `r`, forces `ttymon` to wait until it receives a character from the port before it prints the `login:` message.
`9600`	Identifies the TTY label defined in `/etc/ttydefs` file
`ldterm,ttcompat`	Identifies the STREAMS modules to be pushed
`ttya login\:`	Identifies the prompt to be displayed
`:y/n:`	
message	Identifies any inactive (disabled) response message
`tvi925`	Identifies the terminal type.
`y`	Indicates whether software carrier is set (`y`/`n`).

Service States

The `sacadm` command controls the states of services. The possible states are shown below.

State	Notes
Enabled	*Default state* – When the port monitor is added, the service operates.
Disabled	*Default state* – When the port monitor is removed, the service stops.

To determine the state of any particular service, use the following:

```
# pmadm -l -p portmon_name -s svctag
```

Port Monitor States

The `sacadm` command controls the states of `ttymon` and `listen` port monitors. The possible states are shown below.

State	Notes
`Started`	*Default state* – When the port monitor is added, it is automatically started.
`Enabled`	*Default state* – When the port monitor is added, it is automatically ready to accept requests for service.
`Stopped`	*Default state* – When the port monitor is removed, it is automatically stopped.
`Disabled`	*Default state* – When the port monitor is removed, it automatically continues existing services and refuses to add new services.
`Starting`	*Intermediate state* – The port monitor is in the process of starting.
`Stopping`	*Intermediate state* – The port monitor has been manually terminated, but it has not completed its shutdown procedure. It is on the way to becoming stopped.
`Notrunning`	*Inactive state* – The port monitor has been killed. All ports previously monitored are inaccessible. An external user cannot tell whether a port is `disabled` or `notrunning`.
`Failed`	*Inactive state* – The port monitor is unable to start and remain running.

To determine the state of any particular port monitor, use the following:

```
# sacadm -l -p portmon_name
```

Port States

Ports can be enabled or disabled depending on the state of the port monitor that controls them.

State	Notes
Serial (`ttymon`) Port States	
`Enabled`	The `ttymon` port monitor sends a prompt message to the port and provides login service to it.
`Disabled`	Default state of all ports if `ttymon` is killed or disabled. If you specify this state, `ttymon` will send out the `disabled` message when it receives a connection request.

CHAPTER 15

Managing System Security Topics

This section provides instructions for managing system security in the Solaris environment. This section contains these chapters.

Chapter 16	Provides overview information about file, system, and network security.
Chapter 17	Provides step-by-step instructions to display file information, change file ownership and permissions, and set special permissions.
Chapter 18	Provides step-by-step instructions to check login status, set up dial-up passwords, restrict root access, and monitor root access and `su` attempts.
Chapter 19	Provides overview information and instructions for using role-based access control.
Chapter 20	Provides step-by-step instructions for setting up Kerberos login authentication and Pluggable Authentication Module (PAM).
Chapter 21	Provides overview information about the Sun Enterprise Authentication Mechanism (SEAM) security product.
Chapter 22	Provides step-by-step instructions for configuring SEAM in your network.

Chapter 23	Provides reference information on the SEAM security product.
Chapter 24	Provides overview information about Automated Security Enhancement Tool (ASET) and step-by-step instructions to run ASET interactively or periodically (by using a `cron` job). It also includes information about collecting client ASET reports on a server.

CHAPTER 16

Managing System Security (Overview)

Keeping a system's information secure is an important system administration responsibility. This chapter provides overview information about managing system security at the file, system, and network level.

This is a list of the overview information in this chapter.

- "What's New in Solaris System Security?" on page 279
- "Where to Find System Security Tasks" on page 281
- "Controlling Access to a Computer System" on page 281
- "File Security" on page 284
- "System Security" on page 286
- "Network Security" on page 289

What's New in Solaris System Security?

This section describes new security features.

New Default Ownerships and Permissions on System Files and Directories

Many system files and directories in this Solaris release have different default ownership and stricter permissions than in previous releases. The default ownership and permissions changes are:

- Default file and directory ownership has been changed from bin to root.

- Files and directories previously having default permissions of 775 now have default permissions of 755.
- Files and directories previously having default permissions of 664 now have default permissions of 644.
- Default umask of the system is 022.

Keep the following in mind when creating a package to be added to a system running the Solaris 8 release:

- All files and directories must have root as the default owner.
- Directories and executables must have default permissions of 555 or 755.
- Ordinary files must have default permissions of 644 or 444.
- Files with `setuid` and/or `setgid` ownership cannot be writable by the owner, unless the owner is root

These changes do not apply to all files and directories in this release; for example, the changes do not apply to OpenWindows or CDE files and directories.

Role-Based Access Control

Role-based access control (RBAC) provides a flexible way to package superuser privileges for assignment to user accounts so that you don't have to give all superuser privileges to a user that needs to solve a specific problem.

See Chapter 19 for more information.

Sun Enterprise Authentication Mechanism (SEAM) or Kerberos V5 Client Support

This feature provides the Kerberos V5 client-side infrastructure, an addition to the Pluggable Authentication Module (PAM), and utility programs that can be used to secure RPC based applications, such as the NFS service. Kerberos provides selectable strong user or server level authentication, integrity, or privacy support. The Kerberos clients can be used in conjunction with Sun Enterprise Authentication Mechanism (SEAM), a part of SEAS 3.0, or other Kerberos V5 software (for instance, the MIT distribution) to create a complete single network sign-on solution.

See Chapter 21 for more information.

Where to Find System Security Tasks

Use these references to find step-by-step instructions for setting up system security.

- Chapter 17
- Chapter 18
- Chapter 19
- Chapter 20
- Chapter 24

Controlling Access to a Computer System

At the file level, the SunOS operating system provides some standard security features that you can use to protect files, directories, and devices. At the system and network levels, the security issues are mostly the same. In the workplace, a number of systems connected to a server can be thought of as one large multifaceted system. The system administrator is responsible for the security of this larger system or network. Not only is it important to defend the network from outsiders trying to gain access to the network, but it is also important to ensure the integrity of the data on the systems within the network.

The first line of security defense is to control access to your system. You can control and monitor system access by:

- Maintaining physical site security
- Maintaining login control
- Restricting access to data in files
- Maintaining network control
- Monitoring system usage
- Setting the path variable correctly
- Securing files
- Installing a firewall
- Reporting security problems

Maintaining Physical Site Security

To control access to your system, you must maintain the physical security of your computer environment. For instance, if a system is logged in and left unattended, anyone who can use that system can gain access to the operating system and the network. You need to be aware of your computer's surroundings and physically protect it from unauthorized access.

Maintaining Login and Access Control

You also must restrict unauthorized logins to a system or the network, which you can do through password and login control. All accounts on a system should have a password. An account without a password makes your entire network accessible to anyone who can guess a user name.

Solaris software restricts control of certain system devices to the user login account. Only a process running as superuser or console user can access a system mouse, keyboard, frame buffer, or audio device unless `/etc/logindevperm` is edited. See `logindevperm`(4) for more information.

Restricting Access to Data in Files

After you have established login restrictions, you can control access to the data on your system. You might want to allow some users to read some files, and give other users permission to change or delete some files. You might have some data that you do not want anyone else to see. Chapter 17 discusses how to set file permissions.

Maintaining Network Control

Computers are often part of a configuration of systems called a *network*. A network allows connected systems to exchange information and access data and other resources available from systems connected to the network. Networking has created a powerful and sophisticated way of computing. However, networking has also jeopardized computer security.

For instance, within a network of computers, individual systems are open to allow sharing of information. Also, because many people have access to the network, there is more chance for allowing unwanted access, especially through user error (for example, through a poor use of passwords).

Monitoring System Usage

As system administrator, you need to monitor system activity, being aware of all aspects of your systems, including the following:

- What is the normal load?
- Who has access to the system?
- When do individuals access the system?

With this kind of knowledge, you can use the available tools to audit system use and monitor the activities of individual users. Monitoring is very useful when there is a suspected breach in security.

Setting the Correct Path

It is important to set your path variable correctly; otherwise, you can accidentally run a program introduced by someone else that harms your data or your system. This kind of program, which creates a security hazard, is referred to as a "Trojan horse." For example, a substitute `su` program could be placed in a public directory where you, as system administrator, might run it. Such a script would look just like the regular `su` command; since it removes itself after execution, it is hard to tell that you have actually run a Trojan horse.

The path variable is automatically set at login time through the startup files: `.login`, `.profile`, and `.cshrc`. Setting up the user search path so that the current directory (.) comes last prevents you or your users from running this type of Trojan horse. The path variable for superuser should not include the current directory at all. The ASET utility examines the startup files to ensure that the path variable is set up correctly and that it does not contain a dot (.) entry.

Securing Files

Since the SunOS operating system is a multiuser system, file system security is the most basic, and important, security risk on a system. You can use both the traditional UNIX file protection or the more secure access control lists (ACLs) to protect your files.

Also, many executable programs have to be run as root (that is, as superuser) to work properly. These executables run with the user ID set to 0 (`setuid=0`). Anyone running these programs runs them with the root ID, which creates a potential security problem if the programs are not written with security in mind.

Except for the executables shipped with `setuid` to root, you should disallow the use of `setuid` programs, or at least restrict and keep them to a minimum.

Installing a Firewall

Another way to protect your network is to use a firewall or secure gateway system. A firewall is a dedicated system separating two networks, each of which approaches the other as untrusted. You should consider this setup as mandatory between your internal network and any external networks, such as the Internet, with which you want internal network users to communicate.

A firewall can also be useful between some internal networks. For example, the firewall or secure gateway computer will not send a packet between two networks unless the gateway computer is the origin or the destination address of the packet. A firewall should also be set up to forward packets for particular protocols only. For example, you can allow packets for transferring mail, but not those for `telnet` or `rlogin`. The ASET utility, when run at high security, disables the forwarding of Internet Protocol (IP) packets.

Reporting Security Problems

If you experience a suspected security breach, you can contact the Computer Emergency Response Team/Coordination Center (CERT/CC), which is a Defense Advanced Research Projects Agency (DARPA) funded project located at the Software Engineering Institute at Carnegie Mellon University. It can assist you with any security problems you are having. It can also direct you to other Computer Emergency Response Teams that might be more appropriate to your particular needs. You can call CERT/CC at its 24-hour hotline: (412) 268-7090, or contact the team by email at `cert@cert.sei.cmu.edu`.

File Security

The SunOS operating system is a multiuser system, which means that all the users logged in to a system can read and use files belonging to one another, as long as they have permission to do so. The table below describes file system administration commands. See Chapter 17 for step-by-step instructions on securing files.

File Administration Commands

This table describes the file administration commands for monitoring and securing files and directories.

TABLE 16–1 File Administration Commands

Command	Description
`ls`(1)	Lists the files in a directory and information about them.
`chown`(1)	Changes the ownership of a file.
`chgrp`(1)	Changes the group ownership of a file.
`chmod`(1)	Changes permissions on a file. You can use either symbolic mode (letters and symbols) or absolute mode (octal numbers) to change permissions on a file.

File Encryption

Placing a sensitive file into an inaccessible directory (`700` mode) and making the file unreadable by others (`600` mode) will keep it secure in most cases. However, someone who guesses your password or the root password can read and write to that file. Also, the sensitive file is preserved on backup tapes every time you back up the system files to tape.

Fortunately, an additional layer of security is available to all SunOS system software users in the United States—the optional file encryption kit. The encryption kit includes the `crypt`(1) command which scrambles the data to disguise the text.

Access Control Lists (ACLs)

ACLs (ACLs, pronounced “ackkls”) can provide greater control over file permissions when the traditional UNIX file protection in the SunOS operating system is not enough. The traditional UNIX file protection provides read, write, and execute permissions for the three user classes: owner, group, and other. An ACL provides better file security by enabling you to define file permissions for the owner, owner’s group, others, specific users and groups, and default permissions for each of those categories. See “Using Access Control Lists (ACLs)” on page 312 for step–by–step instructions on using ACLs.

The table below lists the commands for administering ACLs on files or directories.

TABLE 16–2 ACL Commands

Command	Description
`setfacl`(1)	Sets, adds, modifies, and deletes ACL entries
`getfacl`(1)	Displays ACL entries

System Security

This section describes how to safeguard your system against unauthorized access, such as how to prevent an intruder from logging in to your system, how to maintain the password files, and how to prevent unauthorized superuser access to sensitive system files and programs.

You can set up two security barriers on a system. The first security barrier is the login program. To cross this barrier and gain access to a system, a user must supply a user name and a corresponding password known by the local system or by the name service (NIS or NIS+).

The second security barrier is ensuring that the system files and programs can be changed or removed by superuser only. A would-be superuser must supply the root user name and its correct password.

Login Access Restrictions

When a user logs in to a system, the login program consults the appropriate database according to the information listed in the `/etc/nsswitch.conf` file. The entries in this file can include `files` (designating the `/etc` files), `nis` (designating the NIS database), and `nisplus` (designating the NIS+ database). See the *Solaris Naming Administration Guide* or `nsswitch.conf`(4) for a description of this file.

The login program verifies the user name and password entered. If the user name is not in the password file or the password is not correct for the user name, the login program denies access to the system. When the user supplies a name from the password file and the correct password for the name, the system grants the user access to the system.

Special Logins

There are two common ways to access a system—by using a conventional user login, or by using the root login. In addition, a number of special *system* logins allow a user to perform administrative commands without using the root account. The administrator assigns passwords to these login accounts.

The table below lists the system login accounts and their uses. The system logins perform special functions, and each has its own group identifier number (GID). Each of these logins should have its own password, which should be distributed on a need-to-know basis.

TABLE 16–3 System Logins

Login Account	GID	Use
`root`	`0`	Has almost no restrictions and overrides all other logins, protections, and permissions. The root account has access to the entire system. The password for the root login should be very carefully protected. Owns most of the Solaris commands.
`daemon`	`1`	Controls background processing.
`bin`	`2`	Owns some of the Solaris commands.
`sys`	`3`	Owns many system files.
`adm`	`4`	Owns certain administrative files.
`lp`	`71`	Owns the object and spooled data files for the printer.
`uucp`	`5`	Owns the object and spooled data files for UUCP, the UNIX-to-UNIX copy program.
`nuucp`	`9`	Is used by remote systems to log in to the system and start file transfers.

You should also set the security of the `eeprom` command to require a password. See `eeprom`(1M) for more information.

Managing Password Information

When logging in to a system, users must enter both a user name and a password. Although logins are publicly known, passwords must be kept secret, known only to users. You should ask your users to choose their passwords carefully, and they should change them often.

Passwords are initially created when you set up a user account. To maintain security on user accounts, you can set up password aging to force users to routinely change their passwords, and you can also disable a user account by locking the password. See "Managing User Accounts and Groups (Overview)" in *System Administration Guide, Volume 1* and `passwd`(1) for detailed information about setting up and maintaining passwords.

NIS+ Password File

If your network uses NIS+, the password information is kept in the NIS+ database. Information in the NIS+ database can be protected by restricting access to authorized users. You can use AdminSuite™ 2.3's User Manager or the `passwd` command to change a user's NIS+ password.

NIS Password File

If your network uses NIS, the password information is kept in the NIS password map. NIS does not support password aging. You can use AdminSuite 2.3's User Manager or the `passwd` command to change a user's NIS password.

`/etc` Files

If your network uses `/etc` files, the password information is kept in the system's `/etc/passwd` and `/etc/shadow` files. The user name and other information are kept in the password file `/etc/passwd`, while the encrypted password itself is kept in a separate *shadow* file, `/etc/shadow`. This is a security measure that prevents a user from gaining access to the encrypted passwords. While the `/etc/passwd` file is available to anyone who can log in to a machine, only superuser can read the `/etc/shadow` file. You can use AdminSuite 2.3's User Manager, Admintool, or the `passwd` command to change a user's password on a local system.

Using the Restricted Shell

The standard shell allows a user to open files, execute commands, and so on. The restricted shell can be used to limit the ability of a user to change directories and execute commands. The restricted shell (`rsh`) is located in the `/usr/lib` directory.

(Note that this is not the remote shell, which is `/usr/sbin/rsh`.) The restricted shell differs from the normal shell in these ways:

- The user is limited to the home directory (can't use `cd` to change directories).
- The user can use only commands in the `PATH` set by the system administrator (can't change the `PATH` variable).
- The user can access only files in the home directory and its subdirectories (can't name commands or files using a complete path name).
- The user cannot redirect output with `>` or `>>`.

The restricted shell allows the system administrator to limit a user's ability to stray into the system files, and is intended mainly to set up a user who needs to perform specific tasks. The `rsh` is not completely secure, however, and is only intended to keep unskilled users from getting into (or causing) trouble.

See `rsh`(1M) for information about the restricted shell.

Tracking Superuser (Root) Login

Your system requires a root password for superuser mode. In the default configuration, a user cannot remotely log in to a system as root. When logging in remotely, a user must log in as himself and then use the `su` command to become root. This enables you to track who is using superuser privileges on your system.

Monitoring Who is Becoming Superuser or Other Users

You have to use the `su` command to change to another user, for example, if you want to become superuser. For security reasons, you can monitor who has been using the `su` command, especially those users who are trying to gain superuser access.

See "How to Monitor Who Is Using the `su` Command" on page 333 for detailed instructions.

Network Security

The more available access is across a network, the more advantageous it is for networked systems. However, free access and sharing of data and resources create security problems. Network security is usually based on limiting or blocking operations from remote systems. The figure below describes the security restrictions you can impose on remote operations.

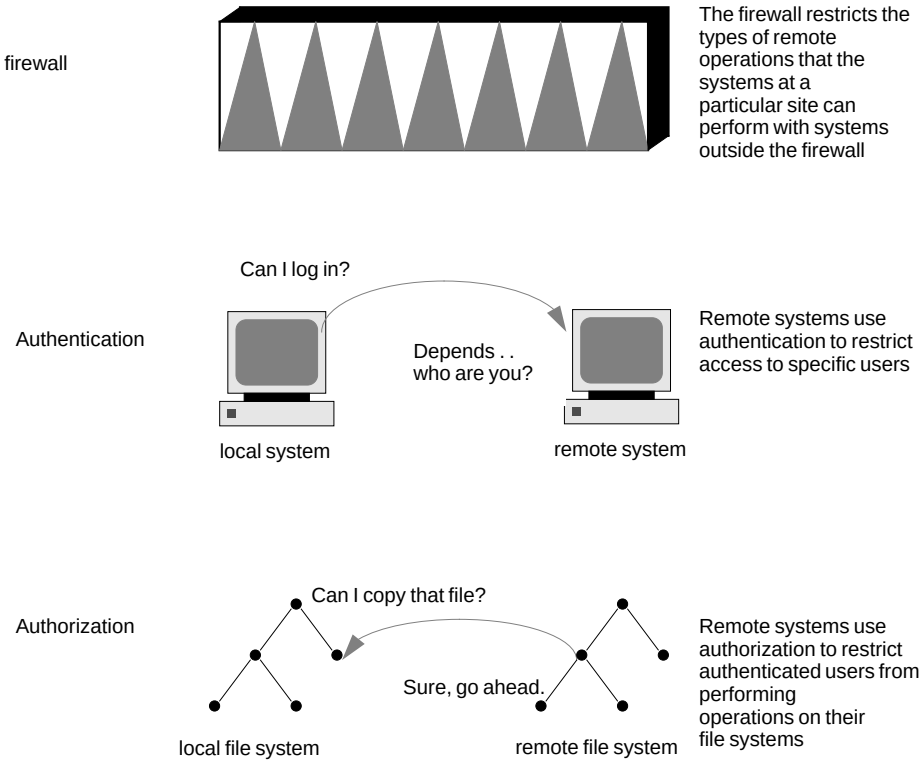


Figure 16–1 Security Restrictions for Remote Operations

Firewall Systems

You can set up a firewall system to protect the resources in your network from outside access. A *firewall system* is a secure host that acts as a barrier between your internal network and outside networks.

The firewall has two functions. It acts as a gateway which passes data between the networks, and it acts as a barrier which blocks the free passage of data to and from the network. The firewall requires a user on the internal network to log in to the firewall system to access hosts on remote networks. Similarly, a user on an outside network must log in to the firewall system before being granted access to a host on the internal network.

In addition, all electronic mail sent from the internal network is sent to the firewall system for transfer to a host on an external network. The firewall system receives all incoming electronic mail, and distributes it to the hosts on the internal network.

Caution - A firewall prevents unauthorized users from accessing hosts on your network. You should maintain strict and rigidly enforced security on the firewall, but security on other hosts on the network can be more relaxed. However, an intruder who can break into your firewall system can then gain access to all the other hosts on the internal network.

A firewall system should not have any *trusted hosts*. (A trusted host is one from which a user can log in without being required to type in a password.) It should not share any of its file systems, or mount any file systems from other servers.

ASET can be used to make a system into a firewall, and to enforce high security on a firewall system, as described in Chapter 24.

Packet Smashing

Most local area networks transmit data between computers in blocks called packets. Through a procedure called *packet smashing*, unauthorized users can harm or destroy data. Packet smashing involves capturing packets before they reach their destination, injecting arbitrary data into the contents, then sending the packets back on their original course. On a local area network, packet smashing is impossible because packets reach all systems, including the server, at the same time. Packet smashing is possible on a gateway, however, so make sure all gateways on the network are protected.

The most dangerous attacks are those that affect the integrity of the data. Such attacks involve changing the contents of the packets or impersonating a user. Attacks that involve eavesdropping—recording conversations and replaying them later without impersonating a user—do not compromise data integrity. These attacks do affect privacy, however. You can protect the privacy of sensitive information by encrypting data that goes over the network.

Authentication and Authorization

Authentication is a way to restrict access to specific users when accessing a remote system, which can be set up at both the system or network level. Once a user gains access to a remote system, *authorization* is a way to restrict operations that the user can perform on the remote system. The table below lists the types of authentications and authorizations that can help protect your systems on the network against unauthorized use.

TABLE 16–4 Types of Authentication and Authorization

Type	Description	Where to Find Information
NIS+	The NIS+ name service can provide both authentication and authorization at the network level.	*Solaris Naming Administration Guide*
Remote Login Programs	The remote login programs (`rlogin`, `rcp`, `ftp`) enable users to log in to a remote system over the network and use its resources. If you are a "trusted host," authentication is automatic; otherwise, you are asked to authenticate yourself.	Chapter 10
Secure RPC	Secure RPC improves the security of network environments by authenticating users who make requests on remote systems. You can use either the UNIX, DES, or Kerberos authentication system for Secure RPC.	*System Administration Guide, Volume 3*
	Secure RPC can also be used to provide additional security to the NFS™ environment, called Secure NFS.	"NFS Services and Secure RPC" on page 350
DES Encryption	The Data Encryption Standard (DES) encryption functions use a 56-bit key to encrypt a secret key.	"DES Encryption" on page 350
Diffie-Hellman Authentication	This authentication method is based on the ability of the sending system to use the common key to encrypt the current time, which the receiving system can decrypt and check against its current time.	"Diffie-Hellman Authentication" on page 351
Kerberos Version 4	Kerberos uses DES encryption to authenticate a user when logging in to the system.	Chapter 20
AdminSuite 2.3	The AdminSuite 2.3 tools provide authentication and authorization mechanisms to remotely manage systems.	*Solstice AdminSuite 2.3 Administration Guide*

Sharing Files

A network file server can control which files are available for sharing. It can also control which clients have access to the files, and what type of access is permitted to those clients. In general, the file server can grant read/write or read-only access either to all clients or to specific clients. Access control is specified when resources are made available with the `share` command.

A server can use the `/etc/dfs/dfstab` file to list the file systems it makes available to clients on the network. See the *System Administration Guide, Volume 3* for more information about sharing files.

Restricting Superuser (Root) Access

In general, superuser is not allowed root access to file systems shared across the network. Unless the server specifically grants superuser privileges, a user who is logged in as superuser on a client cannot gain root access to files that are remotely mounted on the client. The NFS system implements this by changing the user ID of the requester to the user ID of the user name, `nobody`; this is generally `60001`. The access rights of user `nobody` are the same as those given to the public (or a user without credentials) for a particular file. For example, if the public has only execute permission for a file, then user `nobody` can only execute that file.

An NFS server can grant superuser privileges on a shared file system on a per-host basis, using the `root=`*hostname* option to the `share` command.

Using Privileged Ports

If you do not want to run Secure RPC, a possible substitute is the Solaris “privileged port” mechanism. A privileged port is built up by the superuser with a port number of less than 1024. After a client system has authenticated the client’s credential, it builds a connection to the server via the privileged port. The server then verifies the client credential by examining the connection’s port number.

Non-Solaris clients, however, might not be able to communicate via the privileged port. If they cannot, you will see error messages such as these:

```
‘‘Weak Authentication
NFS request from unprivileged port’’
```

Using Automated Security Enhancement Tool (ASET)

The ASET security package provides automated administration tools that enable you to control and monitor your system's security. You specify a security level—low, medium, or high—at which ASET will run. At each higher level, ASET's file-control functions increase to reduce file access and tighten your system security.

See Chapter 24 for more information.

CHAPTER 17

Securing Files (Tasks)

This chapter describes the procedures for securing files. This is a list of the step-by-step instructions in this chapter.

- "How to Display File Information" on page 299
- "How to Change the Owner of a File" on page 301
- "How to Change Group Ownership of a File" on page 302
- "How to Change Permissions in Absolute Mode" on page 306
- "How to Change Special Permissions in Absolute Mode" on page 307
- "How to Change Permissions in Symbolic Mode" on page 308
- "How to Find Files With `setuid` Permissions" on page 309
- "How to Disable Programs From Using Executable Stacks" on page 311
- "How to Set an ACL on a File" on page 315
- "How to Copy an ACL" on page 317
- "How to Check If a File Has an ACL" on page 317
- "How to Modify ACL Entries on a File" on page 318
- "How to Delete ACL Entries From a File" on page 319
- "How to Display ACL Entries for a File" on page 320

File Security Features

This section describes the features that constitute a file's security.

User Classes

For each file, there are three classes of users that specify the levels of security:

- The file or directory owner—usually the user who created the file. The owner of a file can decide who has the right to read it, to write to it (make changes to it), or, if it is a command, to execute it.
- Members of a group.
- All others who are not the file or group owner.

Only the owner of the file or root can assign or modify file permissions.

File Permissions

The table below lists and describes the permissions you can give to each user class for a file.

TABLE 17–1 File Permissions

Symbol	Permission	Means Designated Users ...
`r`	Read	Can open and read the contents of a file
`w`	Write	Can write to the file (modify its contents), add to it, or delete it
`x`	Execute	Can execute the file (if it is a program or shell script), or run it with one of the `exec`(1) system calls
`-`	Denied	Cannot read, write, or execute the file

These file permissions apply to special files such as devices, sockets, and named pipes (FIFOs), as they do to regular files.

For a symbolic link, the permissions that apply are those of the file the link points to.

Directory Permissions

The table below lists and describes the permissions you can give to each user class for a directory.

TABLE 17–2 Directory Permissions

Symbol	Permission	Means Designated Users Can ...
r	Read	List files in the directory.
w	Write	Add or remove files or links in the directory.
x	Execute	Open or execute files in the directory. Also can make the directory and the directories beneath it current.

You can protect the files in a directory (and in its subdirectories) by disallowing access to that directory. Note, however, that superuser has access to all files and directories on the system.

Special File Permissions (`setuid`, `setgid` and Sticky Bit)

Three special types of permissions are available for executable files and public directories. When these permissions are set, any user who runs that executable file assumes the user ID of the owner (or group) of the executable file.

You must be extremely careful when setting special permissions, because special permissions constitute a security risk. For example, a user can gain superuser permission by executing a program that sets the user ID to root. Also, all users can set special permissions for files they own, which constitutes another security concern.

You should monitor your system for any unauthorized use of the `setuid` and `setgid` permissions to gain superuser privileges. See “How to Find Files With `setuid` Permissions” on page 309 to search for the file systems and print out a list of all programs using these permissions. A suspicious listing would be one that grants ownership of such a program to a user rather than to `root` or `bin`.

`setuid` Permission

When set-user identification (`setuid`) permission is set on an executable file, a process that runs this file is granted access based on the owner of the file (usually root), rather than the user who is running the executable file. This allows a user to access files and directories that are normally only available to the owner. For example, the `setuid` permission on the `passwd` command makes it possible for a user to change passwords, assuming the permissions of the root ID:

```
-r-sr-sr-x   3 root     sys       104580 Sep 16 12:02 /usr/bin/passwd
```

This presents a security risk, because some determined users can find a way to maintain the permissions granted to them by the `setuid` process even after the process has finished executing.

Note - Using `setuid` permissions with the reserved UIDs (0-99) from a program might not set the effective UID correctly. Use a shell script instead or avoid using the reserved UIDs with `setuid` permissions.

`setgid` Permission

The set-group identification (`setgid`) permission is similar to `setuid`, except that the process's effective group ID (GID) is changed to the group owner of the file, and a user is granted access based on permissions granted to that group. The `/usr/bin/mail` program has `setgid` permissions:

```
-r-x--s--x   1 root     mail       63628 Sep 16 12:01 /usr/bin/mail
```

When `setgid` permission is applied to a directory, files created in this directory belong to the group to which the directory belongs, not the group to which the creating process belongs. Any user who has write and execute permissions in the directory can create a file there—however, the file belongs to the group owning the directory, not to the user's group ownership.

You should monitor your system for any unauthorized use of the `setuid` and `setgid` permissions to gain superuser privileges. See "How to Find Files With `setuid` Permissions" on page 309 to search for the file systems and print out a list of all programs using these permissions. A suspicious listing would be one that grants ownership of such a program to a user rather than to `root` or `bin`.

Sticky Bit

The *sticky bit* is a permission bit that protects the files within a directory. If the directory has the sticky bit set, a file can be deleted only by the owner of the file, the owner of the directory, or by root. This prevents a user from deleting other users' files from public directories such as `/tmp`:

```
drwxrwxrwt 7  root  sys   400 Sep  3 13:37 tmp
```

Be sure to set the sticky bit manually when you set up a public directory on a TMPFS file system.

Default umask

When you create a file or directory, it has a default set of permissions. These default permissions are determined by the value of umask(1) in the system file /etc/profile, or in your .cshrc or .login file. By default, the system sets the permissions on a text file to 666, granting read and write permission to user, group, and others, and to 777 on a directory or executable file.

The value assigned by umask is subtracted from the default. This has the effect of denying permissions in the same way that chmod grants them. For example, while the command chmod 022 grants write permission to group and others, umask 022 denies write permission for group and others.

The table below shows some typical umask settings, and the effect on an executable file.

TABLE 17–3 umask Settings for Different Security Levels

Level of Security	umask	**Disallows**
Permissive (744)	022	w for group and others
Moderate (740)	027	w for group, rwx for others
Moderate (741)	026	w for group, rw for others
Severe (700)	077	rwx for group and others

Displaying File Information

This section describes how to display file information.

▼ How to Display File Information

Display information about all the files in a directory by using the ls command.

```
$ ls -la
```

`-l`	Displays the long format.
`-a`	Displays all files, including hidden files that begin with a dot (.).

Each line in the display has the following information about a file:

- Type of file

  A file can be one of seven types. The table below lists the possible file types.

TABLE 17–4 File Types

Symbol	Type
`-`	Text or program
`d`	Directory
`b`	Block special file
`c`	Character special file
`p`	Named pipe (FIFO)
`l`	Symbolic link
`s`	Socket

- Permissions; see Table 17–1 and Table 17–2 for descriptions
- Number of hard links
- Owner of the file
- Group of the file
- Size of the file, in bytes
- Date the file was created or last date it was changed
- Name of the file

Example—Displaying File Information

The following example displays the partial list of the files in the `/sbin` directory.

```
$ cd /sbin
$ ls -la
total 13456
drwxr-xr-x   2 root     sys          512 Sep  1 14:11 .
drwxr-xr-x  29 root     root        1024 Sep  1 15:40 ..
-r-xr-xr-x   1 root     bin       218188 Aug 18 15:17 autopush
lrwxrwxrwx   1 root     root          21 Sep  1 14:11 bpgetfile -> ...
-r-xr-xr-x   1 root     bin       505556 Aug 20 13:24 dhcpagent
-r-xr-xr-x   1 root     bin       456064 Aug 20 13:25 dhcpinfo
-r-xr-xr-x   1 root     bin       272360 Aug 18 15:19 fdisk
-r-xr-xr-x   1 root     bin       824728 Aug 20 13:29 hostconfig
-r-xr-xr-x   1 root     bin       603528 Aug 20 13:21 ifconfig
-r-xr-xr-x   1 root     sys       556008 Aug 20 13:21 init
-r-xr-xr-x   2 root     root      274020 Aug 18 15:28 jsh
-r-xr-xr-x   1 root     bin       238736 Aug 21 19:46 mount
-r-xr-xr-x   1 root     sys         7696 Aug 18 15:20 mountall
   .
   .
   .
```

Changing File Ownership

This section describes how to change the ownership of a file.

▼ How to Change the Owner of a File

1. **Become superuser.**

   By default, the owner cannot use the `chown` command to change the owner of a file or directory. However, you can enable the owner to use `chown` by adding the following line to the system's `/etc/system` file and rebooting the system.

   ```
   set rstchown = 0
   ```

   See `chown`(1) for more details. Also, be aware that there can be other restrictions on changing ownership on NFS-mounted file systems.

2. **Change the owner of a file by using the `chown` command.**

   ```
   # chown newowner filename
   ```

newowner	Specifies the user name or UID of the new owner of the file or directory.
filename	Specifies the file or directory.

3. Verify the owner of the file is changed.

```
# ls -l filename
```

Example—Changing the Owner of a File

The following example sets the ownership on `myfile` to the user `rimmer`.

```
# chown rimmer myfile
# ls -l myfile
-rw-r--r--   1 rimmer   scifi   112640 May 24 10:49 myfile
```

▼ How to Change Group Ownership of a File

1. Become superuser.

By default, the owner can only use the `chgrp` command to change the group of a file to a group in which the owner belongs. For example, if the owner of a file only belongs to the `staff` and `sysadm` groups, the owner can only change the group of a file to `staff` or `sysadm` group.

However, you can enable the owner to change the group of a file to a group in which the owner doesn't belong by adding the following line to the system's `/etc/system` file and rebooting the system.

```
set rstchown = 0
```

See `chgrp`(1) for more details. Also, be aware that there can be other restrictions on changing groups on NFS-mounted file systems.

2. Change the group owner of a file by using the `chgrp` command.

```
$ chgrp group filename
```

group	Specifies the group name or GID of the new group of the file or directory.
filename	Specifies the file or directory.

See “Setting Up and Maintaining User Accounts and Groups (Tasks)” in *System Administration Guide, Volume 1* for information on setting up groups.

3. **Verify the group owner of the file is changed.**

```
$ ls -l filename
```

Example—Changing Group Ownership of a File

The following example sets the group ownership on `myfile` to the group `scifi`.

```
$ chgrp scifi myfile
$ ls -l myfile
-rwxrw-- 1 rimmer scifi 12985 Nov 12 16:28 myfile
```

Changing File Permissions

The `chmod` command enables you to change the permissions on a file. You must be superuser or the owner of a file or directory to change its permissions.

You can use the `chmod` command to set permissions in either of two modes:

- **Absolute Mode -** Use numbers to represent file permissions (the method most commonly used to set permissions). When you change permissions by using the absolute mode, represent permissions for each triplet by an octal mode number.
- **Symbolic Mode -** Use combinations of letters and symbols to add or remove permissions.

The table below lists the octal values for setting file permissions in absolute mode. You use these numbers in sets of three to set permissions for owner, group, and other (in that order). For example, the value 644 sets read/write permissions for owner, and read-only permissions for group and other.

TABLE 17–5 Setting File Permissions in Absolute Mode

Octal Value	File Permissions Set	Permissions Description
0	---	No permissions
1	--x	Execute permission only
2	-w-	Write permission only
3	-wx	Write and execute permissions
4	r--	Read permission only
5	r-x	Read and execute permissions
6	rw-	Read and write permissions
7	rwx	Read, write, and execute permissions

You can set special permissions on a file in absolute or symbolic modes. In absolute mode, you set special permissions by adding a new octal value to the left of the permission triplet. The table below lists the octal values to set special permissions on a file.

TABLE 17–6 Setting Special Permissions in Absolute Mode

Octal Value	Special Permissions Set
1	Sticky bit
2	setguid
4	setuid

The table below lists the symbols for setting file permissions in symbolic mode. Symbols can specify whose permissions are to be set or changed, the operation to be performed, and the permissions being assigned or changed.

TABLE 17–7 Setting File Permissions in Symbolic Mode

Symbol	Function	Description
`u`	Who	User (owner)
`g`	Who	Group
`o`	Who	Others
`a`	Who	All
`=`	Operation	Assign
`+`	Operation	Add
`-`	Operation	Remove
`r`	Permission	Read
`w`	Permission	Write
`x`	Permission	Execute
`l`	Permission	Mandatory locking, `setgid` bit is on, group execution bit is off
`s`	Permission	`setuid` or `setgid` bit is on
`S`	Permission	`suid` bit is on, user execution bit is off
`t`	Permission	Sticky bit is on, execution bit for others is on
`T`	Permission	Sticky bit is on, execution bit for others is off

The *who operator permission* designations in the function column specifies the symbols that change the permissions on the file or directory.

who	Specifies whose permissions are changed.
operator	Specifies the operation to perform.
permissions	Specifies what permissions are changed.

▼ How to Change Permissions in Absolute Mode

1. **If you are not the owner of the file or directory, become superuser.**

   Only the current owner or superuser can use the `chmod` command to change file permissions on a file or directory.

2. **Change permissions in absolute mode by using the `chmod` command.**

```
$ chmod nnn filename
```

nnn	Specifies the octal values that represent the permissions for the file owner, file group, and others, in that order. See Table 17–5 for the list of valid octal values.
filename	Specifies the file or directory.

Note - If you use `chmod` to change the file group permissions on a file with ACL entries, both the file group permissions and the ACL mask are changed to the new permissions. Be aware that the new ACL mask permissions can change the effective permissions for additional users and groups who have ACL entries on the file. Use the `getfacl`(1) command to make sure the appropriate permissions are set for all ACL entries.

---

3. **Verify the permissions of the file have changed.**

```
$ ls -l filename
```

Example—Changing Permissions in Absolute Mode

The following example shows changing the permissions of a public directory from 744 (read / write / execute, read-only, and read-only) to 755 (read / write / execute, read / execute, and read / execute).

```
$ ls -ld public_dir
drwxr--r--  1 ignatz   staff    6023 Aug  5 12:06 public_dir
$ chmod 755 public_dir
$ ls -ld public_dir
drwxr-xr-x  1 ignatz   staff    6023 Aug  5 12:06 public_dir
```

The following example shows changing the permissions of an executable shell script from read / write to read / write / execute.

```
$ ls -l my_script
-rw------- 1 ignatz   staff    6023 Aug  5 12:06 my_script
$ chmod 700 my_script
$ ls -l my_script
-rwx------ 1 ignatz   staff    6023 Aug  5 12:06 my_script
```

▼ How to Change Special Permissions in Absolute Mode

1. **If you are not the owner of the file or directory, become superuser.**

   Only the current owner or superuser can use the `chmod` command to change the special permissions on a file or directory.

2. **Change special permissions in absolute mode by using the `chmod` command.**

```
$ chmod nnnn filename
```

nnnn	Specifies the octal values that change the permissions on the file or directory. The first octal value on the left sets the special permissions on the file. See Table 17–6 for the list of valid octal values for the special permissions.
filename	Specifies the file or directory.

Note - If you use `chmod` to change the file group permissions on a file with ACL entries, both the file group permissions and the ACL mask are changed to the new permissions. Be aware that the new ACL mask permissions can change the effective permissions for additional users and groups who have ACL entries on the file. Use the `getfacl`(1) command to make sure the appropriate permissions are set for all ACL entries.

3. **Verify the permissions of the file have changed.**

```
$ ls -l filename
```

Examples—Setting Special Permissions in Absolute Mode

The following example sets `setuid` permission on the `dbprog` file.

```
$ chmod 4555 dbprog
$ ls -l dbprog
-r-sr-xr-x   1 db     staff        12095 May  6 09:29 dbprog
```

The following example sets `setgid` permission on the `dbprog2` file.

```
$ chmod 2551 dbprog2
$ ls -l dbprog2
-r-xr-s--x   1 db     staff       24576 May  6 09:30 dbprog2
```

The following example sets sticky bit permission on the `pubdir` directory.

```
$ chmod 1777 pubdir
```

▼ How to Change Permissions in Symbolic Mode

1. **If you are not the owner of the file or directory, become superuser.**

   Only the current owner or superuser can use the `chmod` command to change file permissions on a file or directory.

2. **Change permissions in symbolic mode by using the `chmod` command.**

```
$ chmod who operator permission filename
```

who operator permission	*who* specifies whose permissions are changed, *operator* specifies the operation to perform, and *permission* specifies what permissions are changed. See Table 17–7 for the list of valid symbols.
filename	Specifies the file or directory.

3. **Verify the permissions of the file have changed.**

```
$ ls -l filename
```

Examples—Changing Permissions in Symbolic Mode

The following example takes away `read` permission from others.

```
$ chmod o-r filea
```

The following example adds `read` and `execute` permissions for user, group, and others.

```
$ chmod a+rx fileb
```

The following example assigns `read`, `write`, and `execute` permissions to group.

```
$ chmod g=rwx filec
```

Searching for Special Permissions

You should monitor your system for any unauthorized use of the `setuid` and `setgid` permissions to gain superuser privileges. A suspicious listing would be one that grants ownership of such a program to a user rather than to `root` or `bin`.

▼ How to Find Files With `setuid` Permissions

1. **Become superuser.**

2. **Find files with `setuid` permissions set by using the `find` command.**

```
# find directory -user root -perm -4000 -exec ls -ldb {} \; >/tmp/filename
```

`find` *directory*	Checks all mounted paths starting at the specified *directory*, which can be root (`/`), `sys`, `bin`, or `mail`.
`–user root`	Displays files only owned by root.
`–perm –4000`	Displays files only with permissions set to 4000.
`–exec ls –ldb`	Displays the output of the `find` command in `ls -ldb` format.
`>/tmp/`*filename*	Writes results to this file.

3. Display the results in `/tmp/`*filename*.

If you need background information about `setuid` permissions, see "`setuid` Permission" on page 297.

Example—Finding Files With `setuid` Permissions

```
# find / -user root -perm -4000 -exec ls -ldb {} \; > /tmp/ckprm
# cat /tmp/ckprm
-r-sr-xr-x 1 root bin 38836 Aug 10 16:16 /usr/bin/at
-r-sr-xr-x 1 root bin 19812 Aug 10 16:16 /usr/bin/crontab
---s--x--x 1 root sys 46040 Aug 10 15:18 /usr/bin/ct
-r-sr-xr-x 1 root sys 12092 Aug 11 01:29 /usr/lib/mv_dir
-r-sr-sr-x 1 root bin 33208 Aug 10 15:55 /usr/lib/lpadmin
-r-sr-sr-x 1 root bin 38696 Aug 10 15:55 /usr/lib/lpsched
---s--x--- 1 root rar 45376 Aug 18 15:11 /usr/rar/bin/sh
-r-sr-xr-x 1 root bin 12524 Aug 11 01:27 /usr/bin/df
-rwsr-xr-x 1 root sys 21780 Aug 11 01:27 /usr/bin/newgrp
-r-sr-sr-x 1 root sys 23000 Aug 11 01:27 /usr/bin/passwd
-r-sr-xr-x 1 root sys 23824 Aug 11 01:27 /usr/bin/su
 #
```

An unauthorized user (`rar`) has made a personal copy of `/usr/bin/sh`, and has set the permissions as `setuid` to root. This means that `rar` can execute `/usr/rar/bin/sh` and become the privileged user. If you want to save this output for future reference, move the file out of the `/tmp` directory.

Executable Stacks and Security

A number of security bugs are related to default executable stacks when their permissions are set to read, write and execute. While stacks with execute permissions set are mandated by the SPARC ABI and Intel ABI, most programs can function correctly without using executable stacks.

The `noexec_user_stack` variable (available starting in the Solaris 2.6 release) enables you to specify whether stack mappings are executable or not. By default, the variable is zero, which provides ABI-compliant behavior. If the variable is set to non-zero, the system will mark the stack of every process in the system as readable and writable, but not executable.

Once this variable is set, programs that attempt to execute code on their stack will be sent a `SIGSEGV` signal, which usually results in the program terminating with a core dump. Such programs also generate a warning message that includes the name of the offending program, the process ID, and real UID of the user who ran the program. For example:

```
a.out[347] attempt to execute code on stack by uid 555
```

The message is logged by the `syslogd`(1M) daemon when the `syslog kern` facility is set to `notice` level. This logging is set by default in the `syslog.conf`(4) file, which means the message is sent to both the console and to the `/var/adm/messages` file.

This message is useful both for observing potential security problems, as well as to identify valid programs that depend upon executable stacks which have been prevented from correct operation by setting this variable. If the administrator does not want any messages logged, then the `noexec_user_stack_log` variable can be set to zero to disable it in the `/etc/system` file, though the `SIGSEGV` signal can continue to cause the executing program to core dump.

You can use `mprotect`(2) if you want programs to explicitly mark their stack as executable.

Because of hardware limitations, the capability of catching and reporting executable stack problems is only available on sun4m, sun4d and sun4u platforms.

▼ How to Disable Programs From Using Executable Stacks

1. **Become superuser.**

2. **Edit the `/etc/system` file and add the following line.**

```
set noexec_user_stack=1
```

3. **Reboot the system.**

```
# init 6
```

▼ How to Disable Executable Stack Message Logging

1. **Become superuser.**

2. **Edit the** `/etc/system file` **and add the following line.**

```
set noexec_user_stack_log=0
```

3. **Reboot the system.**

```
# init 6
```

Using Access Control Lists (ACLs)

Traditional UNIX file protection provides read, write, and execute permissions for the three user classes: file owner, file group, and other. An ACL provides better file security by enabling you to define file permissions for the file owner, file group, other, specific users and groups, and default permissions for each of those categories.

For example, if you wanted everyone in a group to be able to read a file, you would simply give group read permissions on that file. Now, assume you wanted only one person in the group to be able to write to that file. Standard UNIX doesn't provide that level of file security. However, this dilemma is perfect for ACLs.

ACL entries are the way to define an ACL on a file, and they are set through the `setfacl`(1) command. ACL entries consist of the following fields separated by colons:

```
entry_type:[uid|gid]:perms
```

entry_type	Type of ACL entry on which to set file permissions. For example, *entry_type* can be `user` (the owner of a file) or `mask` (the ACL mask).
uid	User name or identification number.
gid	Group name or identification number.
perms	Represents the permissions that are set on *entry_type*. *perms* can be indicated by the symbolic characters `rwx` or a number (the same permissions numbers used with the `chmod` command).

The following example shows an ACL entry that sets read/write permissions for the user `nathan`.

```
user:nathan:rw-
```

Caution - UFS file system attributes such as ACLs are supported in UFS file systems only. This means that if you restore or copy files with ACL entries into the `/tmp` directory, which is usually mounted as a TMPFS file system, the ACL entries will be lost. Use the `/var/tmp` directory for temporary storage of UFS files.

ACL Entries for Files

The table below lists the valid ACL entries. The first three ACL entries provide the basic UNIX file protection.

TABLE 17–8 ACL Entries for Files

ACL Entry	Description
`u[ser]::`*perms*	File owner permissions.
`g[roup]::`*perms*	File group permissions.
`o[ther]:`*perms*	Permissions for users other than the file owner or members of file group.

TABLE 17–8 ACL Entries for Files *(continued)*

ACL Entry	Description
`m[ask]:`*perms*	The ACL mask. The mask entry indicates the maximum permissions allowed for users (other than the owner) and for groups. The mask is a quick way to change permissions on all the users and groups. For example, the `mask:r--` mask entry indicates that users and groups cannot have more than read permissions, even though they might have write/execute permissions.
`u[ser]:`*uid:perms*	Permissions for a specific user. For *uid*, you can specify either a user name or a numeric UID.
`g[roup]:`*gid:perms*	Permissions for a specific group. For *gid*, you can specify either a group name or a numeric GID.

ACL Entries for Directories

In addition to the ACL entries described in Table 17–8, you can set default ACL entries on a directory. Files or directories created in a directory that has default ACL entries will have the same ACL entries as the default ACL entries. The table below lists the default ACL entries for directories.

When you set default ACL entries for specific users and groups on a directory for the first time, you must also set default ACL entries for the file owner, file group, others, and the ACL mask (these are required and are the first four default ACL entries in the table below).

TABLE 17–9 Default ACL Entries for Directories

Default ACL Entry	Description
`d[efault]:u[ser]::`*perms*	Default file owner permissions.
`d[efault]:g[roup]::`*perms*	Default file group permissions.
`d[efault]:o[ther]:`*perms*	Default permissions for users other than the file owner or members of the file group.

TABLE 17–9 Default ACL Entries for Directories *(continued)*

Default ACL Entry	Description
`d[efault]:m[ask]:`*perms*	Default ACL mask.
`d[efault]:u[ser]:`*uid:perms*	Default permissions for a specific user. For *uid*, you can specify either a user name or a numeric UID.
`d[efault]:g[roup]:`*gid:perms*	Default permissions for a specific group. For *gid*, you can specify either a group name or a numeric GID.

▼ How to Set an ACL on a File

1. Set an ACL on a file by using the `setfacl` command.

```
$ setfacl -s user::perms,group::perms,other:perms,mask:perms,acl_entry_list  filename ...
```

`–s`	Sets an ACL on the file. If a file already has an ACL, it is replaced. This option requires at least the file owner, file group, and other entries.
`user::`*perms*	Specifies the file owner permissions.
`group::`*perms*	Specifies the file group permissions.
`other:`*perms*	Specifies the permissions for users other than the file owner or members of the file group.
`mask:`*perms*	Specifies the permissions for the ACL mask. The mask indicates the maximum permissions allowed for users (other than the owner) and for groups.
acl_entry_list	Specifies the list of one or more ACL entries to set for specific users and groups on the file or directory. You can also set default ACL entries on a directory. Table 17–8 and Table 17–9 show the valid ACL entries.
filename	Specifies one or more files or directories on which to set the ACL.

2. **To verify that an ACL was set on the file, see "How to Check If a File Has an ACL" on page 317. To verify which ACL entries were set on the file, use the `getfacl` command.**

```
$ getfacl filename
```

Caution - If an ACL already exists on the file, the `–s` option will replace the entire ACL with the new ACL.

Examples—Setting an ACL on a File

The following example sets the file owner permissions to read/write, file group permissions to read only, and other permissions to none on the `ch1.doc` file. In addition, the user `george` is given read/write permissions on the file, and the ACL mask permissions are set to read/write, which means no user or group can have execute permissions.

```
$ setfacl -s user::rw-,group::r--,other:---,mask:rw-,user:george:rw- ch1.doc
$ ls -l
total 124
-rw-r-----+  1 nathan  sysadmin   34816 Nov 11 14:16 ch1.doc
-rw-r--r--   1 nathan  sysadmin   20167 Nov 11 14:16 ch2.doc
-rw-r--r--   1 nathan  sysadmin    8192 Nov 11 14:16 notes
$ getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group: sysadmin
user::rw-
user:george:rw-    #effective:rw-
group::r--         #effective:r--
mask:rw-
other:---
```

The following example sets the file owner permissions to read/write/execute, file group permissions to read only, other permissions to none, and the ACL mask permissions to read on the `ch2.doc` file. In addition, the user `george` is given read/write permissions; however, due to the ACL mask, the effective permissions for `george` are read only.

```
$ setfacl -s u::7,g::4,o:0,m:4,u:george:7 ch2.doc
$ getfacl ch2.doc
# file: ch2.doc
# owner: nathan
# group: sysadmin
user::rwx
user:george:rwx         #effective:r--
group::r--              #effective:r--
mask:r--
other:---
```

▼ How to Copy an ACL

Copy a file's ACL to another file by redirecting the `getfacl` output.

```
$ getfacl filename1 | setfacl -f - filename2
```

filename1 Specifies the file from which to copy the ACL.

filename2 Specifies the file on which to set the copied ACL.

Example—Copying an ACL

The following example copies the ACL on `ch2.doc` to `ch3.doc`.

```
$ getfacl ch2.doc | setfacl -f - ch3.doc
```

▼ How to Check If a File Has an ACL

Check if a file has an ACL by using the `ls` command.

```
$ ls -l filename
```

filename Specifies the file or directory.

A plus sign (+) to the right of the mode field indicates the file has an ACL.

Note - Unless you have added ACL entries for additional users or groups on a file, a file is considered to be a "trivial" ACL and the + will not display.

Example—Checking If a File Has an ACL

The following example shows that `ch1.doc` has an ACL, because the listing has. a '+' to the right of the mode field.

```
$ ls -l ch1.doc
-rwxr-----+  1 nathan   sysadmin      167 Nov 11 11:13 ch1.doc
```

▼ How to Modify ACL Entries on a File

1. Modify ACL entries on a file by using the `setfacl` command.

```
$ setfacl -m acl_entry_list filename1 [filename2 ...]
```

`–m`	Modifies the existing ACL entry.
acl_entry_list	Specifies the list of one or more ACL entries to modify on the file or directory. You can also modify default ACL entries on a directory. Table 17–8 and Table 17–9 show the valid ACL entries.
filename ...	Specifies one or more files or directories.

2. To verify that the ACL entries were modified on the file, use the `getfacl` command.

```
$ getfacl filename
```

Examples—Modifying ACL Entries on a File

The following example modifies the permissions for the user `george` to read/write.

```
$ setfacl -m user:george:6 ch3.doc
$ getfacl ch3.doc
# file: ch3.doc
# owner: nathan
# group: staff
user::rw-
user::george:rw-    #effective:r--
group::r-               #effective:r--
mask:r--
```

(continued)

```
other:r-
```

The following example modifies the default permissions for the group `staff` to read and the default ACL mask permissions to read/write on the `book` directory.

```
$ setfacl -m default:group:staff:4,default:mask:6 book
```

▼ How to Delete ACL Entries From a File

1. **Delete ACL entries from a file by using the `setfacl` command.**

```
$ setfacl -d acl_entry_list filename1 ...
```

–d	Deletes the specified ACL entries.
acl_entry_list	Specifies the list of ACL entries (without specifying the permissions) to delete from the file or directory. You can only delete ACL entries and default ACL entries for specific users and groups. Table 17–8 and Table 17–9 show the valid ACL entries.
filename ...	Specifies one or more files or directories.

Alternately, you can use the `setfacl -s` command to delete all the ACL entries on a file and replace them with the new ACL entries specified.

2. **To verify that the ACL entries were deleted from the file, use the `getfacl` command.**

```
$ getfacl filename
```

Example—Deleting ACL Entries on a File

The following example deletes the user `george` from the `ch4.doc` file.

```
$ setfacl -d user:george ch4.doc
```

▼ How to Display ACL Entries for a File

Display ACL entries for a file by using the `getfacl` command.

```
$ getfacl [-a | -d] filename1 ...
```

–a	Displays the file name, file owner, file group, and ACL entries for the specified file or directory.
–d	Displays the file name, file owner, file group, and default ACL entries for the specified directory.
filename ...	Specifies one or more files or directories.

If you specify multiple file names on the command line, the ACL entries are separated by a blank line.

Examples—Displaying ACL Entries for a File

The following example shows all the ACL entries for the `ch1.doc` file. The `#effective:` note beside the user and group entries indicates what the permissions are after being modified by the ACL mask.

```
$ getfacl ch1.doc
# file: ch1.doc
# owner: nathan
# group: sysadmin
user::rw-
user:george:r--         #effective:r--
group::rw-              #effective:rw-
mask:rw-
other:---
```

The following example shows the default ACL entries for the `book` directory.

```
$ getfacl -d book# file: book
# owner: nathan
# group: sysadmin
user::rwx
user:george:r-x         #effective:r-x
group::rwx              #effective:rwx
mask:rwx
other:---
default:user::rw-
default:user:george:r--
```

(continued)

(Continuation)

```
default:group::rw-
default:mask:rw-
default:other:---
```

CHAPTER 18

Securing Systems (Tasks)

This chapter describes the procedures for securing systems. This is a list of the step-by-step instructions in this chapter.

- "How to Display a User's Login Status" on page 323
- "How to Display Users Without Passwords" on page 325
- "How to Temporarily Disable User Logins" on page 326
- "How to Save Failed Login Attempts" on page 327
- "How to Create a Dial-up Password" on page 330
- "How to Temporarily Disable Dial-up Logins" on page 332
- "How to Restrict Superuser (root) Login to the Console" on page 332
- "How to Monitor Who Is Using the `su` Command" on page 333
- "How to Display Superuser (root) Access Attempts to the Console" on page 334
- "How to Disable or Enable a System's Abort Sequence" on page 334

For overview information about securing systems, see "System Security" on page 286.

Displaying Security Information

This section describes how to display user login information.

▼ How to Display a User's Login Status

1. **Become superuser.**

2. **Display a user's login status by using the `logins` command.**

```
# logins -x -l username
```

`-x`	Displays an extended set of login status information.
`-l` *username*	Displays login status for the specified user. *username* is a user's login name. Multiple login names must be specified in a comma-separated list.

The `logins`(1M) command uses the local `/etc/passwd` file and the NIS or NIS+ password databases to obtain a user's login status.

Example—Displaying a User's Login Status

The following example displays login status for the user `rimmer`.

```
# logins -x -l rimmer
rimmer       500     staff           10   Arnold J. Rimmer
                     /export/home/rimmer
                     /bin/sh
                     PS 010170 10 7 -1
```

`rimmer`	Identifies the user's login name.
`500`	Identifies the UID (user ID).
`staff`	Identifies the user's primary group.
`10`	Identifies the GID (group ID).
`Arnold J. Rimmer`	Identifies the comment.
`/export/home/rimmer`	Identifies the user's home directory.

`/bin/sh`	Identifies the login shell.
`PS 010170 10 7 -1`	Specifies the password aging information: ■ Last date password was changed ■ Number of days required between changes ■ Number of days allowed before a change is required ■ Warning period

▼ How to Display Users Without Passwords

You should make sure that all users have a valid password.

1. **Become superuser.**
2. **Display users who have no passwords by using the** `logins` **command.**

```
# logins -p
```

`–p`	Displays a list of users with no passwords.

The `logins` command uses the local `/etc/passwd` file and the NIS or NIS+ password databases to obtain a user's login status.

Example—Displaying Users Without Passwords

The following example displays that the user `pmorph` does not have a password.

```
# logins -p
pmorph          501     other           1       Polly Morph
#
```

Temporarily Disabling User Logins

You can temporarily disable user logins by:

- Creating the `/etc/nologin` file.
- Bringing the system to run level 0 (single-user mode). See "Shutting Down a System (Tasks)" in *System Administration Guide, Volume 1* for information on bringing the system to single-user mode.

Creating the /etc/nologin File

Create this file to disallow user logins and notify users when a system will be unavailable for an extended period of time due to a system shutdown or routine maintenance.

If a user attempts to log in to a system where this file exists, the contents of the nologin(4) file is displayed, and the user login is terminated. Superuser logins are not affected.

▼ How to Temporarily Disable User Logins

1. **Become superuser.**
2. **Create the /etc/nologin file using an editor.**

```
# vi /etc/nologin
```

3. **Include a message regarding system availability.**
4. **Close and save the file.**

Example—Disabling User Logins

This example shows how to notify users of system unavailability.

```
# vi /etc/nologin
(Add system message here)

# cat /etc/nologin
***No logins permitted.***

***The system will be unavailable until 12 noon.***
```

Saving Failed Login Attempts

You can save failed login attempts by creating the /var/adm/loginlog file with read and write permission for root only. After you create the loginlog file, all failed login activity will be written to this file automatically after five failed attempts. See “How to Save Failed Login Attempts” on page 327 for detailed instructions.

The loginlog file contains one entry for each failed attempt. Each entry contains the user’s login name, tty device, and time of the failed attempt. If a person makes fewer than five unsuccessful attempts, none of the attempts are logged.

The `loginlog` file may grow quickly. To use the information in this file and to prevent the file from getting too large, you must check and clear its contents occasionally. If this file shows a lot of activity, it may suggest an attempt to break into the computer system. For more information about this file, see `loginlog`(4).

▼ How to Save Failed Login Attempts

1. **Become superuser.**

2. **Create the `loginlog` file in the `/var/adm` directory.**

```
# touch /var/adm/loginlog
```

3. **Set read and write permissions for root on the `loginlog` file.**

```
# chmod 600 /var/adm/loginlog
```

4. **Change group membership to `sys` on the `loginlog` file.**

```
# chgrp sys /var/adm/loginlog
```

5. **Make sure the log works by attempting to log into the system five times with the wrong password after the `loginlog` file is created. Then display the `/var/adm/loginlog` file.**

```
# more /var/adm/loginlog
rimmer:/dev/pts/4:Mon Jul 12 13:52:15 1999
rimmer:/dev/pts/4:Mon Jul 12 13:52:23 1999
rimmer:/dev/pts/4:Mon Jul 12 13:52:31 1999
rimmer:/dev/pts/4:Mon Jul 12 13:52:39 1999
#
```

Password Protection Using Dial-up Passwords

You can add a layer of security to your password mechanism by requiring a *dial-up password* for users who access a system through a modem or dial-up port. A dial-up password is an additional password that a user must enter before being granted access to the system.

Only superuser can create or change a dial-up password. To ensure the integrity of the system, the password should be changed about once a month. The most effective use of this mechanism is to require a dial-up password to gain access to a gateway system.

Two files are involved in creating a dial-up password, `/etc/dialups` and `/etc/d_passwd`. The first contains a list of ports that require a dial-up password, and the second contains a list of shell programs that require an encrypted password as the additional dial-up password.

The `dialups`(4) file is a list of terminal devices, for example:

```
/dev/term/a
/dev/term/b
```

The `d_passwd`(4) file has two fields. The first is the login shell that will require a password, and the second is the encrypted password. The `/etc/dialups` and `/etc/d_passwd` files work like this:

When a user attempts to log in on any of the ports listed in `/etc/dialups`, the login program looks at the user's login entry stored in `/etc/passwd`, and compares the login shell to the entries in `/etc/d_passwd`. These entries determine whether the user will be required to supply the dial-up password.

```
/usr/lib/uucp/uucico:encrypted_password:
/usr/bin/csh:encrypted_password:
/usr/bin/ksh:encrypted_password:
/usr/bin/sh:encrypted_password:
```

The basic dial-up password sequence is shown in the figure below.

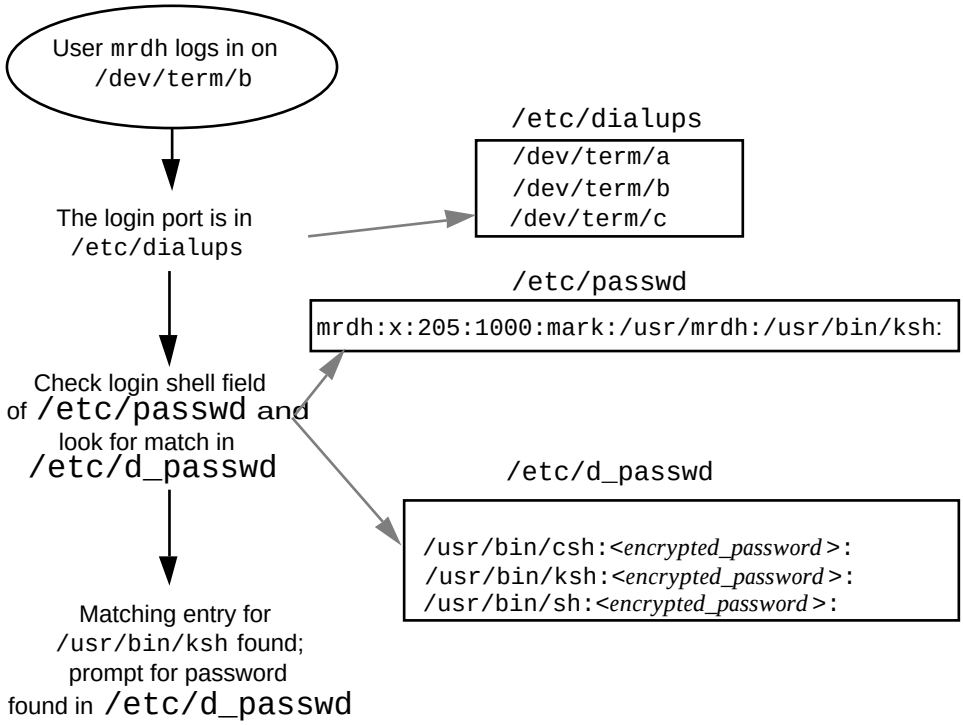


Figure 18–1 Basic Dial-up Password Sequence

The /etc/d_passwd File

Because most users will be running a shell when they log in, all shell programs should have entries in /etc/d_passwd. Such programs include uucico, sh, ksh, and csh. If some users run something else as their login shell, include that login shell in the file, too.

If the user's login program (as specified in /etc/passwd) is not found in /etc/d_passwd, or if the login shell field in /etc/passwd is null, the password entry for /usr/bin/sh is used.

- If the user's login shell in /etc/passwd matches an entry in /etc/d_passwd, the user must supply a dial-up password.
- If the user's login shell in /etc/passwd is not found in /etc/d_passwd, the user must supply the default password. The default password is the entry for /usr/bin/sh.
- If the login shell field in /etc/passwd is empty, the user must supply the default password (the entry for /usr/bin/sh).
- If /etc/d_passwd has no entry for /usr/bin/sh, then those users whose login shell field in /etc/passwd is empty or does not match any entry in /etc/d_passwd will not be prompted for a dial-up password.

- Dial-up logins are disabled if `/etc/d_passwd` has only the following entry: `/usr/bin/sh:*:`

▼ How to Create a Dial-up Password

Caution - When you first establish a dial-up password, be sure to remain logged in on at least one terminal while testing the password on a different terminal. If you make a mistake while installing the extra password and log off to test the new password, you might not be able to log back on. If you are still logged in on another terminal, you can go back and fix your mistake.

1. **Become superuser.**

2. **Create an `/etc/dialups` file containing a list of terminal devices, including all the ports that will require dial-up password protection.**

   The `/etc/dialups` file should look like this:

```
/dev/term/a

/dev/term/b

/dev/term/c
```

3. **Create an `/etc/d_passwd` file containing the login programs that will require a dial-up password, and the encrypted dial-up password.**

   Include shell programs that a user could be running at login, for example, `uucico`, `sh`, `ksh`, and `csh`. The `/etc/d_passwd` file should look like this:

```
/usr/lib/uucp/uucico:encrypted_password:

/usr/bin/csh:encrypted_password:

/usr/bin/ksh:encrypted_password:

/usr/bin/sh:encrypted_password:
```

4. **Set ownership to `root` on the two files.**

```
# chown root /etc/dialups /etc/d_passwd
```

5. **Set group ownership to `root` on the two files.**

```
# chgrp root /etc/dialups /etc/d_passwd
```

6. **Set read and write permissions for `root` on the two files.**

```
# chmod 600 /etc/dialups /etc/d_passwd
```

7. **Create the encrypted passwords.**

   a. **Create a temporary user.**

```
# useradd user-name
```

   b. **Create a password for the temporary user.**

```
# passwd user-name
```

   c. **Capture the encrypted password.**

```
# grep user-name /etc/shadow > user-name.temp
```

   d. **Edit the *user-name*`.temp` file.**

   Delete all fields except the encrypted password (the second field).

   For example, in the following line, the encrypted password is `U9gp9SyA/JlSk`.

```
temp:U9gp9SyA/JlSk:7967:::::7988:
```

   e. **Delete the temporary user.**

```
# userdel user-name
```

8. **Copy the encrypted password from *user-name*`.temp` file into the `/etc/d_passwd` file.**

   You can create a different password for each login shell, or use the same one for each.

▼ How to Temporarily Disable Dial-up Logins

1. **Become superuser.**

2. **Put the following entry by itself into the** `/etc/d_passwd` **file:**

```
/usr/bin/sh:*:
```

Restricting Superuser (root) Access on the Console

The superuser account is used by the operating system to accomplish basic functions, and has wide-ranging control over the entire operating system. It has access to and can execute essential system programs. For this reason, there are almost no security restraints for any program that is run by superuser.

You can protect the superuser account on a system by restricting access to a specific device through the `/etc/default/login` file. For example, if superuser access is restricted to the console, you can log in to a system as superuser only from the console. If anybody remotely logs in to the system to perform an administrative function, they must first log in with their user login and then use the `su`(1M) command to become superuser. See the section below for detailed instructions.

Note - Restricting superuser login to the console is set up by default when you install a system.

▼ How to Restrict Superuser (root) Login to the Console

1. **Become superuser.**

2. **Edit the** `/etc/default/login` **file.**

3. **Uncomment the following line.**

```
CONSOLE=/dev/console
```

   Any users who try to remotely log in to this system must first log in with their user login, and then use the `su` command to become superuser.

4. **Attempt to log in remotely as superuser to this system, and verify that the operation fails.**

Monitoring Who Is Using the su Command

You can start monitoring su attempts through the /etc/default/su file. Through this file, you can enable the /var/adm/sulog file to monitor each time the su command is used to change to another user. See "How to Monitor Who Is Using the su Command" on page 333 for step-by-step instructions.

The sulog file lists all uses of the su command, not only those used to switch user to superuser. The entries show the date and time the command was entered, whether or not it was successful (+ or -), the port from which the command was issued, and finally, the name of the user and the switched identity.

Through the /etc/default/su file, you can also set up the system to display on the console each time an attempt is made to use the su command to gain superuser access from a remote system. This is a good way to immediately detect someone trying to gain superuser access on the system you are currently working on. See the section below for detailed instructions.

▼ How to Monitor Who Is Using the su Command

1. **Become superuser.**

2. **Edit the** /etc/default/su **file.**

3. **Uncomment the following line.**

```
SULOG=/var/adm/sulog
```

4. **After modifying the** /etc/default/su **file, use the** su **command several times and display the** /var/adm/sulog **file. You should see an entry for each time you used the** su **command.**

```
# more /var/adm/sulog
SU 12/20 16:26 + pts/0 nathan-root
SU 12/21 10:59 + pts/0 nathan-root
SU 01/12 11:11 + pts/0 root-joebob
SU 01/12 14:56 + pts/0 pmorph-root
SU 01/12 14:57 + pts/0 pmorph-root
```

▼ How to Display Superuser (root) Access Attempts to the Console

1. **Become superuser.**

2. **Edit the** `/etc/default/su` **file.**

3. **Uncomment the following line.**

```
CONSOLE=/dev/console
```

   Use the `su` command to become root, and verify that a message is printed on the system console.

Modifying a System's Abort Sequence

Use the following procedure to disable or enable a system's abort sequence. The default system behavior is that a system's abort sequence is enabled.

Some server systems have a key switch that if set in the secure position, overrides the software keyboard abort settings, so any changes you make with the following procedure may not be implemented.

▼ How to Disable or Enable a System's Abort Sequence

1. **Become superuser.**

2. **Select one of the following to disable or enable a system's abort sequence:**

   a. **Remove the pound sign (#) from the following line in the** `/etc/default/kbd` **file to disable a system's abort sequence:**

```
#KEYBOARD_ABORT=disable
```

   b. **Add the pound sign (#) to the following line in the** `/etc/default/kbd` **file to enable a system's abort sequence:**

```
KEYBOARD_ABORT=disable
```

3. **Update the keyboard defaults.**

```
# kbd -i
```

CHAPTER 19

Role-Based Access Control

This chapter describes Role-Based Access Control, a new security feature in the Solaris 8 release.

- "Extended User Attributes Database (`user_attr`)" on page 337
- "Authorizations" on page 339
- "Execution Profiles" on page 341
- "Execution Attributes" on page 343
- "How to Assume Role-Based Access Control" on page 346
- "Tools for Managing Role-Based Access Control" on page 347

Overview of Role-Based Access Control

Role-based access control (RBAC) is an alternative to the all-or-nothing security model of traditional superuser-based systems. The problem with the traditional model is not just that superuser is so powerful but that other users are not powerful enough to fix their own problems. RBAC provides the ability to package superuser privileges for assignment to user accounts.

With RBAC, you can give users the ability to solve their own problems by assigning them packages of the appropriate privileges. Superuser's capabilities can be diminished by dividing those capabilities into several packages and assigning them separately to individuals sharing administrative responsibilities.

RBAC thus enables separation of powers, controlled delegation of privileged operations to other users, and a variable degree of access control.

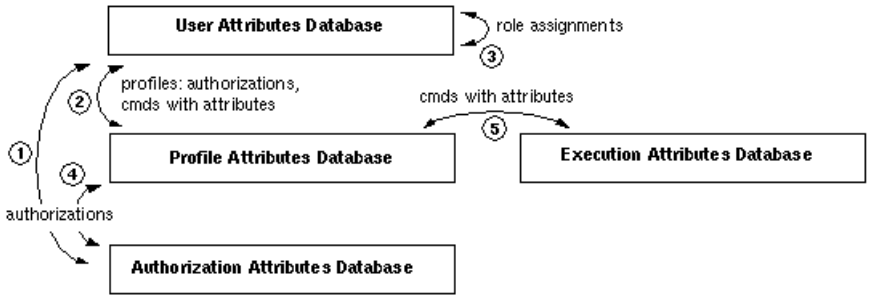
RBAC includes these features:

- Authorization - A right that is used to grant access to a restricted function
- Execution profile (or simply profile) - A bundling mechanism for grouping authorizations and commands with special attributes; for example, user and group IDs
- Role - A special type of user account intended for performing a set of administrative tasks

RBAC relies on four databases to provide users access to privileged operations:

- `user_attr` (extended user attributes database) - Associates users and roles with authorizations and execution profiles
- `auth_attr` (authorization attributes database) - Defines authorizations and their attributes and identifies the associated help file
- `prof_attr` (execution profile attributes database) - Defines profiles, lists the profile's assigned authorizations, and identifies the associated help file
- `exec_attr` (profile execution attributes database) - Defines the privileged operations assigned to a profile

The following figure illustrates how RBAC works. Databases are shown in boxes while the arrows indicate relationships between databases. The entities assigned in the relationships appear next to the arrows.



You can assign authorizations (1) and profiles (2) to users in the `user_attr` database; this is direct assignment of privileged operations. You can also assign the user to a role (3), to give the user access to any privileged operations associated with that role. Profiles are defined in the `prof_attr` database and can include authorizations (4) defined in `auth_attr` and commands with attributes (5) defined for that profile in `exec_attr`.

Commands that are assigned to profiles are run in special shells called *profile shells.* The profile shells are `pfsh`, `pfcsh`, and `pfksh`, and they correspond to Bourne shell (`sh`), C shell (`csh`), and Korn shell (`ksh`) respectively.

Extended User Attributes Database (user_attr)

The /etc/user_attr database supplements the passwd and shadow databases. It contains extended user attributes such as authorizations and execution profiles. It also allows roles to be assigned to a user.

A role is a special type of user account that is intended for performing a set of administrative tasks. It is like a normal user account in most respects except that users can gain access to it only through the su command; it is not accessible for normal logins, for example, through the CDE login window. From a role account, a user can access commands with special attributes, typically root user ID, that are not available to users in normal accounts.

The fields in the user_attr database are separated by colons:

```
user:qualifier:res1:res2:attr
```

The fields are described in the following table.

Field Name	Description
user	The name of the user as specified in the passwd(4) database.
qualifier	Reserved for future use.
res1	Reserved for future use.

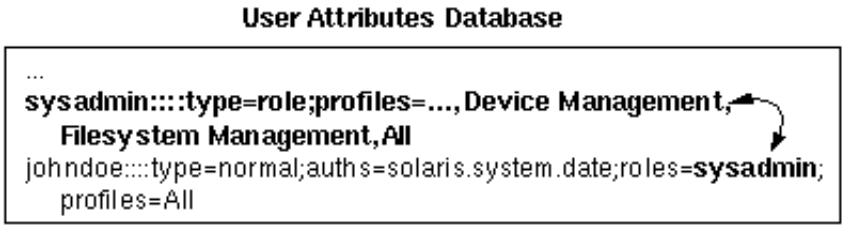
Field Name	Description
res2	Reserved for future use.
attr	An optional list of semicolon-separated (;) key-value pairs that describe the security attributes to be applied when the user runs commands. There are four valid keys: auths, profiles, roles, and type. ■ auths specifies a comma-separated list of authorization names chosen from names defined in the auth_attr(4) database. Authorization names may include the asterisk (*) character as a wildcard. For example, solaris.device.* means all of the Solaris device authorizations. ■ profiles contains an ordered, comma-separated list of profile names chosen from prof_attr(4). A profile determines which commands a user can execute and with which command attributes. At minimum each user in user_attr should have the All profile, which makes all commands available but without any attributes. The order of profiles is important; it works similarly to UNIX search paths. The first profile in the list that contains the command to be executed defines which (if any) attributes are to be applied to the command. ■ roles can be assigned to the user using a comma-separated list of role names. Note that roles are defined in the same user_attr database. They are indicated by setting the type value to role. Roles cannot be assigned to other roles. ■ type can be set to normal, if this account is for a normal user, or to role, if this account is for a role. A role is assumed by a normal user after the user has logged in.

A user_attr database with typical values is shown in the following example.

User Attributes Database

```
root::::type=normal;auths=solaris.*,solaris.grant;profiles=All
sysadmin::::type=role;profiles=...,Device Management,Filesystem
    Management,All
johndoe::::type=normal;auths=solaris.system.date;roles=sysadmin;
    profiles=All
```

A typical role assignment is illustrated in the following user_attr database. In this example, the sysadmin role has been assigned to the user johndoe. When assuming the sysadmin role, johndoe has access to such profiles as Device Management, Filesystem Management, and the All profile.



Authorizations

An authorization is a user right that grants access to a restricted function. It is a unique string that identifies what is being authorized as well as who created the authorization.

Authorizations are checked by certain privileged programs to determine whether users can execute restricted functionality. For example, the `solaris.jobs.admin` authorization is required for one user to edit another user's `crontab` file.

All authorizations are stored in the `auth_attr` database. Authorizations may be assigned directly to users (or roles) in which case they are entered in the `user_attr` database. Authorizations can also be assigned to execution profiles which in turn are assigned to users.

The fields in the `auth_attr` database are separated by colons:

```
authname:res1:res2:short_desc:long_desc:attr
```

The fields are described in the following table.

Field Name	Description
`authname`	A unique character string used to identify the authorization in the format *prefix*.[*suffix*]. Authorizations for the Solaris operating environment use `solaris` as a prefix. All other authorizations should use a prefix that begins with the reverse-order Internet domain name of the organization that creates the authorization (for example, `com.xyzcompany`). The suffix indicates what is being authorized, typically the functional area and operation. When there is no suffix (that is, the `authname` consists of a prefix and functional area and ends with a period), the `authname` serves as a heading for use by applications in their GUIs rather than as an authorization. The `authname solaris.printmgr.` is an example of a heading. When `authname` ends with the word grant, the `authname` serves as a grant authorization and lets the user delegate related authorizations (that is, authorizations with the same prefix and functional area) to other users. The `authname solaris.printmgr.grant` is an example of a grant authorization; it gives the user the right to delegate such authorizations as `solaris.printmgr.admin` and `solaris.printmgr.nobanner` to other users.
`res1`	Reserved for future use.
`res2`	Reserved for future use.
`short_desc`	A terse name for the authorization suitable for displaying in user interfaces, such as in a scrolling list in a GUI.
`long_desc`	A long description. This field identifies the purpose of the authorization, the applications in which it is used, and the type of user interested in using it. The long description can be displayed in the help text of an application.
`attr`	An optional list of semicolon-separated (;) key-value pairs that describe the attributes of an authorization. Zero or more keys may be specified. The keyword help identifies a help file in HTML. Help files can be accessed from the `index.html` file in the `/usr/lib/help/auths/locale/C`directory.

An `auth_attr` database with some typical values is shown in the following example.

Authorization Attributes Database

```
solaris.*:::Primary Administrator::help=PriAdmin.html
solaris..grant:::Grant All Rights::help=PriAdmin.html
...
solaris.device.:::Device Allocation::help=DevAllocHeader.html
solaris.device.allocate:::Allocate Device::help=DevAllocate.html
solaris.device.config:::Configure Device Attributes::help=DevConfig.html
solaris.device.grant::Delegate Device Administration::help=DevGrant.html
solaris.device.revoke:::Revoke or Reclaim Device::help=DevRevoke.html
...
```

The relationship between the `auth_attr` and the `user_attr` databases is illustrated in the following example. The `solaris.system.date` authorization, which is defined in the `auth_attr` database, is assigned to the user `johndoe` in the `user_attr` database.

User Attributes Database

```
root::::type=normal;auths=solaris.*,solaris.grant;profiles=All
...
johndoe::::type=normal;auths=solaris.system.date;roles=sysadmin;
    profiles=All
...
```

Authorization Attributes Database

```
solaris.*:::Primary Administrator::help=PriAdmin.html
...
solaris.system.date:::Set Date & Time::help=SysDate.html
...
```

Execution Profiles

An execution profile is a bundling mechanism for grouping authorizations and commands with special attributes, and assigning them to users or roles. The special attributes include real and effective UIDs and GIDs. The most common attribute is setting the real or effective UID to root. The definitions of execution profiles are stored in the `prof_attr` database.

The fields in the `prof_attr` database are separated by colons:

```
profname:res1:res2:desc:attr
```

The fields are described in the following table.

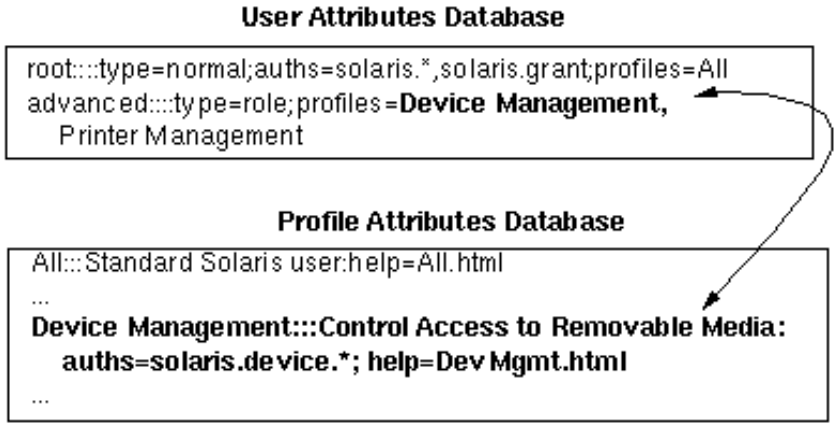
Field Name	Description
profname	The name of the profile. Profile names are case-sensitive.
res1	Reserved for future use.
res2	Reserved for future use.
desc	A long description. This field should explain the purpose of the profile, including what type of user would be interested in using it. The long description should be suitable for displaying in the help text of an application.
attr	An optional list of key-value pairs separated by semicolons (;) that describe the security attributes to apply to the object upon execution. Zero or more keys may be specified. There are two valid keys, help and auths. The keyword help identifies a help file in HTML. Help files can be accessed from the index.html file in the /usr/lib/help/auths/locale/C directory. auths specifies a comma-separated list of authorization names chosen from those names defined in the auth_attr(4) database. Authorization names may be specified using the asterisk (*) character as a wildcard.

A prof_attr database with some typical values is shown in the following example.

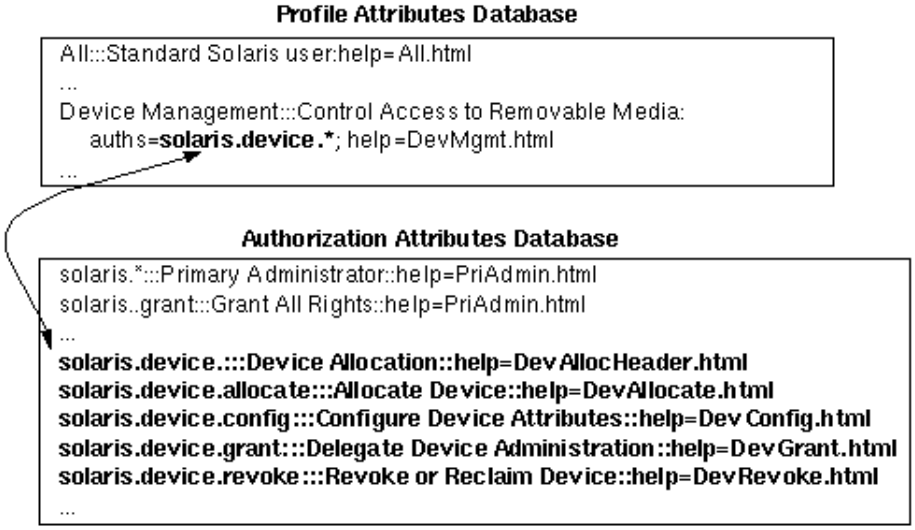
Profile Attributes Database

```
All:::Standard Solaris user:help=All.html
...
Printer Management:::Manage print jobs: help=Printmgt.html
Device Management:::Control Access to Removable Media:
   auths=solaris.device.*; help=DevMgmt.html
...
```

The relationship between the prof_attr and the user_attr databases is illustrated in the following example. The Device Management profile, which is defined in the prof_attr database, is assigned to the sysadmin role in the user_attr database.



The relationship between the `prof_attr` and the `auth_attr` databases is illustrated in the following example. The Device Management profile is defined in the `prof_attr` database as having all authorizations beginning with the `solaris.device.` string assigned to it. These authorizations are defined in the `auth_attr` database.



Execution Attributes

An execution attribute associated with a profile is a command (with any special security attributes) that can be run by those users or roles to whom the profile is assigned. Special security attributes refer to such attributes as UID, EUID, GID, EGID that can be added to a process when the command is run.

The definitions of the execution attributes are stored in the `exec_attr` database.

The fields in the `exec_attr` database are separated by colons:

```
name:policy:type:res1:res2:id:attr
```

The fields are described in the following table.

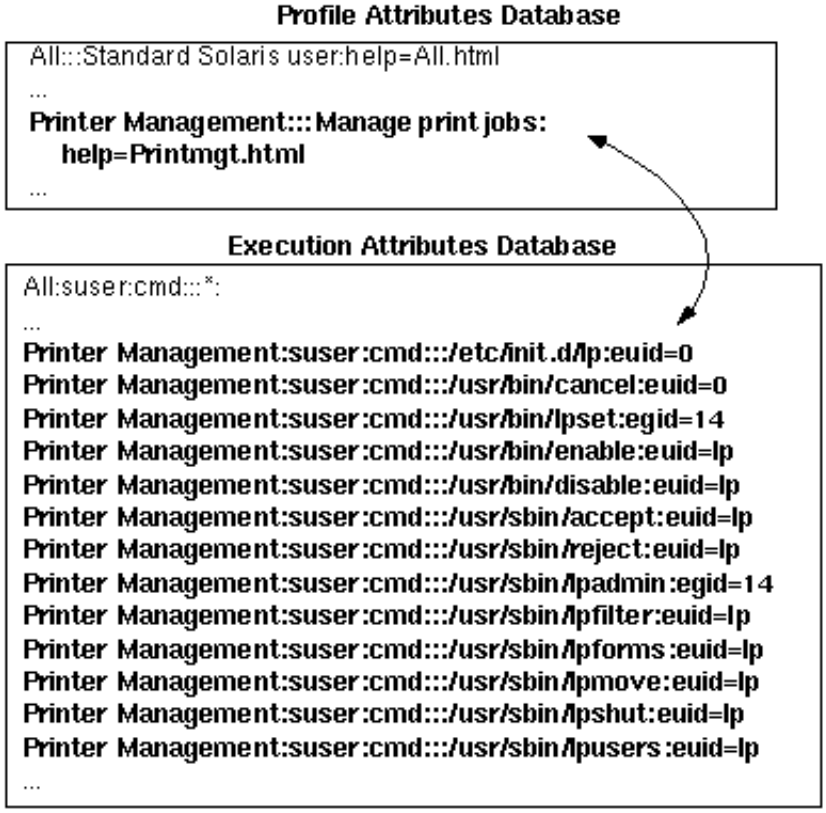
Field Name	Description
`name`	The name of the profile. Profile names are case-sensitive.
`policy`	The security policy associated with this entry. Currently, `suser` (the superuser policy model) is the only valid policy entry.
`type`	The type of entity whose attributes are specified. Currently, the only valid type is `cmd` (command).
`res1`	Reserved for future use.
`res2`	Reserved for future use.
`id`	A string identifying the entity; the asterisk wild card can be used. Commands should have the full path or a path with a wild card. To specify arguments, write a script with the arguments and point the `id` to the script.
`attr`	An optional list of semicolon (;) separated key-value pairs that describe the security attributes to apply to the entity upon execution. Zero or more keys may be specified. The list of valid key words depends on the policy being enforced. There are four valid keys: `euid`, `uid`, `egid`, and `gid`. `euid` and `uid` contain a single user name or a numeric user ID. Commands designated with `euid` run with the effective UID indicated, which is similar to setting the `setuid` bit on an executable file. Commands designated with `uid` run with both the real and effective UIDs. `egid` and `gid` contain a single group name or numeric group ID. Commands designated with `egid` run with the effective GID indicated, which is similar to setting the `setgid` bit on an executable file. Commands designated with `gid` run with both the real and effective GIDs.

An `exec_attr` database with some typical values is shown in the following example.

Execution Attributes Database

```
All:suser:cmd:::*:
...
Printer Management:suser:cmd:::/usr/lib/lp/lpsched:euid=0
Printer Management:suser:cmd:::/usr/lib/lp/lpshut:euid=0
Printer Management:suser:cmd:::/usr/lib/lp/lpmove:euid=0
Printer Management:suser:cmd:::/bin/lp:euid=0
Printer Management:suser:cmd:::/bin/lpadmin:euid=0
Printer Management:suser:cmd:::/usr/sbin/lpadmin:euid=0
Printer Management:suser:cmd:::/usr/bin/enable:euid=0
Printer Management:suser:cmd:::/usr/bin/disable:euid=0
Printer Management:suser:cmd:::/usr/sbin/accept:euid=0
Printer Management:suser:cmd:::/usr/sbin/reject:euid=0
Printer Management:suser:cmd:::/usr/sbin/lpsystem:euid=0
...
```

The relationship between the `exec_attr` and the `prof_attr` databases is illustrated in the following example. The Printer Management profile is defined in the `prof_attr` database. It has 13 execution attributes with the appropriate security attributes assigned to it in the `exec_attr` database.



▼ How to Assume Role-Based Access Control

To assume a role, use the `su` command. You cannot log in to a role. For example:

```
%su my-role
Password: my-role-password
#
```

To use commands in the profile, simply type into a shell. For example:

```
# lpadmin -p myprinter options
```

The `lpadmin` command is executed with any process attributes, special UIDs or GIDs, that have been assigned to the `lpadmin` command in profiles for the role assumed.

Tools for Managing Role-Based Access Control

In addition to editing the databases directly, the following tools are available for managing with role-based access control.

Command	Description
`auths`(1)	Display authorizations for a user.
`makedbm`(1M)	Make a `dbm` file.
`nscd`(1M)	Name service cache daemon, useful for caching the `user_attr`, `prof_attr`, and `exec_attr` databases.
`pam_roles`(5)	Role account management module for PAM. Checks for the authorization to assume role.
`pfexec`(1)	Profile shells, used by profile shells to execute commands with attributes specified in the `exec_attr` database.
`policy.conf`(4)	Configuration file for security policy. Lists granted authorizations.
`profiles`(1)	Display profiles for a specified user.
`roles`(1)	Display roles granted to a user.
`roleadd`(1M)	Add a role account on the system.
`roledel`(1M)	Delete a role's account from the system.
`rolemod`(1M)	Modify a role's account information on the system.
`useradd`(1M)	Add a user account on the system. The `–P` option assigns a role to a user's account.

Command	Description
`userdel`(1M)	Delete a user's login from the system.
`usermod`(1M)	Modify a user's account information on the system.

CHAPTER 20

Using Authentication Services (Tasks)

The first section of this chapter provides information about the Diffie-Hellman authentication mechanism that may be used with Secure RPC. The second section covers the Pluggable Authentication Module (PAM) framework. PAM provides a method to "plug-in" authentication services and provides support for multiple authentication services.

This is a list of the step-by-step instructions in this chapter.

- "How to Restart the Keyserver" on page 354
- "How to Set Up NIS+ Credentials for Diffie-Hellman Authentication" on page 355
- "How to Set Up NIS Credentials With Diffie-Hellman Authentication" on page 357
- "How to Share and Mount Files With Diffie-Hellman Authentication" on page 358
- "How to Add a PAM Module" on page 370
- "How to Prevent Unauthorized Access From Remote Systems With PAM" on page 370
- "How to Initiate PAM Error Reporting" on page 370

Overview of Secure RPC

Secure RPC is a method of authentication that authenticates both the host and the user making a request. Secure RPC uses Diffie-Hellman. This authentication mechanisms use DES encryption. Applications that use Secure RPC include NFS and the NIS+ name service.

NFS Services and Secure RPC

The NFS software enables several hosts to share files over the network. Under the NFS system, a server holds the data and resources for several clients. The clients have access to the file systems that the server shares with the clients. Users logged in to the client machine can access the file systems by mounting them from the server. To the user on the client machine, it appears as if the files are local to the client. One of the most common uses of the NFS environment is to allow systems to be installed in offices, while keeping all user files in a central location. Some features of the NFS system, such as the `mount –nosuid` option, can be used to prohibit the opening of devices as well as file systems by unauthorized users.

The NFS environment uses Secure RPC to authenticate users who make requests over the network. This is known as Secure NFS. The authentication mechanism, `AUTH_DH`, uses DES encryption with Diffie-Hellman authentication to ensure authorized access. The `AUTH_DH` mechanism has also been called `AUTH_DES`.

The *System Administration Guide, Volume 3* describes how to set up and administer Secure NFS. Setting up the NIS+ tables and entering names in the `cred` table are discussed in *Solaris Naming Administration Guide*. See “Implementation of Diffie-Hellman Authentication” on page 351 for an outline of the steps involved in RPC authentication.

DES Encryption

The Data Encryption Standard (DES) encryption functions use a 56-bit key to encrypt data. If two credential users (or principals) know the same DES key, they can communicate in private, using the key to encipher and decipher text. DES is a relatively fast encryption mechanism. A DES chip makes the encryption even faster; but if the chip is not present, a software implementation is substituted.

The risk of using just the DES key is that an intruder can collect enough cipher-text messages encrypted with the same key to be able to discover the key and decipher the messages. For this reason, security systems such as Secure NFS change the keys frequently.

Kerberos Authentication

Kerberos is an authentication system developed at MIT. Encryption in Kerberos is based on DES. Kerberos V4 support is no longer supplied as part of Secure RPC, but a client-side implementation of Kerberos V5, which uses RPCSEC_GSS, is included with the Solaris 8 release. For more information see Chapter 21.

Diffie-Hellman Authentication

The Diffie-Hellman method of authenticating a user is non-trivial for an intruder to crack. The client and the server each has its own private key (sometimes called a secret key) which they use together with the public key to devise a common key. They use the common key to communicate with each other, using an agreed-upon encryption/decryption function (such as DES). This method was identified as DES authentication in previous Solaris releases.

Authentication is based on the ability of the sending system to use the common key to encrypt the current time, which the receiving system can decrypt and check against its current time. Make sure you synchronize the time on the client and the server.

The public and private keys are stored in an NIS or NIS+ database. NIS stores the keys in the `publickey` map, and NIS+ stores the keys in the `cred` table. These files contain the public key and the private key for all potential users.

The system administrator is responsible for setting up NIS or NIS+ tables and generating a public key and a private key for each user. The private key is stored encrypted with the user's password. This makes the private key known only to the user.

Implementation of Diffie-Hellman Authentication

This section describes the series of transactions in a client-server session using DH authorization (`AUTH_DH`).

Generating the Public and Secret Keys

Sometime prior to a transaction, the administrator runs either the `newkey` or `nisaddcred` commands that generates a public key and a secret key. (Each user has a unique public key and secret key.) The public key is stored in a public database; the secret key is stored in encrypted form in the same database. To change the key pair, use the `chkey` command.

Running the `keylogin` *Command*

Normally, the login password is identical to the secure RPC password. In this case, a `keylogin` is not required. If the passwords are different, the users have to log in, and then do a `keylogin` explicitly.

The `keylogin` program prompts the user for a secure RPC password and uses the password to decrypt the secret key. The `keylogin` program then passes the decrypted secret key to a program called the keyserver. (The keyserver is an RPC service with a local instance on every computer.) The keyserver saves the decrypted secret key and waits for the user to initiate a secure RPC transaction with a server.

If the passwords are the same, the login process passes the secret key to the keyserver. If the passwords are required to be different and the user must always run `keylogin`, then the `keylogin` program may be included in the user's environment configuration file, such as `~/.login`, `~/.cshrc`, or `~/.profile`, so that it runs automatically whenever the user logs in.

Generating the Conversation Key

When the user initiates a transaction with a server:

1. The keyserver randomly generates a conversation key.
2. The kernel uses the conversation key to encrypt the client's time stamp (among other things).
3. The keyserver looks up the server's public key in the public-key database (see the `publickey`(4) man page).
4. The keyserver uses the client's secret key and the server's public key to create a common key.
5. The keyserver encrypts the conversation key with the common key.

First Contact With the Server

The transmission including the encrypted time stamp and the encrypted conversation key is then sent to the server. The transmission includes a credential and a verifier. The credential contains three components:

- The client's net name
- The conversation key, encrypted with the common key
- A "window," encrypted with the conversation key

The window is the difference the client says should be allowed between the server's clock and the client's time stamp. If the difference between the server's clock and the time stamp is greater than the window, the server would reject the client's request. Under normal circumstances this will not happen, because the client first synchronizes with the server before starting the RPC session.

The client's verifier contains:

- The encrypted time stamp
- An encrypted verifier of the specified window, decremented by 1

The window verifier is needed in case somebody wants to impersonate a user and writes a program that, instead of filling in the encrypted fields of the credential and verifier, just stuffs in random bits. The server will decrypt the conversation key into some random key and use it to try to decrypt the window and the time stamp. The result will be random numbers. After a few thousand trials, however, there is a good chance that the random window/time stamp pair will pass the authentication system. The window verifier makes guessing the right credential much more difficult.

Decrypting the Conversation Key

When the server receives the transmission from the client:

1. The keyserver local to the server looks up the client's public key in the publickey database.
2. The keyserver uses the client's public key and the server's secret key to deduce the common key—the same common key computed by the client. (Only the server and the client can calculate the common key because doing so requires knowing one secret key or the other.)
3. The kernel uses the common key to decrypt the conversation key.
4. The kernel calls the keyserver to decrypt the client's time stamp with the decrypted conversation key.

Storing Information on the Server

After the server decrypts the client's time stamp, it stores four items of information in a credential table:

- The client's computer name
- The conversation key
- The window
- The client's time stamp

The server stores the first three items for future use. It stores the time stamp to protect against replays. The server accepts only time stamps that are chronologically greater than the last one seen, so any replayed transactions are guaranteed to be rejected.

Note - Implicit in these procedures is the name of the caller, who must be authenticated in some manner. The keyserver cannot use DES authentication to do this because it would create a deadlock. To solve this problem, the keyserver stores the secret keys by UID and grants requests only to local root processes.

Verifier Returned to the Client

The server returns a verifier to the client, which includes:

- The index ID, which the server records in its credential cache
- The client's time stamp minus 1, encrypted by conversation key

The reason for subtracting 1 from the time stamp is to ensure that the time stamp is invalid and cannot be reused as a client verifier.

Client Authenticates the Server

The client receives the verifier and authenticates the server. The client knows that only the server could have sent the verifier because only the server knows what time stamp the client sent.

Additional Transactions

With every transaction after the first, the client returns the index ID to the server in its second transaction and sends another encrypted time stamp. The server sends back the client's time stamp minus 1, encrypted by the conversation key.

Administering Diffie-Hellman Authentication

A system administrator can implement policies that help secure the network. The level of security required will differ with each site. This section provides instructions for some tasks associated with network security.

▼ How to Restart the Keyserver

1. **Become superuser.**

2. **Verify whether the** `keyserv` **daemon (the keyserver) is running.**

```
# ps -ef | grep keyserv
root  100      1   16   Apr 11 ?      0:00 /usr/sbin/keyserv
root 2215   2211    5  09:57:28 pts/0 0:00 grep keyserv
```

3. **Start the keyserver if it isn't running.**

```
# /usr/sbin/keyserv
```

▼ How to Set Up NIS+ Credentials for Diffie-Hellman Authentication

For detailed description of NIS+ security, see *Solaris Naming Administration Guide.*

To set up a new key for root on an NIS+ client:

1. **Become superuser.**

2. **Edit the** `/etc/nsswitch.conf` **file and add the following line:**

```
publickey: nisplus
```

3. **Initialize the NIS+ client.**

```
# nisinit -cH hostname
```

   hostname is the name of a trusted NIS+ server that contains an entry in its tables for the client machine.

4. **Add the client to the** `cred` **table by typing the following commands.**

```
# nisaddcred local
# nisaddcred des
```

5. **Verify the setup by using the** `keylogin` **command.**

   If you are prompted for a password, the procedure has succeeded.

Example—Setting Up a New Key for root on a NIS+ Client

The following example uses the host `pluto` to set up `earth` as an NIS+ client. You can ignore the warnings. The `keylogin` command is accepted, verifying that `earth` is correctly set up as a secure NIS+ client.

```
# nisinit -cH pluto
NIS Server/Client setup utility.
This machine is in the North.Abc.COM. directory.
Setting up NIS+ client ...
All done.
# nisaddcred local
```

```
# nisaddcred des
DES principal name : unix.earth@North.Abc.COM
Adding new key for unix.earth@North.Abc.Com (earth.North.Abc.COM.)

Network password: xxx <Press Return>
Warning, password differs from login password.
Retype password: xxx <Press Return>

# keylogin
Password:
#
```

To set up a new key for an NIS+ user:

1. **Add the user to the** `cred` **table on the root master server by typing the following command**:

   ```
   # nisaddcred -p unix.UID@domainname -P username.domainname. des
   ```

   Note that, in this case, the *username-domainname* must end with a dot (.)

2. **Verify the setup by logging in as the client and typing the** `keylogin` **command.**

Example—Setting Up a New Key for an NIS+ User

The following example gives DES security authorization to user `george`.

```
# nisaddcred -p unix.1234@North.Abc.com -P george.North.Abc.COM. des
DES principal name : unix.1234@North.Abc.COM
Adding new key for unix.1234@North.Abc.COM (george.North.Abc.COM.)

Password:
Retype password:

# rlogin rootmaster -l george
# keylogin
Password:
#
```

▼ How to Set Up NIS Credentials With Diffie-Hellman Authentication

To create a new key for superuser on a client:

1. **Become superuser on the client.**

2. **Edit the** `/etc/nsswitch.conf` **file and add the following line:**

```
publickey: nis
```

3. **Create a new key pair by using the** `newkey` **command.**

```
# newkey -h hostname
```

hostname is the name of the client.

Example—Setting Up an NIS+ Client to Use Diffie-Hellman Security

The following example sets up `earth` as a secure NIS client.

```
# newkey -h earth
Adding new key for unix.earth@North.Abc.COM
New Password:
Retype password:
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

To create a new key for a user:

1. **Log in to the server as superuser.**

   Only the system administrator, logged in to the NIS+ server, can generate a new key for a user.

2. **Create a new key for a user.**

```
# newkey -u username
```

username is the name of the user. The system prompts for a password. The system administrator can type a generic password. The private key is stored encrypted with the generic password.

```
# newkey -u george
Adding new key for unix.12345@Abc.North.Acme.COM
New Password:
Retype password:
Please wait for the database to get updated...
Your new key has been successfully stored away.
#
```

3. **Tell the user to log in and type the** `chkey -p` **command.**

   This allows the user to re-encrypt their private key with a password known only to the user.

```
earth% chkey -p
Updating nis publickey database.
Reencrypting key for unix.12345@Abc.North.Acme.COM
Please enter the Secure-RPC password for george:
Please enter the login password for george:
Sending key change request to pluto...
#
```

Note - The `chkey` command can be used to create a new key-pair for a user.

▼ How to Share and Mount Files With Diffie-Hellman Authentication

Prerequisite

The Diffie-Hellman `publickey` authentication must be enabled on the network. See "How to Set Up NIS+ Credentials for Diffie-Hellman Authentication" on page 355 and "How to Set Up NIS Credentials With Diffie-Hellman Authentication" on page 357.

To share a file system with Diffie-Hellman authentication:

1. **Become superuser.**

2. **Share the file system with Diffie-Hellman authentication.**

```
# share -F nfs -o sec=dh /filesystem
```

To mount a file system with Diffie-Hellman authentication:

1. **Become superuser.**
2. **Mount the file system with Diffie-Hellman authentication.**

```
# mount -F nfs -o sec=dh server:resource    mountpoint
```

The `-o sec=dh` option mounts the file system with `AUTH_DH` authentication.

Introduction to PAM

The Pluggable Authentication Module (PAM) framework lets you "plug in" new authentication technologies without changing system entry services such as `login`, `ftp`, `telnet`, and so on. You can also use PAM to integrate UNIX login with other security mechanisms like DCE or Kerberos. Mechanisms for account, session, and password management can also be "plugged in" using this framework.

Benefits of Using PAM

The PAM framework allows a system administrator to choose any combination of system entry services (`ftp`, `login`, `telnet`, or `rsh`, for example) for user authentication. Some of the benefits PAM provides are:

- Flexible configuration policy
  - Per application authentication policy
  - The ability to choose a default authentication mechanism
  - Multiple passwords on high-security systems
- Ease of use for the end user
  - No retyping of passwords if they are the same for different mechanisms
  - The ability to use a single password for multiple authentication methods with the password mapping feature, even if the passwords associated with each authentication method are different

- The ability to prompt the user for passwords for multiple authentication methods without having the user enter multiple commands

- The ability to pass optional parameters to the user authentication services

Overview of PAM

PAM employs run-time pluggable modules to provide authentication for system entry services. These modules are broken into four different types based on their function: authentication, account management, session management, and password management. A stacking feature is provided to let you authenticate users through multiple services, as well as a password-mapping feature to not require that users remember multiple passwords.

PAM Module Types

It is important to understand the PAM module types because the module type defines the interface to the module. These are the four types of run-time PAM modules:

- The *authentication modules* provide authentication for the users and allow for credentials to be set, refreshed, or destroyed. They provide a valuable administration tool for user identification.
- The *account modules* check for password aging, account expiration, and access hour restrictions. After the user is identified through the authentication modules, the account modules determine if the user should be given access.
- The *session modules* manage the opening and closing of an authentication session. They can log activity or provide for clean-up after the session is over.
- The *password modules* allow for changes to the actual password.

Stacking Feature

The PAM framework provides a method for authenticating users with multiple services using *stacking*. Depending on the configuration, the user can be prompted for passwords for each authentication method. The order in which the authentication services are used is determined through the PAM configuration file.

Password-Mapping Feature

The stacking method can require that a user remember several passwords. With the *password-mapping* feature, the primary password is used to decrypt the other passwords, so the user doesn't need to remember or enter multiple passwords. The other option is to synchronize the passwords across each authentication mechanism. Note that this could increase the security risk, since the security of each mechanism is limited by the least secure password method used in the stack.

PAM Functionality

The PAM software consists of a library, several modules, and a configuration file. New versions of several system entry commands or daemons which take advantage of the PAM interfaces are also included.

The figure below illustrates the relationship between the applications, the PAM library, the `pam.conf` file, and the PAM modules.

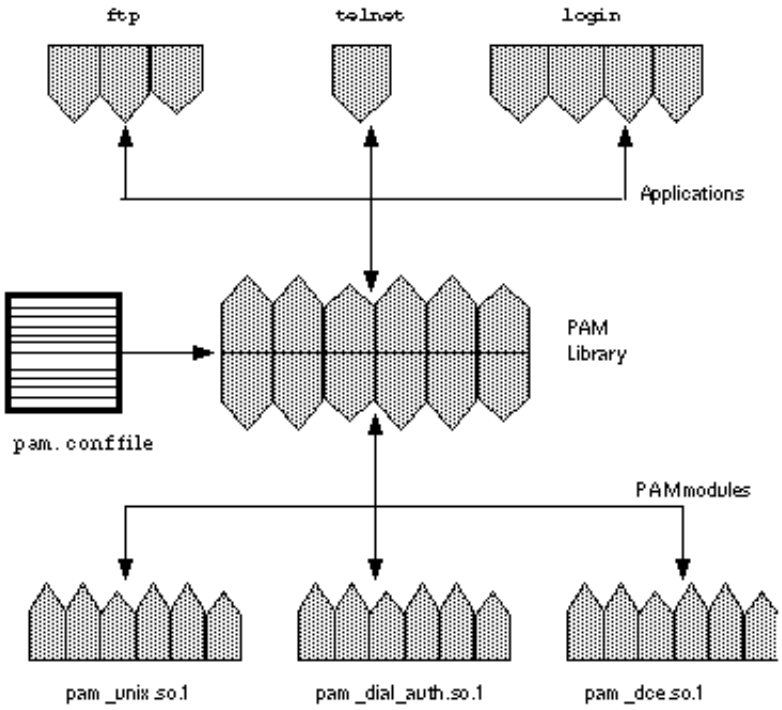


Figure 20–1 How PAM Works

The applications (`ftp`, `telnet`, and `login`) use the PAM library to access the appropriate module. The `pam.conf` file defines which modules to use, and in what order they are to be used with each application. Responses from the modules are passed back through the library to the application.

The following sections describe this relationship.

PAM Library

The PAM library, `/usr/lib/libpam`, provides the framework to load the appropriate modules and manage the stacking process. It provides a generic structure to which all of the modules can plug in.

PAM Modules

Each PAM module implements a specific mechanism. When setting up PAM authentication, you need to specify both the module and the module type, which

defines what the module will do. More than one module type (auth, account, session, or password) may be associated with each module.

The following list describes each of the PAM modules.

- The `pam_unix` module, `/usr/lib/security/pam_unix.so.1`, provides support for authentication, account management, session management, and password management. Any of the four module type definitions can be used with this module. It uses UNIX passwords for authentication. In the Solaris environment, the selection of appropriate name services to get password records is controlled through the `/etc/nsswitch.conf` file. See `pam_unix`(5) for more information.
- The `dial_auth` module, `/usr/lib/security/pam_dial_auth.so.1`, can only be used for authentication. It uses data stored in the `/etc/dialups` and `/etc/d_passwd` files for authentication. This is mainly used by `login`. See `pam_dial_auth`(5) for more information.
- The `rhosts_auth` module, `/usr/lib/security/pam_rhosts_auth.so.1`, can also only be used for authentication. It uses data stored in the `~/.rhosts` and `/etc/host.equiv` files through `ruserok( )`. This is mainly used by the `rlogin` and `rsh` commands. See `pam_rhosts_auth`(5) for more information.
- The `krb5` module, `/usr/lib/security/pam_krb5_auth.so.1`, provides support for authentication, account management, session management, and password management. Kerberos credentials are used for authentication.

For security reasons, these module files must be owned by root and must not be writable through `group` or `other` permissions. If the file is not owned by root, PAM will not load the module.

PAM Configuration File

The PAM configuration file, `/etc/pam.conf`, determines the authentication services to be used, and in what order they are used. This file can be edited to select authentication mechanisms for each system-entry application.

Configuration File Syntax

The PAM configuration file consists of entries with the following syntax:

```
service_name  module_type  control_flag  module_path  module_options
```

service_name	Name of the service (for example, `ftp`, `login`, `telnet`).
module_type	Module type for the service.
control_flag	Determines the continuation or failure semantics for the module.
module_path	Path to the library object that implements the service functionality.
module_options	Specific options that are passed to the service modules.

You can add comments to the `pam.conf` file by starting the line with a # (pound sign). Use white space to delimit the fields.

Note - An entry in the PAM configuration file is ignored if one of the following conditions exist: the line has less than four fields, an invalid value is given for *module_type* or *control_flag*, or the named module is not found.

Valid Service Names

The table below lists some of the valid service names, the module types that can be used with that service, and the daemon or command associated with the service name.

There are several module types that are not appropriate for each service. For example, the `password` module type is only specified to go with the `passwd` command. There is no `auth` module type associated with this command since it is not concerned with authentication.

TABLE 20–1 Valid Service Names for `/etc/pam.conf`

Service Name	Daemon or Command	Module Type
`dtlogin`	`/usr/dt/bin/dtlogin`	`auth, account, session`
`ftp`	`/usr/sbin/in.ftpd`	`auth, account, session`
`init`	`/usr/sbin/init`	`session`
`login`	`/usr/bin/login`	`auth, account, session`

TABLE 20–1 Valid Service Names for `/etc/pam.conf` *(continued)*

Service Name	Daemon or Command	Module Type
passwd	/usr/bin/passwd	password
rexd	/usr/sbin/rpc.rexd	auth
rlogin	/usr/sbin/in.rlogind	auth, account, session
rsh	/usr/sbin/in.rshd	auth, account, session
sac	/usr/lib/saf/sac	session
su	/usr/bin/su	auth, account, session
telnet	/usr/sbin/in.telnetd	auth, account, session
ttymon	/usr/lib/saf/ttymon	session
uucp	/usr/sbin/in.uucpd	auth, account, session

Control Flags

To determine continuation or failure behavior from a module during the authentication process, you must select one of four *control flags* for each entry. The control flags indicate how a successful or a failed attempt through each module is handled. Even though these flags apply to all module types, the following explanation assumes that these flags are being used for authentication modules. The control flags are as follows:

- `required` - This module must return success in order to have the overall result be successful.

  If all of the modules are labeled as `required`, then authentication through all modules must succeed for the user to be authenticated.

  If some of the modules fail, then an error value from the first failed module is reported.

  If a failure occurs for a module flagged as `required`, all modules in the stack are still tried but failure is returned.

If none of the modules are flagged as `required`, then at least one of the entries for that service must succeed for the user to be authenticated.

- `requisite` - This module must return success for additional authentication to occur.

  If a failure occurs for a module flagged as `requisite`, an error is immediately returned to the application and no additional authentication is done. If the stack does not include prior modules labeled as `required` that failed, then the error from this module is returned. If a earlier module labeled as `required` has failed, the error message from the `required` module is returned.

- `optional` - If this module fails, the overall result can be successful if another module in this stack returns success.

  The `optional` flag should be used when one success in the stack is enough for a user to be authenticated. This flag should only be used if it is not important for this particular mechanism to succeed.

  If your users need to have permission associated with a specific mechanism to get their work done, then you should not label it as `optional`.

- `sufficient` - If this module is successful, skip the remaining modules in the stack, even if they are labeled as `required`.

  The `sufficient` flag indicates that one successful authentication will be enough for the user to be granted access.

  More information about these flags is provided in the section below, which describes the default `/etc/pam.conf` file.

Generic `pam.conf` File

The following is an example of a generic `pam.conf` file:

```
# PAM configuration
# Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth required /usr/lib/security/pam_dial_auth.so.1
rlogin auth sufficient /usr/lib/security/pam_rhost_auth.so.1
rlogin auth required /usr/lib/security/pam_unix.so.1
dtlogin auth required /usr/lib/security/pam_unix.so.1
telnet auth required /usr/lib/security/pam_unix.so.1
su auth required /usr/lib/security/pam_unix.so.1
ftp auth required /usr/lib/security/pam_unix.so.1
uucp auth required /usr/lib/security/pam_unix.so.1
rsh auth required /usr/lib/security/pam_rhost_auth.so.1
OTHER auth required /usr/lib/security/pam_unix.so.1
#
# Account management
#
```

(continued)

```
login account required /usr/lib/security/pam_unix.so.1
rlogin account required /usr/lib/security/pam_unix.so.1
dtlogin account required /usr/lib/security/pam_unix.so.1
telnet account required /usr/lib/security/pam_unix.so.1
ftp account required /usr/lib/security/pam_unix.so.1
OTHER account required /usr/lib/security/pam_unix.so.1
#
# Session management
#
login session required /usr/lib/security/pam_unix.so.1
rlogin session required /usr/lib/security/pam_unix.so.1
dtlogin session required /usr/lib/security/pam_unix.so.1
telnet session required /usr/lib/security/pam_unix.so.1
uucp session required /usr/lib/security/pam_unix.so.1
OTHER session required /usr/lib/security/pam_unix.so.1
#
# Password management
#
passwd password required /usr/lib/security/pam_unix.so.1
OTHER password required /usr/lib/security/pam_unix.so.1
```

This generic `pam.conf` file specifies:

1. When running `login`, authentication must succeed for both the `pam_unix` and the `pam_dial_auth` modules.
2. For `rlogin`, authentication through the `pam_unix` module must succeed, if authentication through `pam_rhost_auth` fails.
3. The `sufficient` control flag indicates that for `rlogin` the successful authentication provided by the `pam_rhost_auth` module is sufficient and the next entry will be ignored.
4. Most of the other commands requiring authentication require successful authentication through the `pam_unix` module.
5. Authentication for `rsh` must succeed through the `pam_rhost_auth` module.

The `OTHER` service name allows a default to be set for any other commands requiring authentication that are not included in the file. The `OTHER` option makes it easier to administer the file, since many commands that are using the same module can be covered using only one entry. Also, the `OTHER` service name, when used as a "catch-all," can ensure that each access is covered by one module. By convention, the `OTHER` entry is included at the bottom of the section for each module type.

The rest of the entries in the file control the account, session, and password management.

With the use of the default service name, `OTHER`, the generic PAM configuration file is simplified to:

```
# PAM configuration
#
# Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth required /usr/lib/scurty/pam_dial_auth.so.1
rlogin auth sufficient /usr/lib/security/pam_unix.so.1
rlogin auth required /usr/lib/security/pam_rhost_auth.so.1
rsh auth required /usr/lib/security/pam_rhost_auth.so.1
OTHER auth required /usr/lib/security/pam_unix.so.1
#
# Account management
#
OTHER account required /usr/lib/security/pam_unix.so.1
#
# Session management
#
OTHER session required /usr/lib/security/pam_unix.so.1
#
# Password management
#
OTHER password required /usr/lib/security/pam_unix.so.1
```

Normally, the entry for the *module_path* is "root-relative." If the file name you enter for *module_path* does not begin with a slash (/), the path `/usr/lib/security/` is prepended to the file name. A full path name must be used for modules located in other directories.

The values for the *module_options* can be found in the man pages for the module. (For example, `pam_unix`(5)).

The `use_first_pass` and `try_first_pass` options, which are supported by the `pam_unix` module, let users reuse the same password for authentication without retyping it.

If `login` specifies authentication through both `pam_local` and `pam_unix`, then the user is prompted to enter a password for each module. In situations where the passwords are the same, the `use_first_pass` module option prompts for only one password and uses that password to authenticate the user for both modules. If the passwords are different, the authentication fails. In general, this option should be used with an `optional` control flag, as shown below, to make sure that the user can still log in.

```
# Authentication management
#
login auth required /usr/lib/security/pam_unix.so.1
login auth optional /usr/lib/security/pam_local.so.1 use_first_pass
```

If the `try_first_pass` module option is used instead, the local module prompts for a second password if the passwords do not match or if an error is made. If both methods of authentication are necessary for a user to get access to all the needed

tools, using this option could cause some confusion since the user could get access with only one type of authentication.

Configuring PAM

The section below discusses some of the tasks that may be required to make the PAM framework fully functional. In particular, you should be aware of some of the security issues associated with the PAM configuration file.

Planning for PAM

When deciding how best to employ PAM in your environment, start by focusing on these issues:

- Determine what your needs are, especially which modules you should select.
- Identify the services that need special attention; use `OTHER` if appropriate.
- Decide on the order in which the modules should be run.
- Select the control flag for that module.
- Choose any options necessary for the module.

Here are some suggestions to consider before changing the configuration file:

- Use the `OTHER` entry for each module type so that every application does not have to be included.
- Make sure to consider the security implications of the `sufficient` and `optional` control flags.
- Review the man pages associated with the modules to understand how each module will function, what options are available, and the interactions between stacked modules.

Warning - If the PAM configuration file is misconfigured or gets corrupted, it is possible that even the superuser would be unable to log in. Since `sulogin` does not use PAM, the superuser would then be required to boot the machine into single user mode and fix the problem.

After changing the `/etc/pam.conf` file, review it as much as possible while still logged in as superuser. Test all of the commands that might have been affected by your changes. For example, if you added a new module to the `telnet` service, use the `telnet` command and verify that the changes you made behave as expected.

▼ How to Add a PAM Module

1. **Become superuser.**
2. **Determine which control flags and other options should be used.**
   Refer to “PAM Modules” on page 362 information on the module.
3. **Copy the new module to** `/usr/lib/security`**.**
4. **Set the permissions so that the module file is owned by root and permissions are** `555`**.**
5. **Edit the PAM configuration file,** `/etc/pam.conf`**, and add this module to the appropriate services.**

Verification

It is very important to do some testing *before* the system is rebooted in case the configuration file is misconfigured. Run `rlogin`, `su`, and `telnet` before rebooting the system. If the service is a daemon spawned only once when the system is booted, it may be necessary to reboot the system before you can verify that the module has been added.

▼ How to Prevent Unauthorized Access From Remote Systems With PAM

Remove the `rlogin auth rhosts_auth.so.1` entry from the PAM configuration file. This prevents reading the `~/.rhosts` files during an `rlogin` session and therefore prevents unauthenticated access to the local system from remote systems. All `rlogin` access requires a password, regardless of the presence or contents of any `~/.rhosts` or `/etc/hosts.equiv` files.

Note - To prevent other unauthenticated access to the `~/.rhosts` files, remember to disable the `rsh` service. The best way to disable a service is to remove the service entry from `/etc/inetd.conf`. Changing the PAM configuration file does not prevent the service from being started.

▼ How to Initiate PAM Error Reporting

1. **Edit the** `/etc/syslog.conf` **to add any of the following PAM error reporting entries**:
   - `auth.alert` — messages about conditions that should be fixed immediately
   - `auth.crit` — critical messages

- `auth.err` — error messages
- `auth.info` — informational messages
- `auth.debug` — debugging messages

2. **Restart the `syslog` daemon or send a `SIGHUP` signal to it to activate the PAM error reporting.**

Example—Initiating PAM Error Reporting

The example below displays all alert messages on the console. Critical messages are mailed to root. Informational and debug messages are added to the `/var/log/pamlog` file.

```
auth.alert /dev/console
auth.crit 'root'
auth.info;auth.debug /var/log/pamlog
```

Each line in the log contains a time stamp, the name of the system that generated the message, and the message itself. The `pamlog` file is capable of logging a large amount of information.

CHAPTER 21

SEAM Overview

This chapter provides an introduction to the Solaris 8 version of the SEAM product. The SEAM 1.0 product includes an implementation of the Kerberos V5 network authentication protocol. It is available in the Sun Easy Access Server (SEAS) 3.0 release.The Solaris 8 release does not include all parts of the SEAM product. Only the client-side product is included. This chapter includes information for both the client-side and the server-side parts of the SEAM product, so that the interaction of the whole product can be described. The following topics are covered:

- "What Is SEAM?" on page 373
- "SEAM Terminology" on page 374
- "SEAM Components" on page 376
- "How SEAM Works" on page 377
- "Security Services" on page 380

What Is SEAM?

Sun Enterprise Authentication Mechanism (SEAM) is a client/server architecture that offers strong user authentication, as well as data integrity and privacy, for providing secure transactions over networks. *Authentication* guarantees that the identities of both the sender and recipient of a network transaction are true; SEAM can also verify the validity of data being passed back and forth (*integrity*) and encrypt it during transmission (*privacy*). Using SEAM, you can log on to other machines, execute commands, exchange data, and transfer files securely. Additionally, SEAM provides *authorization* services, allowing administrators to restrict access to services and machines; moreover, as a SEAM user you can regulate other people's access to your account.

SEAM is a *single-sign-on* system, meaning that you only need to authenticate yourself to SEAM once per session, and all subsequent transactions during the session are automatically authenticated. You will not need to re-enter the password once your are authenticated. This means you do not have to send your password over the network, where it can be intercepted, each time you use these services.

SEAM is based on the Kerberos V5 network authentication protocol developed at the Massachusetts Institute of Technology (MIT). People who have used Kerberos V5 should therefore find SEAM very familiar. Since Kerberos V5 is a industry standard for network security (see RFC 1510), SEAM promotes interoperability with other systems. In other words, because SEAM works with systems using Kerberos V5, it allows for secure transactions even over heterogeneous networks. Moreover, SEAM provides authentication and security both between domains and within a single domain.

Note - Because SEAM is based on, and designed to interoperate with, Kerberos V5, this manual often uses the terms “Kerberos” and “SEAM” more or less interchangeably — for example, “Kerberos realm” or “SEAM-based utility.” (“Kerberos” and “Kerberos V5” are used interchangeably as well.) The manual draws distinctions when necessary.

SEAM allows for flexibility in running Solaris applications. You can configure SEAM to allow both SEAM-based and non-SEAM-based requests for network services, such as the NFS service. That means current Solaris applications still work even if they are running on systems on which SEAM is not installed. Of course, you can also configure SEAM to allow only SEAM-based network requests.

Additionally, applications do not have to remain committed to SEAM if other security mechanisms are developed. Because SEAM is designed to layer modularly under the Generic Security Service API, applications that make use of the GSS-API can utilize whichever security mechanism best suits their needs.

SEAM Terminology

The following section presents terms and their definitions that are used throughout the SEAM documentation. In order to follow many of the discussions, a understanding of these terms is essential.

Kerberos-Specific Terminology

Understanding the terms presented in this section, is needed when studying the sections about the administering the KDCs.

The *Key Distribution Center* or *KDC* is the portion of SEAM that is responsible for issuing credentials. These credentials are created using information stored in the KDC database. Each realm should have at least two KDCs, a master and at least one slave. All KDCs generate credentials, but only the master handles any changes to the KDC database.

A *stash file* contains a encrypted copy of the master key for the KDC. This key is used when a server is rebooted to automatically authenticate the KDC before starting `kadmind` and `krb5kdc`. Because this file includes the master key, the file and any backups of the file should be kept secure. If the encryption is compromised, then the key could be used to access or modify the KDC database.

Authentication-Specific Terminology

The terms discussed below are necessary for an understanding of the authentication process. Programmers and system administrators should be familiar with these terms.

A *client* is the software running on a user's workstation. The SEAM software running on the client makes many requests during this process, and it is important to differentiate the actions of this software from the user.

The terms server and service are often used interchangeably. To make things clearer, the term *server* is used to define the physical system that SEAM software is running on. The term *service* corresponds to a particular function that is being supported on a server (for instance, `ftp` or `nfs`). Documentation often mentions servers as part of a service, but using this definition clouds the meaning of the terms; therefore, servers refer to the physical system and service refers to the software.

The SEAM product includes three types of keys. One of them is the *private key*. This key is given to each user principal and is known only to the user of the principal and to the KDC. For user principals, the key is based on the user's password. For servers and services, the key is known as a *service key*. This key serves the same purpose as the private key, but is used by servers and services. The third type of key is a *session key*. This is a key generated by the authentication service or the ticket-granting service. A session key is generated to provide secure transactions between a client and a service.

A *ticket* is an information packet used to securely pass the identity of a user to a server or service. A ticket is good for only a single client and a particular service on a specific server. It contains the principal name of the service, the principal name of the user, the IP address of the user's host, a timestamp, and a value to define the lifetime of the ticket. A ticket is created with a random session key to be used by the client and the service. After a ticket has been created, it can be reused until the ticket expires.

A *credential* is a packet of information that includes a ticket and a matching session key. Credentials are often encrypted using either a private key or a service key depending on what will be decrypting the credential.

An *authenticator* is another type of information. When used with a ticket, an authenticator can be used to authenticate a user principal. An authenticator includes the principal name of the user, the IP address of the user's host, and a timestamp. Unlike a ticket, an authenticator can be used once only, usually when access to a service is requested. An authenticator is encrypted using the session key for that client and that server.

SEAM Components

The full release of SEAM 1.0 in SEAS 3.0 includes many components, including:

- Key Distribution Center (KDC)
- Database administration programs
- User programs for obtaining, viewing and destroying tickets
- Kerberized applications — `telnet`
- Administration utilities
- Additions to the Pluggable Authentication Module (PAM)

The list of all of the components in the SEAM 1.0 release can be found in `Introduction to SEAM 1.0`.

The Solaris 8 release includes only the client-side portions of SEAM, so many of these components are not included. This enables systems running the Solaris 8 release to become SEAM clients without having to install SEAM separately. To use this funtionality you must install a KDC using either SEAS 3.0, the MIT distribution, or Windows2000. The client-side components are not useful without a configured KDC to distribute tickets. The following components are included in this release:

- User programs for obtaining, viewing, and destroying tickets — `kinit`, `klist`, `kdestroy` — and for changing your SEAM password — `kpasswd`
- Key table administration utility — `ktutil`
- Additions to the Pluggable Authentication Module (PAM) — Allows applications to use various authentication mechanisms; PAM can be used to make login and logouts transparent to the user.
- GSS_API plug–ins — Provides Kerberos protocol and cryptographic support
- NFS client and server support

How SEAM Works

The following is a generalized overview of the SEAM authentication system. For a more detailed description, see “How the Authentication System Works” on page 408.

From the user’s standpoint, SEAM is mostly invisible after the SEAM session has been started. Initializing a SEAM session often involves no more than logging in and providing a Kerberos password.

The SEAM system revolves around the concept of a *ticket*. A ticket is a set of electronic information that serves as identification for a user or a service such as the NFS service. Just as your driver’s license identifies you and indicates what driving permissions you have, so a ticket identifies you and your network access privileges. When you perform a SEAM-based transaction — for example, if you `kinit` to a new principal — you transparently send a request for a ticket to a *Key Distribution Center*, or KDC, which accesses a database to authenticate your identity. The KDC returns a ticket granting you permission to access the other machine. “Transparently” means that you do not need to explicitly request a ticket; it happens as part of the `kinit` command. Because only the authenticated client can get a ticket for a specific service, another client cannot use `kinit` under an assumed identity.

Tickets have certain attributes associated with them. For example, a ticket can be *forwardable* (meaning that it can be used on another machine without a new authentication process), or *postdated* (not valid until a specified time). How tickets are used — for example, which users are allowed to obtain which types of ticket — is set by *policies* determined when SEAM is installed or administered.

Note - You will frequently see the terms *credential* and *ticket*. In the greater Kerberos world, they are often used interchangeably. Technically, however, a credential is a ticket plus the *session key* for that session. This difference is explained in more detail in “Gaining Access to a Service Using SEAM” on page 409.

Principals

A client in SEAM is identified by its *principal*. A principal is a unique identity to which the KDC can assign tickets. A principal can be a user, such as `joe`, or a service, such as `nfs`.

By convention, a principal name is divided into three parts: the *primary*, the *instance*, and the *realm*. A typical SEAM principal would be, for example, `joe/admin@ENG.ACME.COM`, where:

- `joe` is the primary. This can be a username, as shown here, or a service, such as `nfs`. It can also be the word `host`, signifying that this is a service principal set up to provide various network services.

- `admin` is the instance. An instance is optional in the case of user principals, but it is required for service principals. For example: if the user `joe` sometimes acts as a system administrator, he can use `joe/admin` to distinguish himself from his usual user identity. Likewise, if `joe` has accounts on two different hosts, he can use two principal names with different instances (for example, `joe/denver.acme.com` and `joe/boston.acme.com`). Notice that SEAM treats `joe` and `joe/admin` as two completely different principals.

  In the case of a service principal, the instance is the fully qualified hostname. `bigmachine.eng.acme.com` is an example of such an instance, so that the primary/instance might be, for example, `nfs/bigmachine.eng.acme.com` or `host/bigmachine.eng.acme.com`.
- `ENG.ACME.COM` is the SEAM realm. Realms are discussed in the next section.

The following are all valid principal names:

- `joe`
- `joe/admin`
- `joe/admin@ENG.ACME.COM`
- `nfs/host.eng.acme.com@ENG.ACME.COM`
- `host/eng.acme.com@ENG.ACME.COM`

Realms

A realm is a logical network, like a domain, which defines a group of systems under the same *master KDC*. The figure below shows how realms can relate to one another. Some realms are hierarchical, with one being a superset of the other. Otherwise, the realms are non-hiearchical and the mapping between the two realms must be defined. A feature of SEAM is that it permits authentication across realms; each realm only needs to have a principal entry for the other realm in its KDC.

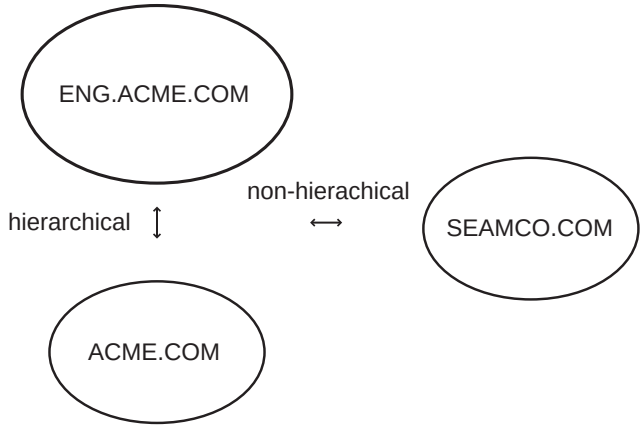


Figure 21–1 Realms

Realms and Servers

Each realm must include a server that maintains the master copy of the principal database. This is called the *master KDC server*. Additionally, each realm should contain at least one *slave KDC server*, which contains duplicate copies of the principal database. Both the master and the slave KDC servers create tickets used to establish authentication.

The realm can also include two additional types of SEAM servers. A SEAM network *application server* is a server that provides access to Kerberized applications (such as `ftp`, `telnet` and `rsh`). Realms can also include *NFS servers*, which provide NFS services, using Kerberos authentication.

The figure below shows what a hypothetical realm might contain.

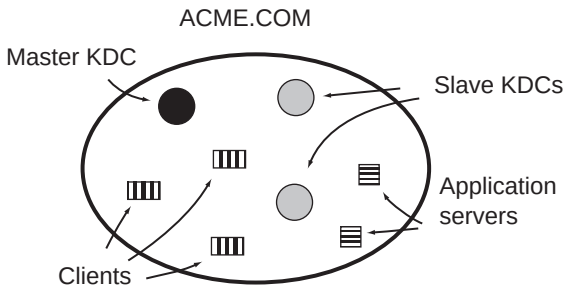


Figure 21–2 A Typical Realm

Security Services

In addition to providing secure authentication of users, SEAM provides two security services:

- *Integrity*. Just as authentication ensures that clients on a network are who they claim to be, integrity ensures that the data they send is valid and has not been tampered with during transit. This is done through cryptographic checksumming of the data. Integrity also includes user authentication.
- *Privacy*. Privacy takes security a step further. It not only includes verifying the integrity of transmitted data, but it encrypts the data before transmission, protecting it from eavesdroppers. It authenticates users, as well.

Note - Privacy support is included in the Solaris Encryption Kit CD.

Developers can design their RPC-based applications to choose a security service by using the RPCSEC_GSS programming interface.

CHAPTER 22

Configuring SEAM

This chapter provides configuration procedures for network application servers, NFS servers and SEAM clients. Many of these procedures require root access, so they should be used by System Administrators or advanced users.

- "SEAM Administration Task Map" on page 381
- "Configuring SEAM Clients" on page 382
- "Configuring SEAM NFS Servers Task Map" on page 385
- "Synchronizing Clocks Between KDCs and SEAM Clients" on page 391
- "SEAM Client Error Messages" on page 392

SEAM Administration Task Map

This table lists the administration tasks required for the Solaris 8 version of SEAM. These procedures require that a KDC with an admin server is installed.

TABLE 22–1 SEAM Administration Task Map

Task	Description	For Instructions, Go To ...
Configure SEAM client	Steps to manually configure a SEAM client	"Configuring SEAM Clients" on page 382
(Optional) Install NTP	For SEAM to work properly, the clocks on all systems in the realm must be kept in sync.	"Synchronizing Clocks Between KDCs and SEAM Clients" on page 391
Configure SEAM NFS Server	Steps to enable a server to share a file system requiring Kerberos authentication.	"Configuring SEAM NFS Servers Task Map" on page 385

Configuring SEAM Clients

SEAM clients include any host, not a KDC server, on the network that needs to use SEAM services. This section provides a procedure for installing a SEAM client, as well as specific information about using root authentication to mount NFS file systems.

There are two procedures which can be used to configure a SEAM client. "How to Finish the Configuration of a SEAM Client" on page 385 provides information for configuring a SEAM client that has been partially setup during the installation of the system."How to Configure a SEAM Client" on page 382 provides the steps for configuring a SEAM client where no configuration of SEAM was attempted during the installation of the Solaris 8 release.

▼ How to Configure a SEAM Client

The following configuration parameters are used:

realm name = `ACME.COM`

DNS domain name = `acme.com`

master KDC = `kdc1.acme.com`

slave KDC = `kdc2.acme.com`

client = `client.acme.com`

admin principal = `kws/admin`

user principal = `mre`

1. **Prerequisites for configuring a SEAM client.**

   A KDC with an admin server must be configured and running. In addition, DNS must be installed and the `/etc/resolv.conf` file should be configured properly.

2. **Become superuser on the client.**

3. **Edit the PAM configuration file (`pam.conf`).**

   Remove the comments from the last eight lines to enable the Kerberos PAM module.

```
client1 # tail -11 /etc/pam.conf
#
# Support for Kerberos V5 authentication (uncomment to use Kerberos)
#
rlogin auth optional   /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
login  auth optional   /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
dtlogin        auth optional   /usr/lib/security/$ISA/
pam_krb5.so.1 try_first_pass
other  auth optional   /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
dtlogin        account optional /usr/lib/security/$ISA/pam_krb5.so.1
other  account optional /usr/lib/security/$ISA/pam_krb5.so.1
other  session optional /usr/lib/security/$ISA/pam_krb5.so.1
other  password optional /usr/lib/security/$ISA/
pam_krb5.so.1 try_first_pass
```

4. **Edit the NFS security service configuration file (`nfssec.conf`).**

   Remove the comments from the lines describing the Kerberos services.

```
client1 # cat /etc/nfssec.conf
  .
  .
#
# Uncomment the following lines to use Kerberos V5 with NFS
#
krb5            390003  kerberos_v5     default -       # RPCSEC_GSS
krb5i           390004  kerberos_v5     default integrity       # RPCSEC_GSS
default          1       -       -       -       # default is AUTH_SYS
```

5. **Edit the Kerberos configuration file (`krb5.conf`).**

   To change the file from the default version, you need to change the realm names and the names of the servers.

```
client1 # cat /etc/krb5/krb5.conf
[libdefaults]
        default_realm = ACME.COM

[realms]
                ACME.COM = {
                kdc = kdc1.acme.com
                kdc = kdc2.acme.com
                admin_server = kdc1.acme.com
        }

[domain_realm]
        .acme.com = ACME.COM
```

6. **(Optional) Synchronize with the master KDC's clock using NTP or another clock synchronization mechanism.**

   See "Synchronizing Clocks Between KDCs and SEAM Clients" on page 391 for information about NTP.

7. **Add new principals.**

   Using the administration tool provided with your KDC add new principals for the client.

   a. **Create the NFS service principal.**

      Create a principal named: `nfs/client1.acme.com`.

   b. **Create a `root` principal.**

      Create a principal named: `root/client1.acme.com`.

   c. **Create a `host` principal.**

      Create a principal named: `host/client1.acme.com`.

   d. **Add the `root` principal to the keytab file.**

      Make sure that the `root/client1.acme.com` principal is included in the keytab file.

8. **If you want the client to warn users about Kerberos ticket expiration, configure an entry in the `/etc/krb5/warn.conf` file.**

   See `warn.conf(4)` for more information.

▼ How to Finish the Configuration of a SEAM Client

To configure a SEAM client, after a partial installation has been done when installing the client, follow the instructions in “How to Configure a SEAM Client” on page 382. Because the installation has been started, verify the contents of `pam.conf`, `nfssec.conf`, and `krb5.conf` instead of editting them.

Configuring SEAM NFS Servers Task Map

NFS services use UNIX UIDs to identify a user and cannot directly use principals. To translate the principal to a UID, a credential table that maps user principals to UNIX UIDs must be created. The procedures below focus on the tasks necessary to configure a SEAM NFS server, administer the credential table, and to initiate Kerberos security modes for NFS-mounted file systems. The following table describes the tasks covered in this section.

TABLE 22–2 Configuring SEAM NFS Server Task Map

Task	Description	For Instructions, Go To ...
Configure a SEAM NFS Server	Steps to enable a server to share a file system requiring Kerberos authentication.	“How to Configure SEAM NFS Servers” on page 386
Change the Back-end Mechanism for the Credential Table	Steps to define the back-end mechanism that is used by `gsscred`.	“How to Change the Back-end Mechanism for the `gsscred` Table” on page 387
Create a Credential Table	Steps to generate a credential table.	“How to Create a Credential Table” on page 387

TABLE 22–2 Configuring SEAM NFS Server Task Map *(continued)*

Task	Description	For Instructions, Go To ...
How to Change the Credential Table That Maps User Principles to UNIX UIDs.	Steps to update information in the credential table.	"How to Add a Single Entry to the Credential Table" on page 388
Share a File System With Kerberos Authentication	Steps to share a file system with security modes so that Kerberos authentication is required.	"How to Set Up a Secure NFS Environment With Multiple Kerberos Security Modes" on page 389

▼ How to Configure SEAM NFS Servers

This procedure requires that the master KDC has been configured. To fully test the process you need several clients. The following configuration parameters are used:

realm name = `ACME.COM`

DNS domain name = `acme.com`

NFS server = `denver.acme.com`

admin principle = `kws/admin`

1. **Prerequisites for configuring a SEAM NFS server.**
   The SEAM client software must be installed.

2. **(Optional) Install NTP client or other clock synchronization mechanism.**
   See "Synchronizing Clocks Between KDCs and SEAM Clients" on page 391 for information about NTP.

3. **Add new principals.**
   Using the administration tool provided with your KDC add new principals for the NFS server.
   a. **Create the server's NFS service principal.**
      Create a principal named: `nfs/denver.acme.com`.

   b. **(Optional) Create a `root` principal for the NFS server.**
      Create a principal named: `root/denver.acme.com`.

   c. **Add the server's NFS service principal to the server's keytab.**
      Make sure that the `nfs/denver.acme.com` principal is included in the keytab file.

4. **Create the `gsscred` table.**

   See "How to Create a Credential Table" on page 387 for more information.

5. **Share the NFS file system using Kerberos security modes.**

   See "How to Set Up a Secure NFS Environment With Multiple Kerberos Security Modes" on page 389 for more information.

6. **On each client, authenticate both the user and root principals.**

▼ How to Change the Back-end Mechanism for the `gsscred` Table

1. **Become superuser on the NFS server.**

2. **Edit `/etc/gss/gsscred.conf` and change the mechanism.**

   One of the following back-end mechanisms can be used: `files`, `xfn_files`, `xfn_nis`, `xfn_nisplus`, or `xfn`. The advantages of each of these mechanisms is covered in "Using the `gsscred` Table" on page 412.

▼ How to Create a Credential Table

The `gsscred` credential table is used by an NFS server to map SEAM principals to a UID. In order for NFS clients to be able to mount file systems from an NFS server using Kerberos authentication, this table must be created or made available.

1. **Become superuser on the appropriate server.**

   Which server you run this command from and under what ID you run the command depends on the back-end mechanism that has been selected to support the `gsscred` table. For all mechanisms except `xfn_nisplus`, you must become `root`.

If Your Back-end Mechanism Is ...	Then ....
`files`	Run on the NFS server.
`xfn`	Select host based on the default `xfn` file setting.
`xfn_files`	Run on the NFS server.

If Your Back-end Mechanism Is ...	Then ....
`xfn_nis`	Run on the NIS master.
`xfn_nisplus`	Run anywhere as long as the permissions to change the NIS+ data are in place.

2. **(Optional) If `/var/fn` does not exist and you want to use one of the `xfn` options, create an initial XFN database.**

```
# fnselect files
# fncreate -t org -o org//
```

3. **Create the credential table using `gsscred`.**

   The command gathers information from all of the sources listed with the `passwd` entry in `/etc/nsswitch.conf`. You might need to temporarily remove the `files` entry, if you do not want the local password entries included in the credential table. See the `gsscred`(1M) man page for more information.

```
# gsscred -m kerberos_v5 -a
```

▼ How to Add a Single Entry to the Credential Table

This procedure requires that the `gsscred` table has already been installed on the NFS server.

1. **Become superuser on a NFS server.**

2. **Add an entry to the table using `gsscred`.**

```
# gsscred -m [mech] -n [name] -u [uid] -a
```

mech	The security mechanism to be used.
name	The principal name for the user, as defined in the KDC.
uid	The UID for the user, as defined in the password database.
–a	Adds the UID to principal name mapping.

Example—Changing a Single Entry to the Credential Table

The following example adds an entry for the user named `sandy`, which is mapped to UID `3736`. The UID is pulled from the password file, if it is not included on the command line.

```
# gsscred -m kerberos_v5 -n sandy -u 3736 -a
```

▼ How to Set Up a Secure NFS Environment With Multiple Kerberos Security Modes

1. **Become superuser on the NFS server.**
2. **Edit the `/etc/dfs/dfstab` file and add the `sec=` option with the required security modes to the appropriate entries.**

```
# share -F nfs -o sec=mode filesystem
```

mode	The security modes to be used when sharing. When using multiple security modes, the first mode in the list is used as the default by autofs.
filesystem	The path to the file system to be shared.

All clients attempting to access files from the named file system require Kerberos authentication. To complete accessing files, both the user principal and the `root` principal on the NFS client should be authenticated.

3. **Check to be sure the NFS service is running on the server.**

   If this is the first share command or set of share commands that you have initiated, it is likely that the NFS daemons are not running. The following set of commands kill the daemons and restart them.

```
# /etc/init.d/nfs.server stop
# /etc/init.d/nfs.server start
```

4. **Optional: If autofs is being used, edit the `auto_master` data to select a security mode other than the default.**

   You need not follow this procedure if you are not using autofs to access the file system or if the default selection for the security mode is acceptable.

```
/home   auto_home  -nosuid,sec=krbi
```

5. **Optional: Manually issue the `mount` command to access the file system using a non-default mode.**

   Alternatively, you could use the `mount` command to specify the security mode, but this does not take advantage of the automounter:

```
# mount -F nfs -o sec=krb5p /export/home
```

Example—Sharing a File System With One Kerberos Security Mode

This example will require Kerberos authentication before files can be accessed.

```
# share -F nfs -o sec=krb5 /export/home
```

Example—Sharing a File System With Multiple Kerberos Security Modes

In this example, all three Kerberos security modes have been selected. If no security mode is specified when a mount request is made, the first mode listed is used on all NFS V3 clients (in this case, `krb5`). Additional information can be found in “Changes to the `share` Command” on page 403.

```
# share -F nfs -o sec=krb5:krb5i:krb5p /export/home
```

Synchronizing Clocks Between KDCs and SEAM Clients

All hosts participating in the Kerberos authentication system must have their internal clocks synchronized within a specified maximum amount of time (known as *clock skew*), which provides another Kerberos security check. If the clock skew is exceeded between any of the participating hosts, client requests will be rejected.

The clock skew also determines how long application servers must keep track of all Kerberos protocol messages, in order to recognize and reject replayed requests. So, the longer the clock skew value, the more information that application servers have to collect.

The default value for the maximum clock skew is 300 seconds (five minutes), which you can change in the `libdefaults` section of the `krb5.conf` file.

Note - For security reasons, do not increase the clock skew beyond 300 seconds.

Since it is important to maintain synchronized clocks between the KDCs and SEAM clients, it is recommended that you use the Network Time Protocol (NTP) software to do this. The NTP public domain software from the University of Delaware is included in the Solaris software starting with the Solaris 2.6 release.

Note - Another way to synchronize clocks is to use the `rdate` command with `cron` jobs, which can be a less involved process than using NTP. However, this section will continue to focus on using NTP. And, if you use the network to synchronize the clocks, the clock synchronization protocol must itself be secure.

NTP enables you to manage precise time and network clock synchronization in a network environment. NTP is basically a server/client implementation. You pick one system to be the master clock (NTP server), and then you set up all your other systems to synchronize their clocks with the master clock (NTP clients). This is done through the `xntpd` daemon, which sets and maintains a UNIX system time-of-day in agreement with Internet standard time servers. The figure below shows an example of server/client NTP implementation.

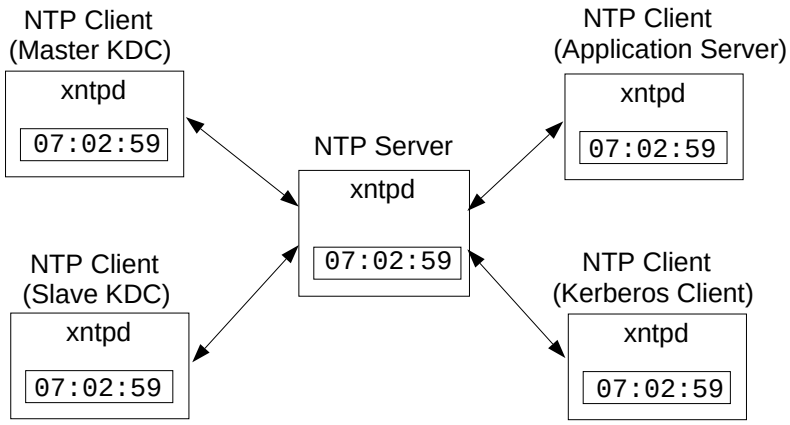


Figure 22–1 Synchronizing Clocks Using NTP

Ensuring that the KDCs and SEAM clients maintain synchronized clocks involves the following:

1. Set up an NTP server on your network (this can be any system except the master KDC). See "How to Set Up an NTP Server" on page 460.
2. As you configure the KDCs and SEAM clients on the network, set them up to be NTP clients of the NTP server. See "How to Set Up an NTP Client" on page 460.

SEAM Client Error Messages

Refer to `SEAM 1.0 Error Messages and Troubleshooting` for a complete list of SEAM error messages.

CHAPTER 23

SEAM Reference

This chapter includes information on getting, viewing and destroying tickets and choosing or changing a Kerberos password on a system running SEAM. In addition, this chapter lists many of the SEAM product files. Also, a more detailed description of how the Kerberos authentication system works is provided.

- "Ticket Management" on page 393
- "Password Management" on page 397
- "SEAM Files" on page 401
- "SEAM Daemons" on page 404
- "Ticket Reference" on page 404
- "How the Authentication System Works" on page 408
- "Gaining Access to a Service Using SEAM" on page 409
- "Using the `gsscred` Table" on page 412

Ticket Management

This section explains how to obtain, view, and destroy tickets. For an introduction to tickets, see "How SEAM Works" on page 377.

Do You Need to Worry About Tickets?

PAM can be set up to automatically get tickets when you log in. It is possible that your SEAM configuration does not include this automatic forwarding of tickets, but it is the default behavior.

Most of the Kerberized commands also automatically destroy your tickets when they exit. However, you might want to explicitly destroy your Kerberos tickets with `kdestroy` when you are through with them, just to be sure. See "How to Destroy Tickets" on page 396 for more information on `kdestroy`.

For information on ticket lifetimes, see "Ticket Lifetimes" on page 406.

▼ How to Create a Ticket

Normally a ticket is created automatically when you log in and you need not do anything special to obtain one. However, you might need to create a ticket in the following cases:

- Your ticket expires.
- You need to use a different principal besides your default principal. (For example, if you use `rlogin -l` to log in to a machine as someone else.)

To create a ticket, use the `kinit` command.

```
% /usr/bin/kinit
```

`kinit` prompts you for your password. For the full syntax of the `kinit` command, see the `kinit(1)` man page.

Example—Creating a Ticket

This example shows a user, `jennifer`, creating a ticket on her own system.

```
% kinit
Password for jennifer@ENG.ACME.COM:  <enter password>
```

Here the user `david` creates a ticket good for three hours with the `–l` option.

```
% kinit -l 3h david@ACME.ORG
Password for david@ACME.ORG:  <enter password>
```

This example shows `david` creating a forwardable ticket (with `–f`) for himself. With this forwardable ticket, he can (for example) log in to a second system, and then `telnet` to a third system.

```
% kinit -f david@ACME.ORG
Password for david@ACME.ORG:      <enter password>
```

For more on how forwarding tickets works, see “Types of Tickets” on page 404.

▼ How to View Tickets

Not all tickets are alike. One ticket might be, for example, *forwardable*; another might be *postdated*; while a third might be both. You can see which tickets you have, and what their attributes are, by using the `klist` command with the `–f` option:

```
% /usr/bin/klist -f
```

The following symbols indicate the attributes associated with each ticket, as displayed by `klist`:

F	Forwardable
f	Forwarded
P	Proxiable
p	Proxy
D	Postdateable
d	Postdated
R	Renewable
I	Initial
i	Invalid

“Types of Tickets” on page 404 describes the various attributes a ticket can have.

Example—Viewing Tickets

This example shows that the user `jennifer` has an *initial* ticket, which is *forwardable* (`F`) and *postdated* (`d`), but not yet validated (`i`).

```
% /usr/bin/klist -f
Ticket cache: /tmp/krb5cc_74287
Default principal: jenniferm@ENG.ACME.COM

Valid starting                Expires                Service principal
09 Mar 99 15:09:51  09 Mar 99 21:09:51  nfs/ACME.SUN.COM@ACME.SUN.COM
```

```
        renew until 10 Mar 99 15:12:51, Flags: Fdi
```

The example below shows that the user `david` has two tickets that were *forwarded* (`f`) to his host from another host. The tickets are also (re)*forwardable* (`F`):

```
% klist -f
Ticket cache: /tmp/krb5cc_74287
Default principal: david@ACME.SUN.COM

Valid starting                 Expires                 Service principal
07 Mar 99 06:09:51  09 Mar 99 23:33:51  host/ACME.COM@ACME.COM
        renew until 10 Mar 99 17:09:51, Flags: fF

Valid starting                 Expires                 Service principal
08 Mar 99 08:09:51  09 Mar 99 12:54:51  nfs/ACME.COM@ACME.COM
        renew until 10 Mar 99 15:22:51, Flags: fF
```

▼ How to Destroy Tickets

Tickets are generally destroyed automatically when the commands that created them exit; however, you might want to explicitly destroy your Kerberos tickets when you are through with them, just to be sure. Tickets can be stolen, and if this happens, the person who has them can use them until they expire (although stolen tickets must be decrypted).

To destroy your tickets, use the `kdestroy` command.

```
% /usr/bin/kdestroy
```

`kdestroy` destroys *all* your tickets. You cannot use it to selectively destroy a particular ticket.

If you are going to be away from your system and are concerned about an intruder using your permissions, you should either use `kdestroy` or a screensaver that locks the screen.

Note - One way to help ensure that tickets are always destroyed is to add the `kdestroy` command to the `.logout` file in your home directory.

In cases where the PAM module has been configured, tickets are destroyed automatically upon logout, so adding a call to `kdestroy` to your `.login` file is not necessary. However, if the PAM module has not been configured, or if you don't know whether it has or not, you might want to add `kdestroy` to your `.login` file to be sure that tickets are destroyed when you exit your system.

Password Management

With SEAM installed, you now have two passwords: your regular Solaris password, and a Kerberos password. You can make both passwords the same or they can be different.

Non-Kerberized commands, such as `login`, can be set up through PAM to authenticate with both Kerberos and UNIX. If you have different passwords, you must provide both passwords to log on with the appropriate authentication. However, if both passwords are the same, the first password you enter for UNIX is also accepted by Kerberos.

Unfortunately, using the same password for both can compromise security. That is, if someone discovers your Kerberos password, then your UNIX password is no longer a secret. However, using the same passwords for UNIX and Kerberos is still more secure than a site without Kerberos, because passwords in a Kerberos environment are not sent across the network. Usually, your site will have a policy to help you determine your options.

Your Kerberos password is the only way Kerberos has of verifying your identity. If someone discovers your Kerberos password, Kerberos security becomes meaningless, for that person can masquerade as you — send email that comes from "you," read, edit, or delete your files, or log into other hosts as you — and no one will be able to tell the difference. For this reason, it is vital that you choose a good password and keep it secret. You should *never* reveal your password to anyone else, not even your system administrator. Additionally, you should change your password frequently, particularly any time you believe someone might have discovered it.

Advice on Choosing a Password

Your password can include almost any character you can type (the main exceptions being control keys and the Return key). A good password is one that you can

remember readily, but which no one else can easily guess. Examples of bad passwords include:

- Words that can be found in a dictionary
- Any common or popular name
- The name of a famous person or character
- Your name or username in any form (for example: backward, repeated twice, and so forth)
- A spouse's, child's, or pet's name
- Your birth date or a relative's birth date
- Your Social Security number, driver's license number, passport number, or similar identifying number
- Any sample password that appears in this or any other manual

A good password is at least eight characters long. Moreover, a password should include a mix of characters, such as upper- and lower-case letters, numbers, and punctuation marks. Examples of passwords that would be good if they didn't appear in this manual include:

- Acronyms, such as "I2LMHinSF" (recalled as "I too left my heart in San Francisco")
- Easy-to-pronounce nonsense words, like "WumpaBun" or "WangDangdoodle!"
- Deliberately misspelled phrases, such as "6o'cluck" or "RrriotGrrrlsRrrule!"

Caution - Don't use these examples. Passwords that appear in manuals are the first ones an intruder will try.

Changing Your Password

You can change your Kerberos password in two ways:

- With the usual UNIX `passwd` command. With SEAM installed, the Solaris `passwd` command also automatically prompts for a new Kerberos password.

  The advantage of using `passwd` instead of `kpasswd` is that you can set both passwords (UNIX and Kerberos) at the same time. However, generally you do not *have* to change both passwords with `passwd`; often you can change only your UNIX password and leave the Kerberos password untouched, or vice-versa.

Note - The behavior of `passwd` depends on how the PAM module is configured. You may be required to change both passwords in some configurations. For some sites the UNIX password must be changed, while others require the Kerberos password to change.

- With the kpasswd command. kpasswd is very similar to passwd. One difference is that kpasswd changes only Kerberos passwords — you must use passwd if you want to change your UNIX password.

  Another difference is that kpasswd can change a password for a Kerberos principal that is not a valid UNIX user. For example, david/admin is a Kerberos principal, but not an actual UNIX user, so you must use kpasswd instead of passwd.

Warning - Using kpasswd requires the use of the SEAM 1.0 administration system which is included in the SEAS 3.0 release. In addition, privacy support must be loaded to protect the requests to change the password.

After you change your password, it takes some time for the change to propagate through a system (especially over a large network). Depending on how your system is set up, this might be anywhere from a few minutes to an hour or more. If you need to get new Kerberos tickets shortly after changing your password, try the new password first. If the new password doesn't work, try again using the old one.

Kerberos V5 allows system administrators to set criteria about allowable passwords for each user. Such criteria is defined by the *policy* set for each user (or by a default policy)— see XREF for more on policies. For example, suppose that jennifer's policy (call it jenpol) mandates that passwords be at least eight letters long and include a mix of at least two kinds of characters. kpasswd will therefore reject an attempt to use sloth as a password:

```
% kpasswd
kpasswd: Changing password for jennifer@ENG.ACME.COM.
Old password:    <jennifer enters her existing password>
kpasswd: jennifer@ENG.ACME.COM's password is controlled by
the policy jenpol
which requires a minimum of 8 characters from at least 2 classes
(the five classes are lowercase, uppercase, numbers, punctuation,
and all other characters).
New password: <jennifer enters 'sloth'>
New password (again):  <jennifer re-enters 'sloth'>
kpasswd: New password is too short.
Please choose a password which is at least 4 characters long.
```

Here jennifer uses slothrop49 as a password. slothrop49 meets the criteria, because it is over eight letters long and contains two different kinds of characters (numbers and lowercase letters):

```
% kpasswd
kpasswd: Changing password for jennifer@ENG.ACME.COM.
Old password:  <jennifer enters her existing password>
kpasswd: jennifer@ENG.ACME.COM's password is controlled by
the policy jenpol
which requires a minimum of 8 characters from at least 2 classes
```

(Continuation)

```
(the five classes are lowercase, uppercase, numbers, punctuation,
and all other characters).
New password:  <jennifer enters 'slothrop49'>
New password (again):  <jennifer re-enters 'slothrop49'>
Kerberos password changed.
```

Examples—Changing Your Password

The following example shows `david` changing both his UNIX and Kerberos passwords with `passwd`.

```
% passwd
 passwd:  Changing password for david
 Enter login (NIS+) password:          <enter the current UNIX password>
 New password:                         <enter the new UNIX password>
 Re-enter password:                    <confirm the new UNIX password>
 Old KRB5 password:                    <enter the current Kerberos password>
 New KRB5 password:                    <enter the new Kerberos password>
 Re-enter new KRB5 password:           <confirm the new Kerberos password>
```

In the above example `passwd` asks for both the UNIX and Kerberos password; however, if *try_first_pass* is set in the PAM module, the Kerberos password is automatically set to be the same as the UNIX password. (That is the default configuration.) In that case, `david` must use `kpasswd` to set his Kerberos password to something else, as shown next.

This example shows him changing only his Kerberos password with `kpasswd`:

```
% kpasswd
kpasswd: Changing password for david@ENG.ACME.COM.
Old password:            <enter the current Kerberos password>
New password:            <enter the new Kerberos password>
New password (again):    <confirm the new Kerberos password>
Kerberos password changed.
```

In this example, `david` changes the password for the Kerberos principal `david/admin` (which is not a valid UNIX user). To do this he must use `kpasswd`.

```
% kpasswd david/admin
kpasswd:  Changing password for david/admin.
Old password:            <enter the current Kerberos password>
New password:            <enter the new Kerberos password>
New password (again):    <confirm the new Kerberos password>
Kerberos password changed.
```

SEAM Files

This section lists the files included in the SEAM product.

TABLE 23–1 SEAM Files

File Name	Description
`/etc/gss/gsscred.conf`	Default file types for the `gsscred` table
`/etc/gss/mech`	Mechanisms for RPCSEC_GSS
`/etc/gss/qop`	Quality of Protection parameters for RPCSEC_GSS
`/etc/nfssec.conf`	Defines NFS authentication security modes
`/etc/krb5/krb5.conf`	Kerberos realm configuration file
`/etc/krb5/krb5.keytab`	Keytab for network application servers
`/etc/krb5/warn.conf`	Kerberos warning configuration file
`/etc/pam.conf`	PAM configuration file
`/tmp/krb5cc_`*uid*	Default credentials cache (*uid* is the decimal UID of the user)
`/tmp/ovsec_adm.`*xxxxxx*	Temporary credentials cache for the lifetime of the password changing operation (*xxxxxx* is a random string)

PAM Configuration File

The default PAM configuration file delivered with SEAM includes commented out entries to use the Kerberos capabilities. The new file includes entries for the authentication service, account management, session management, and password management modules.

For the authentication module, the new entries are for `rlogin`, `login`, and `dtlogin`. An example of these entries is shown below. All of these services use the new PAM library, `/usr/lib/security/pam_krb5.so.1`, to provide Kerberos authentication.

The first three entries employ the `try_first_pass` option, which requests authentication using the user's initial password. Using the initial password means that the user is not prompted for another password even if multiple mechanisms are listed. An `other` entry is included as the default entry for all entries requiring authentication that are not specified.

```
# cat /etc/pam.conf
 .
 .
rlogin auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
login auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
dtlogin auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
krlogin auth required /usr/lib/security/pam_krb5.so.1 acceptor
ktelnet auth required /usr/lib/security/pam_krb5.so.1 acceptor
krsh auth required /usr/lib/security/pam_krb5.so.1 acceptor
other auth optional /usr/lib/security/pam_krb5.so.1 try_first_pass
```

For the account management, `dtlogin` has a new entry that uses the Kerberos library, as shown below. An `other` entry is included to provide a default rule. Currently no actions are taken by the `other` entry.

```
dtlogin account optional /usr/lib/security/pam_krb5.so.1
other account optional /usr/lib/security/pam_krb5.so.1
```

The last two entries in the `/etc/pam.conf` file are shown below. The `other` entry for session management destroys user credentials. The new `other` entry for password management selects the Kerberos library.

```
other session optional /usr/lib/security/pam_krb5.so.1
other password optional /usr/lib/security/pam_krb5.so.1 try_first_pass
```

SEAM Commands

This section lists some of the commands included in the SEAM product.

TABLE 23–2 SEAM Commands

File Name	Description
`/usr/bin/kdestroy`	Destroys Kerberos tickets
`/usr/bin/kinit`	Obtains and caches Kerberos ticket-granting ticket
`/usr/bin/klist`	Lists current Kerberos tickets
`/usr/bin/kpasswd`	Changes Kerberos passwords
`/usr/bin/ktutil`	Keytab maintenance utility
`/usr/sbin/gsscred`	Generates and validates GSS-API tokens for NFS services

Changes to the `share` Command

In addition to the new SEAM commands, the Solaris 8 release includes new security flavors to be used with the `share` command. These modes are defined in the `/etc/nfssec.conf` file. These new security modes can be used by the `share` command:

krb5 Select Kerberos authentication

krb5i Select Kerberos authentication with integrity

krb5p Select Kerberos authentication with integrity and privacy

When multiple modes are included with the `share` command, the first mode listed is used by default if the client does not specify a security mode. Otherwise, the mode that the client selected is used.

If a mount request using a Kerberos mode fails, the mount completes using `none` as the security mode. This often occurs when the root principal on the NFS client is not authenticated. The mount request might succeed, but the user will be unable to access the files unless they are authenticated through Kerberos. Any transactions between the client and the server require Kerberos authentication, even if the file system is not mounted using a Kerberos security mode.

SEAM Daemons

The daemons that are used by the SEAM product are listed in the following table.

TABLE 23–3 SEAM Daemons

File Name	Description
`/usr/lib/krb5/ktkt_warnd`	Kerberos warning daemon
`/usr/lib/gss/gssd`	GSSAPI daemon

Ticket Reference

The following section presents additional information about tickets.

Types of Tickets

Tickets have properties that govern how they can be used. These properties are assigned to the ticket when it is created, although you can modify a ticket's properties later. (For example, a ticket can change from *forwardable* to *forwarded*.) You can view ticket properties with the `klist` command (see "How to View Tickets" on page 395).

Tickets can be described by one or more of the following terms:

forwardable/forwarded — A forwardable ticket can be sent from one host to another, obviating the need for a client to reauthenticate itself. For example, if the user `david` obtains a forwardable ticket while on

`jennifer`'s machine, he can log in to his own machine without having to get a new ticket (and thus authenticate himself again). (See XREF for an example of a forwardable ticket.) Compare a forwardable ticket to a *proxiable* ticket, below.

initial

An *initial* ticket is one that is issued directly, not based on a ticket-granting ticket. Some services, such as applications that change passwords, can require tickets to be marked *initial* in order to assure themselves that the client can demonstrate a knowledge of its secret key — because an *initial* ticket indicates that the client has recently authenticated itself (instead of relying on a ticket-granting ticket, which might have been around for a long time).

invalid

An *invalid* ticket is a postdated ticket that has not yet become usable. (See *postdated*, below.) It will be rejected by an application server until it becomes validated. To be validated, it must be presented to the KDC by the client in a TGS request, with the `VALIDATE` flag set, after its start time has passed.

postdatable/postdated

A *postdated* ticket is one that does not become valid until some specified time after its creation. Such a ticket is useful, for example, for batch jobs intended to be run late at night, since the ticket, if stolen, cannot be used until the batch job is to be run. When a *postdated* ticket is issued, it is issued as *invalid* and remains that way until: its start time has passed, and the client requests validation by the KDC. (See *invalid*, above.) A *postdated* ticket is normally valid until the expiration time of the ticket-granting ticket; however, if it is marked *renewable*, its lifetime is normally set to be equal to the duration of the full life of the ticket-granting ticket. See *renewable*, below.

proxiable/proxy

At times it can be necessary for a principal to allow a service to perform an operation on its behalf. (An example might be when a principal requests a service to run a print job on a third host.) The service must be able to take on the identity of the client, but need only do so for that

single operation. In that case, the server is said to be acting as a *proxy* for the client. The principal name of the proxy must be specified when the ticket is created.

A *proxiable* ticket is similar to a *forwardable* ticket, except that it is valid only for a single service, whereas a *forwardable* ticket grants the service the complete use of the client's identity. A *forwardable* ticket can therefore be thought of as a sort of super-proxy.

renewable

Because it is a security risk to have tickets with very long lives, tickets can be designated as *renewable*. A *renewable* ticket has two expiration times: the time at which the current instance of the ticket expires, and the maximum lifetime for any ticket. If a client wants to continue to use a ticket, it renews it before the first expiration occurs. For example, a ticket can be valid for one hour, with all tickets having a maximum lifetime of ten hours. If the client holding the ticket wants to keep it for more than an hour, the client must renew it within that hour. When a ticket reaches the maximum ticket lifetime (10 hours), it automatically expires and cannot be renewed.

For information on how to view tickets to see what their attributes are, see "How to View Tickets" on page 395.

Ticket Lifetimes

Any time a principal obtains a ticket, including a ticket–granting ticket, the ticket's lifetime is set as the smallest of the following lifetime values:

- The lifetime value specified by the `–l` option of `kinit`, if `kinit` is used to get the ticket.
- The maximum lifetime value (`max_life`) specified in the `kdc.conf` file.
- The maximum lifetime value specified in the Kerberos database for the service principal providing the ticket. (In the case of `kinit`, the service principal is `krbtgt/`*realm*.)
- The maximum lifetime value specified in the Kerberos database for the user principal requesting the ticket.

The following figure shows how a TGT's lifetime is determined and illustrates where the four lifetime values come from. Even though the figure shows how a TGT's

lifetime is determined, basically the same thing happens when any principal obtains a ticket. The only differences are that `kinit` doesn't provide a lifetime value, and the service principal providing the ticket provides a maximum lifetime value (instead of the `krbtgt/`*realm* principal).

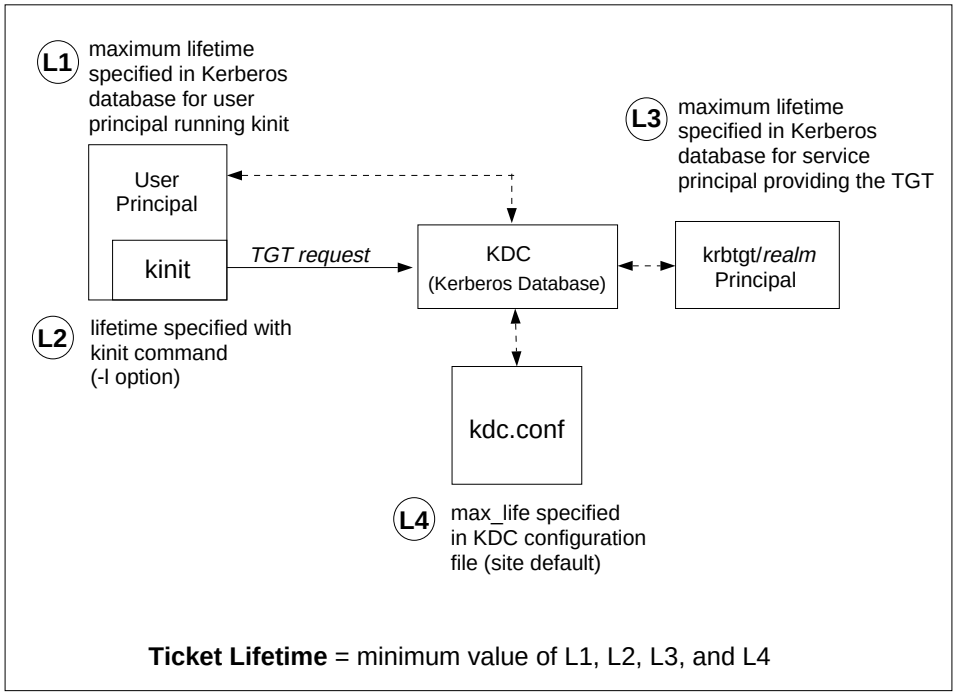


Figure 23–1 How a TGT's Lifetime Is Determined

The *renewable* ticket lifetime is also determined from the minimum of four values, but renewable lifetime values are used instead:

- The renewable lifetime value specified by the `–r` option of `kinit`, if `kinit` is used to obtain or renew the ticket
- The maximum renewable lifetime value (`max_renewable_life`) specified in the `kdc.conf` file
- The maximum lifetime renewable value specified in the Kerberos database for the service principal providing the ticket (in the case of `kinit`, the service principal is `krbtgt/`*realm*)
- The maximum lifetime renewable value specified in the Kerberos database for the user principal requesting the ticket

Principal Names

Each ticket is identified by a principal name. The principal name can identify a user or a service. Here are examples of several of the principal names.

TABLE 23–4 Examples of Principal Names

Principal Name	Description
`root/boston.acme.com@ACME.COM`	A principal associated with the `root` account on an NFS client. This is called a `root` principal and is needed for authenticated NFS-mounting to succeed.
`host/boston.acme.com@ACME.COM`	A principal used by the Kerberized applications (`klist` for example) or services (such as the NFS service).
username`@ACME.COM`	A principal for a user.
username`/admin@ACME.COM`	An `admin` principal that can be used to administer the KDC database.
`nfs/boston.acme.com@ACME.COM`	A principal used by the `nfs` service. This can be used instead of a `host` principal.

How the Authentication System Works

Applications allow you to log on to a remote system if you can provide a ticket that proves your identity and a matching session key. The session key contains information that is specific to the user and the service being accessed. A ticket and session key are created by the KDC for all users when they first log in. The ticket and matching session key form a credential. While using multiple networking services, a user can gather many credentials. The user needs to have a credential for each service running on a particular server. For instance, access to the `ftp` service on a server named `boston` requires one credential, and access to the `ftp` service on another server requires its own credential.

The process of creating and storing the credentials is transparent. Credentials are created by the KDC that sends the credential to the requestor. When received, the credential is stored in a credential cache.

Gaining Access to a Service Using SEAM

In order for a user to access a specific service on a specific server, the user must obtain two things. The first is a credential for the ticket-granting service (known as the TGT). Once the ticket-granting service has decrypted this credential, the service creates a second credential for the server for which the user requests access. This second credential can then be used to request access to the service on the server. After the server has successfully decrypted the second credential, the user is given access. This process is described in more detail below, and in the figures that follow.

Obtaining a Credential for the Ticket-Granting Service

1. To start the authentication process, the client sends a request to the authentication server for a specific user principal. This request is sent without encryption. There is no secure information included in the request, so it is not necessary to use encryption.
2. When the request is received by the authentication service, the principal name of the user is looked up in the KDC database. If a principal matches, the authentication service obtains the private key for that principal. The authentication service then generates a session key to be used by the client and the ticket-granting service (call it session key 1) and a ticket for the ticket-granting service (ticket 1). This ticket is also known as the ticket-granting ticket (TGT). Both the session key and the ticket are encrypted using the user's private key, and the information is sent back to the client.
3. The client uses this information to decrypt session key 1 and ticket 1, using the private key for the user principal. Since the private key should only be known by the user and the KDC database, the information in the packet should be safe. The client stores the information in the credentials cache.

Normally during this process, a user is prompted for a password. If the password entered is the same as the one used to build the private key stored in the KDC database, then the client can successfully decrypt the information sent by the authentication service. Now the client has a credential to be used with the ticket-granting service. The client is ready to request a credential for a server.

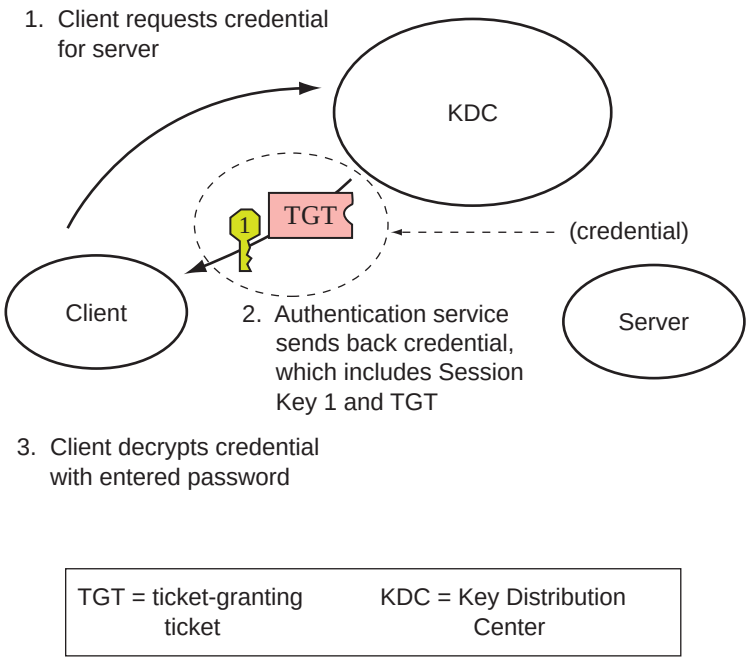


Figure 23–2 Obtaining a Credential for the Ticket-Granting Service

Obtaining a Credential for a Server

1. To request access to a specific server, a client must first have obtained a credential for that server from the authentication service (see “Obtaining a Credential for the Ticket-Granting Service” on page 409). The client then sends a request to the ticket-granting service, which includes the service principal name, ticket 1, and an authenticator encrypted with session key 1. Ticket 1 was originally encrypted by the authentication service using the service key of the ticket-granting service.
2. Because the service key of the ticket-granting service is known to the ticket-granting service, ticket 1 can be decrypted. The information included in ticket 1 includes session key 1, so the ticket-granting service can decrypt the authenticator. At this point, the user principal is authenticated with the ticket-granting service.
3. Once the authentication is successful, the ticket-granting service generates a session key for the user principal and the server (session key 2) and a ticket for the server (ticket 2). Session key 2 and ticket 2 are then encrypted using session key 1. Since session key 1 is known only to the client and the ticket-granting service, this information is secure and can be safely set over the net.

4. When the client receives this information packet, it decrypts the information using session key 1, which it had stored in the credential cache. The client has obtained a credential to be used with the server. Now the client is ready to request access to a particular service on that server.

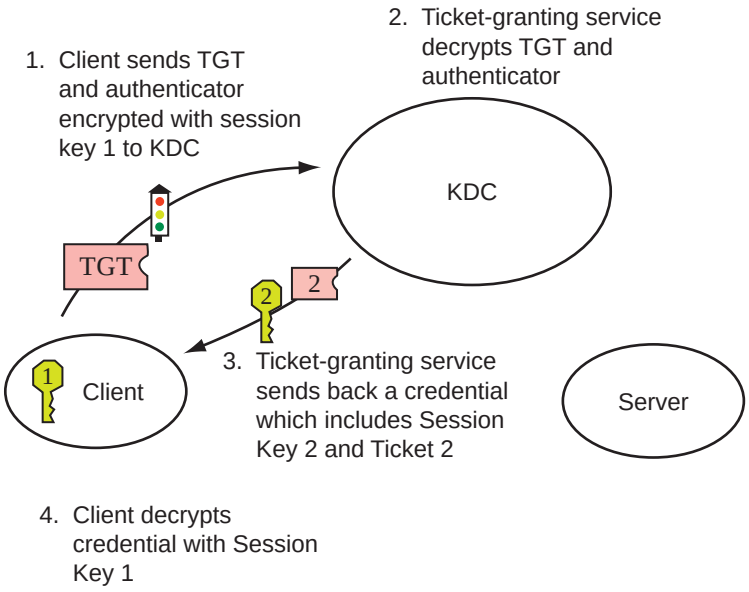


Figure 23–3 Obtaining a Credential for a Server

Obtaining Access to a Specific Service

1. To request access to a specific service, the client must first have obtained a credential for the ticket-granting service from the authentication server, and a server credential from the ticket-granting service (see “Obtaining a Credential for the Ticket-Granting Service” on page 409 and “Obtaining a Credential for a Server” on page 410). The client can send a request to the server including ticket 2 and another authenticator. The authenticator is encrypted using session key 2.
2. Ticket 2 was encrypted by the ticket-granting service with the service key for the service. Since the service key is known by the service principal, the service can decrypt ticket 2 and get session key 2. Session key 2 can then be used to decrypt the authenticator. If the authenticator is successfully decrypted, the client is given access to the service.

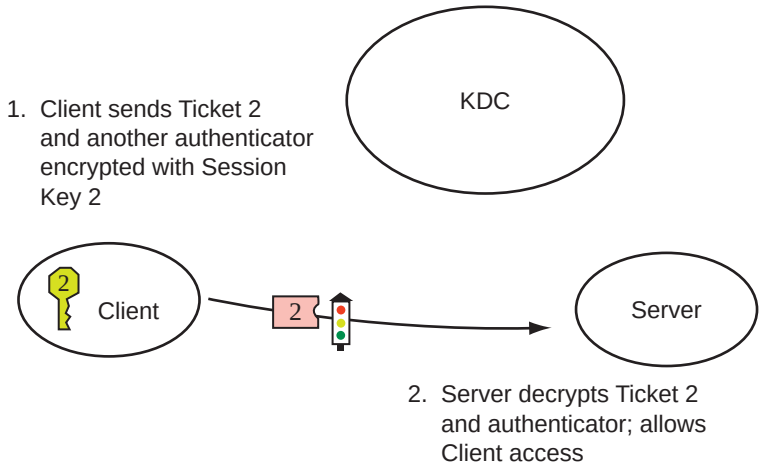


Figure 23–4 Obtaining Access to a Specific Service

Using the `gsscred` Table

The `gsscred` table is used by an NFS server when the server is trying to identify a SEAM user. The NFS services use UNIX IDs to identify users and these IDs are not part of a user principal or credential. The `gsscred` table provides a mapping from UNIX UIDs (from the password file) to principal names. The table must be created and administered after the KDC database is populated.

When a client request comes in, the NFS services try to map the principal name to a UNIX ID. If the mapping fails, the `gsscred` table is consulted. With the `kerberos_v5` mechanism, a `root/hostname` principal is automatically mapped to UID 0, and the `gsscred` table is not consulted. This means that there is no way to do special remappings of `root` through the `gsscred` table.

Which Mechanism to Select for the `gsscred` Table

Choosing the correct mechanism for the `gsscred` table depends on several factors.

- Are you interested in improving the lookup time?
- Are you interested in increasing data access security?
- Do you need to build the file quickly?

This is a list of all of the back-end mechanisms that can be selected along with a description of advantages of the mechanism.

files	The `gsscred` table is stored on a file system. A local file system that is not shared provides the most secure back-end, since no transmissions are done over the net after the table is created. This version of the file builds the quickest.
xfn_files	The `gsscred` table is stored within the `/var/fn` file system. This file system can be shared or not. All xfn files take a long time to build.
xfn_nis	The `gsscred` table is stored within the NIS namespace. The lookups in this file system are not secure. All xfn files take a long time to build.
xfn_nisplus	The `gsscred` table is stored within the NIS+ namespace. The lookups in this file system are not secure. All xfn files take a long time to build.
xfn	The `gsscred` table is stored within the default system for xfn. All xfn files take a long time to build.

For the `files` back-end mechanism, the initial lookup can be slow. For the other mechanisms, the initial lookup can be faster using a name service. For all of the mechanisms, after the data is cached the retrieval time should be about the same.

CHAPTER 24

Using Automated Security Enhancement Tool (Tasks)

This chapter describes how to use the Automated Security Enhancement Tool (ASET) to monitor or restrict access to system files and directories.

This is a list of step-by-step instructions in this chapter.

- "How to Run ASET Interactively" on page 435
- "How to Run ASET Periodically" on page 436
- "How to Stop Running ASET Periodically" on page 437
- "How to Collect ASET Reports on a Server" on page 437

Automated Security Enhancement Tool (ASET)

SunOS 5.8 system software includes the Automated Security Enhancement Tool (ASET). ASET helps you monitor and control system security by automatically performing tasks that you would otherwise do manually.

The ASET security package provides automated administration tools that enable you to control and monitor your system's security. You specify a security level—low, medium, or high—at which ASET will run. At each higher level, ASET's file-control functions increase to reduce file access and tighten your system security.

There are seven tasks involved with ASET, each performing specific checks and adjustments to system files. The ASET tasks tighten file permissions, check the contents of critical system files for security weaknesses, and monitor crucial areas. ASET can safeguard a network by applying the basic requirements of a firewall

system to a system that serves as a gateway system. (See “Firewall Setup” on page 420.)

ASET uses master files for configuration. Master files, reports, and other ASET files are in the `/usr/aset` directory. These files can be changed to suit the particular requirements of your site.

Each task generates a report noting detected security weaknesses and changes the task has made to the system files. When run at the highest security level, ASET will attempt to modify all system security weaknesses. If it cannot correct a potential security problem, ASET reports the existence of the problem.

You can initiate an ASET session by using the `/usr/aset` command interactively, or you can also set up ASET to run periodically by putting an entry into the `crontab` file.

ASET tasks are disk-intensive and can interfere with regular activities. To minimize the impact on system performance, schedule ASET to run when system activity level is lowest, for example, once every 24 or 48 hours at midnight.

ASET Security Levels

ASET can be set to operate at one of three security levels: low, medium, or high. At each higher level, ASET’s file-control functions increase to reduce file access and heighten system security. These functions range from monitoring system security without limiting users’ file access, to increasingly tightening access permissions until the system is fully secured.

The three levels are outlined in the table below.

Security Level	This Level ...
Low Security	Ensures that attributes of system files are set to standard release values. ASET performs several checks and reports potential security weaknesses. At this level, ASET takes no action and does not affect system services.
Medium Security	Provides adequate security control for most environments. ASET modifies some of the settings of system files and parameters, restricting system access to reduce the risks from security attacks. ASET reports security weaknesses and any modifications it makes to restrict access. At this level, ASET does not affect system services.
High Security	Renders a highly secure system. ASET adjusts many system files and parameter settings to minimum access permissions. Most system applications and commands continue to function normally, but at this level, security considerations take precedence over other system behavior.

Note - ASET does not change the permissions of a file to make it less secure, unless you downgrade the security level or intentionally revert the system to the settings that existed prior to running ASET.

ASET Tasks

This section discusses what ASET does. You should understand each ASET task—what its objectives are, what operations it performs, and what system components it affects—to interpret and use the reports effectively.

ASET report files contain messages that describe as specifically as possible any problems discovered by each ASET task. These messages can help you diagnose and correct these problems. However, successful use of ASET assumes that you possess a general understanding of system administration and system components. If you are a new administrator, you can refer to other SunOS 5.8 system administration documentation and related manual pages to prepare yourself for ASET administration.

The `taskstat` utility identifies the tasks that have been completed and the ones that are still running. Each completed task produces a report file. For a complete description of the `taskstat` utility, refer to `taskstat`(1M).

System Files Permissions Verification

This task sets the permissions on system files to the security level you designate. It is run when the system is installed. If you decide later to alter the previously established levels, run this task again. At low security, the permissions are set to values that are appropriate for an open information-sharing environment. At medium security, the permissions are tightened to produce adequate security for most environments. At high security, they are tightened to severely restrict access.

Any modifications that this task makes to system files permissions or parameter settings are reported in the `tune.rpt` file. “Tune Files” on page 433 shows an example of the files that ASET consults when setting permissions.

System Files Checks

This task examines system files and compares each one with a description of that file listed in a master file. The master file is created the first time ASET runs this task. The master file contains the system file settings enforced by `checklist` for the specified security level.

A list of directories whose files are to be checked is defined for each security level. You can use the default list, or you can modify it, specifying different directories for each level.

For each file, the following criteria are checked:

- Owner and group
- Permission bits
- Size and checksum
- Number of links
- Last modification time

Any discrepancies found are reported in the `cklist.rpt` file. This file contains the results of comparing system file size, permission, and checksum values to the master file.

User/Group Checks

This task checks the consistency and integrity of user accounts and groups as defined in the `passwd` and `group` files. It checks the local, and NIS or NIS+ password files. NIS+ password file problems are reported but not corrected. This task checks for the following violations:

- Duplicate names or IDs
- Entries in incorrect format
- Accounts without a password

- Invalid login directories
- The nobody account
- Null group password
- A plus sign (+) in the /etc/passwd file on an NIS (or NIS+) server

Discrepancies are reported in the usrgrp.rpt file.

System Configuration Files Check

During this task, ASET checks various system tables, most of which are in the /etc directory. These files are:

- /etc/default/login
- /etc/hosts.equiv
- /etc/inetd.conf
- /etc/aliases
- /var/adm/utmpx
- /.rhosts
- /etc/vfstab
- /etc/dfs/dfstab
- /etc/ftpusers

ASET performs various checks and modifications on these files, and reports all problems in the sysconf.rpt file.

Environment Check

This task checks how the PATH and UMASK environment variables are set for root, and other users, in the /.profile, /.login, and /.cshrc files.

The results of checking the environment for security are reported in the env.rpt file.

eeprom Check

This task checks the value of the eeprom security parameter to ensure that it is set to the appropriate security level. You can set the eeprom security parameter to none, command, or full.

ASET does not change this setting, but reports its recommendations in the eeprom.rpt file.

Firewall Setup

This task ensures that the system can be safely used as a network relay. It protects an internal network from external public networks by setting up a dedicated system as a firewall, which is described in “Firewall Systems” on page 290. The firewall system separates two networks, each of which approaches the other as untrusted. The firewall setup task disables the forwarding of Internet Protocol (IP) packets and hides routing information from the external network.

The firewall task runs at all security levels, but takes action only at the highest level. If you want to run ASET at high security, but find that your system does not require firewall protection, you can eliminate the firewall task by editing the `asetenv` file.

Any changes made are reported in the `firewall.rpt` file.

ASET Execution Log

ASET generates an execution log whether it runs interactively or in the background. By default, ASET generates the log file on standard output. The execution log confirms that ASET ran at the designated time, and also contains any execution error messages. The `aset -n` command directs the log to be delivered by electronic mail to a designated user. For a complete list of ASET options, refer to `aset`(1M).

Example of an ASET Execution Log File

```
ASET running at security level low

Machine=example; Current time = 0325_08:00


aset: Using /usr/aset as working directory

Executing task list...
        firewall
        env
        sysconfig
        usrgrp
        tune
        cklist
        eeprom
All tasks executed. Some background tasks may still be running.

Run /usr/aset/util/taskstat to check their status:
     $/usr/aset/util/taskstat     aset_dir
Where aset_dir is ASET's operating directory, currently=/usr/aset

When the tasks complete, the reports can be found in:
     /usr/aset/reports/latest/*.rpt
```

(continued)

```
You can view them by:
more /usr/aset/reports/latest/*.rpt
```

The log first shows the system and time that ASET was run. Then it lists each task as it is started.

ASET invokes a background process for each of these tasks, which are described in “ASET Tasks” on page 417. The task is listed in the execution log when it starts; this does not indicate that it has been completed. To check the status of the background tasks, use the `taskstat` utility.

ASET Reports

All report files generated from ASET tasks are in subdirectories under the `/usr/aset/reports` directory. This section describes the structure of the `/usr/aset/reports` directory, and provides guidelines on managing the report files.

ASET places the report files in subdirectories that are named to reflect the time and date when the reports are generated. This enables you to keep an orderly trail of records documenting the system status as it varies between ASET executions. You can monitor and compare these reports to determine the soundness of your system’s security.

The figure below shows an example of the `reports` directory structure.

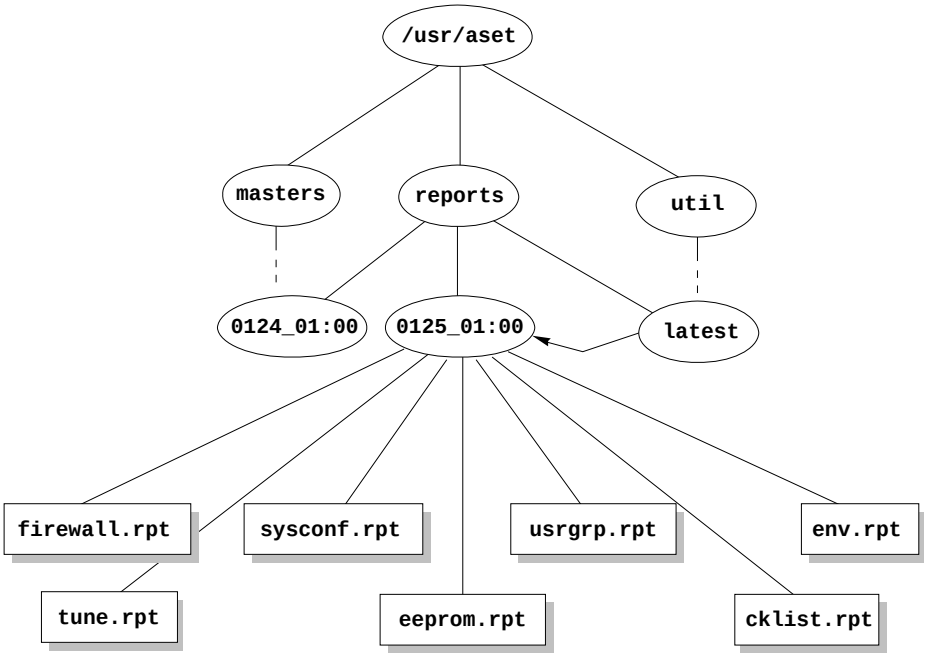


Figure 24–1 ASET `reports` Directory Structure

Two report subdirectories are shown in this example:

- `0124_01:00`
- `0125_01:00`

The subdirectory names indicate the date and time the reports were generated. Each report subdirectory name has the following format:

monthdate_hour:minute

where *month*, *date*, *hour*, and *minute* are all two-digit numbers. For example, `0125_01:00` represents January 25, at 1 a.m.

Each of the two report subdirectories contains a collection of reports generated from one execution of ASET.

The `latest` directory is a symbolic link that always points to the subdirectory that contains the latest reports. Therefore, to look at the latest reports that ASET has generated, you can go to the `/usr/aset/reports/latest` directory. There is a report file in this directory for each task that ASET performed during its most recent execution.

Format of ASET Report Files

Each report file is named after the task that generates it. See the table below for a list of tasks and their reports.

TABLE 24–1 ASET Tasks and Resulting Reports

Tasks	Report
System file permissions tuning (`tune`)	`tune.rpt`
System files checklist (`cklist`)	`cklist.rpt`
User/group checks (`usrgrp`)	`usrgrp.rpt`
System configuration files check (`sysconf`)	`sysconf.rpt`
Environment check (`env`)	`env.rpt`
`eeprom` check (`eeprom`)	`eeprom.rpt`
Firewall setup (`firewall`)	`firewall.rpt`

Within each report file, messages are bracketed by a beginning and an ending banner line. Sometimes a task terminates prematurely; for example, when a component of ASET is accidentally removed or damaged. In most cases, the report file will contain a message near the end that indicates the reason for the premature exit.

The following is a sample report file, `usrgrp.rpt`.

```
*** Begin User and Group Checking ***

Checking /etc/passwd ...
Warning! Password file, line 10, no passwd
:sync::1:1::/:/bin/sync
..end user check; starting group check ...
Checking /etc/group...
*** End User And group Checking ***
```

Examining ASET Report Files

After initially running or reconfiguring ASET, you should examine the report files closely. (Reconfiguration includes modifying the `asetenv` file or the master files in the `masters` subdirectory, or changing the security level at which ASET operates.) The reports record any errors introduced when you reconfigured. By watching the reports closely, you can react to, and solve, problems as they arise.

Comparing ASET Report Files

After you monitor the report files for a period during which there are no configuration changes or system updates, you might find that the content of the reports begin to stabilize and that it contains little, if any, unexpected information. You can use the `diff` utility to compare reports.

ASET Master Files

ASET's master files, `tune.high`, `tune.low`, `tune.med`, and `uid_aliases`, are located in the `/usr/aset/masters` directory. ASET uses the master files to define security levels.

Tune Files

The `tune.low`, `tune.med`, and `tune.high` master files define the available ASET security levels. They specify the attributes of system files at each level and are used for comparison and reference purposes.

The `uid_aliases` File

The `uid_aliases` file contains a list of multiple user accounts sharing the same ID. Normally, ASET warns about such multiple user accounts because this practice lessens accountability. You can allow for exceptions to this rule by listing the exceptions in the `uid_aliases` file. ASET does not report entries in the `passwd` file with duplicate user IDs if these entries are specified in the `uid_aliases` file.

Avoid having multiple user accounts (password entries) share the same user ID. You should consider other methods of achieving your objective. For example, if you intend for several users to share a set of permissions, you could create a group account. Sharing user IDs should be your last resort, used only when absolutely necessary and when other methods will not accomplish your objectives.

You can use the `UID_ALIASES` environment variable to specify an alternate aliases file. The default is `/usr/aset/masters/uid_aliases`.

The Checklist Files

The master files used by the systems files checklist are generated when you first execute ASET, or when you run ASET after you change the security level.

The files checked by this task are defined by the following environment variables:

- `CKLISTPATH_LOW`
- `CKLISTPATH_MED`

- CKLISTPATH_HIGH

ASET Environment File (asetenv)

The environment file, asetenv, contains a list of variables that affect ASET tasks. These variables can be changed to modify ASET operation.

Configuring ASET

This section discusses how ASET is configured and the environment under which it operates.

ASET requires minimum administration and configuration, and in most cases, you can run it with the default values. You can, however, fine-tune some of the parameters that affect the operation and behavior of ASET to maximize its benefit. Before changing the default values, you should understand how ASET works, and how it affects the components of your system.

ASET relies on four configuration files to control behavior of its tasks:

- /usr/aset/asetenv
- /usr/aset/masters/tune.low
- /usr/aset/masters/tune.med
- /usr/aset/masters/tune.high

Modifying the Environment File (asetenv)

The /usr/aset/asetenv file has two main sections:

- A user-configurable parameters section
- An internal environment variables section

You can alter the user-configurable parameters section. However, the settings in the internal environment variables section are for internal use only and should not be modified.

You can edit the entries in the user-configurable parameters section to:

- Choose which tasks to run
- Specify directories for checklist task
- Schedule ASET execution
- Specify an aliases file
- Extend checks to NIS+ tables

Choose Which Tasks to Run: TASKS

Each of the tasks ASET performs monitors a particular area of system security. In most system environments, all the tasks are necessary to provide balanced security coverage. However, you might decide to eliminate one or more of the tasks.

For example, the firewall task runs at all security levels, but takes action only at the high security level. You might want to run ASET at the high-security level, but do not require firewall protection.

It's possible to set up ASET to run at the high level without the firewall feature by editing the TASKS list of environment variables in the `asetenv` file. By default, the TASKS list contains all of the ASET tasks. (An example is shown below.) To delete a task, remove the task setting from the file. In this case, you would delete the `firewall` environment variable from the list. The next time ASET runs, the excluded task will not be performed.

```
TASKS=''env sysconfig usrgrp tune cklist eeprom firewall''
```

Specify Directories for Checklist Task: CKLISTPATH

The system files check checks attributes of files in selected system directories. You define which directories to check by using these checklist path environment variables:

- CKLISTPATH_LOW
- CKLISTPATH_MED
- CKLISTPATH_HIGH

The CKLISTPATH_LOW variable defines the directories to be checked at the low security level. CKLISTPATH_MED and CKLISTPATH_HIGH environment variables function similarly for the medium and high security levels.

The directory list defined by a variable at a lower security level should be a subset of the directory list defined at the next higher level. For example, all directories specified for CKLISTPATH_LOW should be included in CKLISTPATH_MED, and all the directories specified for CKLISTPATH_MED should be included in CKLISTPATH_HIGH.

Checks performed on these directories are not recursive; ASET only checks those directories explicitly listed in the variable. It does not check their subdirectories.

You can edit these variable definitions to add or delete directories that you want ASET to check. Note that these checklists are useful only for system files that do not normally change from day to day. A user's home directory, for example, is generally too dynamic to be a candidate for a checklist.

Schedule ASET Execution: PERIODIC_SCHEDULE

When you start ASET, you can start it interactively, or use the –p option to request that the ASET tasks run at a scheduled time and period. You can run ASET periodically, at a time when system demand is light. For example, ASET consults PERIODIC_SCHEDULE to determine how frequently to execute the ASET tasks, and at what time to run them. For detailed instructions about setting up ASET to run periodically, see “How to Run ASET Periodically” on page 436.

The format of PERIODIC_SCHEDULE follows the format of crontab entries. See crontab(1) for complete information.

Specify an Aliases File: UID_ALIASES

The UID_ALIASES variable specifies an aliases file that lists shared user IDs. The default is /usr/aset/masters/uid_aliases.

Extend Checks to NIS+ Tables: YPCHECK

The YPCHECK environment variable specifies whether ASET should also check system configuration file tables. YPCHECK is a Boolean variable; you can specify only true or false for it. The default value is false, disabling NIS+ table checking.

To understand how this variable works, consider its effect on the passwd file. When this variable is set to false, ASET checks the local passwd file. When it is set to true, the task also checks the NIS+ passwd file for the domain of the system.

Note - Although ASET automatically repairs the local tables, it only reports potential problems in the NIS+ tables; it does not change them.

Modifying the Tune Files

ASET uses the three master tune files, tune.low, tune.med, and tune.high, are used by ASET to ease or tighten access to critical system files. These master files are located in the /usr/aset/masters directory, and they can be modified to suit your environment. For additional information, see “Tune Files” on page 433.

The tune.low file sets permissions to values appropriate for default system settings. The tune.med file further restricts these permissions and includes entries not present in tune.low. The tune.high file restricts permissions even further.

Note - Modify settings in the tune file by adding or deleting file entries. Setting a permission to a less restrictive value than the current setting has no effect; the ASET tasks do not relax permissions unless you downgrade your system security to a lower level.

Restoring System Files Modified by ASET

When ASET is executed for the first time, it saves and archives the original system files. The `aset.restore` utility reinstates these files. It also deschedules ASET, if it is currently scheduled for periodic execution. The `aset.restore` utility is located in `/usr/aset`, the ASET operating directory.

Changes made to system files are lost when you run `aset.restore`.

You should use `aset.restore`:

- When you want to remove ASET changes and restore the original system. If you want to deactivate ASET permanently, you can remove it from `cron` scheduling if the `aset` command had been added to root's `crontab` previously. For directions on how to use `cron` to remove automatic execution, see "How to Stop Running ASET Periodically" on page 437.
- After a brief period of experimenting with ASET, to restore the original system state.
- When some major system functionality is not working properly and you suspect that ASET is causing the problem.

Network Operation Using the NFS System

Generally, ASET is used in standalone mode, even on a system that is part of a network. As system administrator for your standalone system, you are responsible for the security of your system and for running and managing ASET to protect your system.

You can also use ASET in the NFS distributed environment. As a network administrator, you are responsible for installing, running, and managing various administrative tasks for all of your clients. To facilitate ASET management across several client systems, you can make configuration changes that are applied globally to all clients, eliminating the need for you to log in to each system to repeat the process.

When deciding how to set up ASET on your networked systems, you should consider how much you want users to control security on their own systems, and how much you want to centralize responsibility for security control.

Providing a Global Configuration for Each Security Level

A case might arise where you want to set up more than one network configuration. For example, you might want to set up one configuration for clients that are designated with low security level, another configuration for those with medium level, and yet another one with high level.

If you need to create a separate ASET network configuration for each security level, you can create three ASET configurations on the server—one for each level. You would export each configuration to the clients with the appropriate security level. Some ASET components that are common to all three configurations could be shared using links.

Collecting ASET Reports

Not only can you centralize the ASET components on a server to be accessed by clients with or without superuser privilege, but you can also set up a central directory on a server to collect all reports produced by tasks running on various clients. For instructions on setting up a collection mechanism, see “How to Collect ASET Reports on a Server” on page 437.

Setting up the collection of reports on a server allows you to review reports for all clients from one location. You can use this method whether a client has superuser privilege or not. Alternatively, you can leave the reports directory on the local system when you want users to monitor their own ASET reports.

ASET Environment Variables

The table below lists the ASET environment variables and the values that they specify.

TABLE 24–2 ASET Environment Variables and Their Meanings

Environment Variable	Specifies ...
`ASETDIR` (See below)	ASET working directory
`ASETSECLEVEL` (See below)	Security level
`PERIODIC_SCHEDULE`	Periodic schedule
`TASKS`	Tasks to run
`UID_ALIASES`	Aliases file
`YPCHECK`	Extends check to NIS and NIS+
`CKLISTPATH_LOW`	Directory lists for low security

TABLE 24–2 ASET Environment Variables and Their Meanings *(continued)*

Environment Variable	Specifies ...
CKLISTPATH_MED	Directory list for medium security
CKLISTPATH_HIGH	Directory list for high security

The environment variables listed below are found in the /usr/aset/asetenv file. The ASETDIR and ASETSECLEVEL variables are optional and can be set only through the shell by using the aset command. The other environment variables can be set by editing the file. The variables are described below.

ASETDIR Variable

ASETDIR specifies an ASET working directory.

From the C shell, type:

```
% setenv ASETDIR pathname
```

From the Bourne shell or the Korn shell, type:

```
$ ASETDIR=pathname
$ export ASETDIR
```

Set *pathname* to the full path name of the ASET working directory.

ASETSECLEVEL Variable

The ASETSECLEVEL variable specifies a security level at which ASET tasks are executed.

From the C shell, type:

```
% setenv ASETSECLEVEL level
```

From the Bourne shell or the Korn shell, type:

```
$ ASETSECLEVEL=level
export ASETSECLEVEL
```

In the above commands, *level* can be set to one of the following:

`low`	Low security level
`med`	Medium security level
`high`	High security level

PERIODIC_SCHEDULE Variable

The value of `PERIODIC_SCHEDULE` follows the same format as the `crontab` file. Specify the variable value as a string of five fields enclosed in double quotation marks, each field separated by a space:

```
"minutes hours day-of-month month day-of-week"
```

TABLE 24–3 `Periodic_Schedule` Variable Values

Variable	Value
minutes hours	Specifies start time in number of minutes (0-59) after the hour and the hour (0-23)
day-of-month	Specifies the day of the month when ASET should be run, using values from 1 through 31
month	Specifies the month of the year when ASET should be run, using values from 1 through 12
day-of-week	Specifies the day of the week when ASET should be run, using values from 0 through 6; Sunday is day 0 in this scheme

The following rules apply:

- You can specify a list of values, each delimited by a comma, for any field.
- You can specify a value as a number, or you can specify it as a range; that is, a pair of numbers joined by a hyphen. A range states that the ASET tasks should be executed for every time included in the range.
- You can specify an asterisk (*) as the value of any field. An asterisk specifies all possible values of the field, inclusive.

The default entry for PERIODIC_SCHEDULE variable causes ASET to execute at 12:00 midnight every day:

```
PERIODIC_SCHEDULE=''0 0 * * *''
```

TASKS Variable

The TASKS variable lists the tasks that ASET performs. The default is to list all seven tasks:

```
TASKS=''env sysconfig usrgrp tune cklist eeprom firewall''
```

UID_ALIASES Variable

The UID_ALIASES variable specifies an aliases file. If present, ASET consults this file for a list of permitted multiple aliases. The format is UID_ALIASES=*pathname*. *pathname* is the full path name of the aliases file.

The default is:

```
UID_ALIASES=${ASETDIR}/masters/uid_aliases
```

YPCHECK Variable

The YPCHECK variable extends the task of checking system tables to include NIS or NIS+ tables. It is a Boolean variable, which can be set to either true or false.

The default is false, confining checking to local system tables:

```
YPCHECK=false
```

CKLISTPATH_level Variable

The three checklist path variables list the directories to be checked by the checklist task. The following definitions of the variables are set by default; they illustrate the relationship between the variables at different levels:

```
CKLISTPATH_LOW=${ASETDIR}/tasks:${ASETDIR}/util:${ASETDIR}/masters:
/etc
CKLISTPATH_MED=${CKLISTPATH_LOW}:/usr/bin:/usr/ucb
CKLISTPATH_HIGH=${CKLISTPATH_MED}:/usr/lib:/sbin:/usr/sbin:/usr/ucblib
```

The values for the checklist path environment variables are similar to those of the shell path variables, in that they are lists of directory names separated by colons (:). You use an equal sign (=) to connect the variable name to its value.

ASET File Examples

This section has examples of some of the ASET files, including the tune files and the aliases file.

Tune Files

ASET maintains three tune files. The entry format in all three tune files are described in the table below.

TABLE 24–4 Tune File Entry Format

Entry	Description
pathname	The full path name of the file
mode	A five-digit number that represents the permission setting
owner	The owner of the file
group	The group owner of the file
type	The type of the file

The following rules apply:

- You can use regular shell wildcard characters, such as an asterisk (*) and a question mark (?), in the path name for multiple references. See `sh`(1) for more information.
- *mode* represents the least restrictive value. If the current setting is already more restrictive than the specified value, ASET does not loosen the permission settings. For example, if the specified value is 00777, the permission will remain unchanged, because 00777 is always less restrictive than whatever the current setting is.

  This is how ASET handles mode setting, unless the security level is being downgraded or you are removing ASET. When you decrease the security level from what it was for the previous execution, or when you want to restore the system files to the state they were in before ASET was first executed, ASET recognizes what you are doing and decreases the protection level.
- You must use names for *owner* and *group* instead of numeric IDs.
- You can use a question mark (?) in place of *owner*, *group*, and *type* to prevent ASET from changing the existing values of these parameters.

- *type* can be `symlink` (symbolic link), directory, or file (everything else).
- Higher security level tune files reset file permissions to be at least as restrictive as they are at lower levels. Also, at higher levels, additional files are added to the list.
- A file can match more than one tune file entry. For example, `etc/passwd` matches `etc/pass*` and `/etc/*` entries.
- Where two entries have different permissions, the file permission is set to the most restrictive value. In the following example, the permission of `/etc/passwd` will be set to 00755, which is the more restrictive of 00755 and 00770.

```
/etc/pass*    00755   ? ?  file
/etc/*        00770   ? ?  file
```

- If two entries have different *owner* or *group* designations, the last entry takes precedence. The following example shows the first few lines of the `tune.low` file.

```
/ 02755 root root directory
/bin 00777 root bin symlink
/sbin 02775 root sys directory
/usr/sbin 02775 root bin directory
/etc 02755 root sys directory
/etc/chroot 00777 bin bin symlink
```

Aliases File

An aliases file contains a list of aliases that share the same user ID.

Each entry is in this form:

uid=alias1=alias2=alias3=...

uid	Shared user ID.
aliasn	User account sharing the user ID.

For example, the following entry lists the user ID `0` being shared by `sysadm` and `root`:

`0=root=sysadm`

Running ASET

This section describes how to run ASET either interactively or periodically.

▼ How to Run ASET Interactively

1. **Become superuser.**

2. **Run ASET interactively by using the** `aset` **command.**

```
# /usr/aset/aset -l level -d pathname
```

level	Specifies the level of security. Valid values are `low`, `medium`, or `high`. The default setting is `low`. See "ASET Security Levels" on page 416 for detailed information about security levels.
pathname	Specifies the working directory for ASET. The default is `/usr/aset`.

3. **Verify ASET is running by viewing the ASET execution log that is displayed on the screen.**

   The execution log message identifies which tasks are being run.

Example—Running ASET Interactively

The following example runs ASET at low security with the default working directory.

```
# /usr/aset/aset -l low
======= ASET Execution Log =======

ASET running at security level low

Machine = jupiter; Current time = 0111_09:26

aset: Using /usr/aset as working directory

Executing task list ...
 firewall
 env
 sysconf
 usrgrp
```

(continued)

```
 tune
 cklist
 eeprom

All tasks executed. Some background tasks may still be running.

Run /usr/aset/util/taskstat to check their status:
 /usr/aset/util/taskstat [aset_dir]

where aset_dir is ASET's operating
directory,currently=/usr/aset.

When the tasks complete, the reports can be found in:
 /usr/aset/reports/latest/*.rpt

You can view them by:
 more /usr/aset/reports/latest/*.rpt
```

▼ How to Run ASET Periodically

1. **Become superuser.**

2. **If necessary, set up the time when you want ASET to run periodically.**

   You should have ASET run when system demand is light. The `PERIODIC_SCHEDULE` environment variable in the `/usr/aset/asetenv` file is used to set up the time for ASET to run periodically. By default, the time is set for midnight every 24 hours.

   If you want to set up a different time, edit the `PERIODIC_SCHEDULE` variable in the `/usr/aset/asetenv` file. See "`PERIODIC_SCHEDULE` Variable" on page 431 for detailed information about setting the `PERIODIC_SCHEDULE` variable.

3. **Add an entry to the `crontab` file using the `aset` command.**

   ```
   # /usr/aset/aset -p
   ```

`–p`	Inserts a line in the `crontab` file that starts ASET running at the time determined by the `PERIODIC_SCHEDULE` environment variable in the `/usr/aset/asetenv` file.

4. **Display the `crontab` entry to verify when ASET will run.**

```
# crontab -l root
```

▼ How to Stop Running ASET Periodically

1. **Become superuser.**
2. **Edit the** `crontab` **file.**

```
# crontab -e root
```

3. **Delete the ASET entry.**
4. **Save the changes and exit.**
5. **Display the** `crontab` **entry to verify the ASET entry is deleted.**

```
# crontab -l root
```

▼ How to Collect ASET Reports on a Server

1. **Become superuser.**
2. **Set up a directory on the server:**
   a. **Change to the** `/usr/aset` **directory.**

```
mars# cd /usr/aset
```

   b. **Create a *rptdir* directory.**

```
mars# mkdir rptdir
```

   c. **Change to the *rptdir* directory and create a *client_rpt* directory.**

```
mars# cd rptdir
mars# mkdir client_rpt
```

d. This creates a subdirectory (*client_rpt*) for a client. Repeat this step for each client whose reports you need to collect.

The following example creates the directory `all_reports`, and the subdirectories `pluto_rpt` and `neptune_rpt`.

```
mars# cd /usr/aset
mars# mkdir all_reports
mars# cd all_reports
mars# mkdir pluto_rpt
mars# mkdir neptune_rpt
```

3. Add the *client_rpt* directories to the `/etc/dfs/dfstab` file.

The directories should have read/write options.

For example, the following entries in `dfstab` are shared with read/write permissions.

```
share -F nfs -o rw=pluto /usr/aset/all_reports/pluto_rpt
share -F nfs -o rw=neptune /usr/aset/all_reports/neptune_rpt
```

4. Make the resources in the `dfstab` file available to the clients.

```
# shareall
```

5. On each client, mount the client subdirectory from the server at the mount point, `/usr/aset/masters/reports`.

```
# mount server:/usr/aset/client_rpt /usr/aset/masters/reports
```

6. Edit the `/etc/vfstab` file to mount the directory automatically at boot time.

The following sample entry in `/etc/vfstab` on `neptune` lists the directory to be mounted from `mars`, `/usr/aset/all_reports/neptune_rpt`, and the mount point on `neptune`, `/usr/aset/reports`. At boot time, the directories listed in `vfstab` are automatically mounted.

```
mars:/usr/aset/all_reports/neptune.rpt /usr/aset/reports nfs - yes
hard
```

Troubleshooting ASET Problems

This section documents the error messages generated by ASET.

ASET Error Messages

```
ASET failed: no mail program found.
```

Cause

ASET is directed to send the execution log to a user, but no mail program can be found.

Action

Install a mail program.

```
Usage: aset [-n user[@host]] in /bin/mail or /usr/ucb/mail.
Cannot decide current and previous security levels.
```

Cause

ASET cannot determine what the security levels are for the current and previous invocations.

Action

Ensure the current security level is set either through the command line option or the ASETSECLEVELenvironment variable. Also, ensure that the last line of ASETDIR/archives/asetseclevel.arch correctly reflects the previous security level. If these values are not set or are incorrect, specify them correctly.

```
ASET working directory undefined.
To specify, set ASETDIR environment variable or use command line
option -d.
ASET startup unsuccessful.
```

Cause

The ASET working (operating) directory is not defined, or defined incorrectly.

Action

Use the ASETDIR environment variable or the -d command line option to specify it correctly, and restart ASET.

```
ASET working directory $ASETDIR missing.
ASET startup unsuccessful.
```

Cause

The ASET working (operating) directory is not defined, or it is defined incorrectly. This might be because the ASETDIR variable or the -d command line option refers to a nonexistent directory.

Action

Ensure that the correct directory—that is, the directory containing the ASET directory hierarchy—is referred to correctly.

```
Cannot expand $ASETDIR to full pathname.
```

Cause

ASET cannot expand the directory name given by the ASETDIR variable or the -d command line option to a full path name.

Action

Ensure that the directory name is given correctly, and that it refers to an existing directory to which the user has access.

```
aset: invalid/undefined security level.
To specify, set ASETSECLEVEL environment variable or use command
line option -l, with argument= low/med/high.
```

Cause

The security level is not defined or it is defined incorrectly. Only the values `low`, `med`, or `high` are acceptable.

Action

Use the `ASETSECLEVEL` variable or the `-l` command line option to specify one of the three values.

```
ASET environment file asetenv not found in $ASETDIR.
ASET startup unsuccessful.
```

Cause

ASET cannot locate an `asetenv` file in its working directory.

Action

Ensure there is an `asetenv` file in ASET's working directory. See `asetenv`(4) for the details about this file.

```
filename doesn't exist or is not readable.
```

Cause

The file referred to by *filename* doesn't exist or is not readable. This can specifically occur when using the –u option where you can specify a file that contains a list of users whom you want to check.

Action

Ensure the argument to the –u option exists and is readable.

```
ASET task list TASKLIST undefined.
```

Cause

The ASET task list, which should be defined in the `asetenv` file, is not defined. This can mean that your `asetenv` file is bad.

Action

Examine your `asetenv` file. Ensure the task list is defined in the `User Configurable` section. Also check other parts of the file to ensure the file is intact. See `asetenv`(4) for the content of a good `asetenv` file.

```
ASET task list $TASKLIST missing.
ASET startup unsuccessful.
```

Cause

The ASET task list, which should be defined in the `asetenv` file, is not defined. This can mean that your `asetenv` file is bad.

Action

Examine your `asetenv` file. Ensure the task list is defined in the `User Configurable` section. Also check other parts of the file to ensure the file is intact. See `asetenv`(4) for the content of a good `asetenv` file.

```
Schedule undefined for periodic invocation.
No tasks executed or scheduled. Check asetenv file.
```

Cause

ASET scheduling is requested using the `–p` option, but the variable `PERIODIC_SCHEDULE` is undefined in the `asetenv` file.

Action

Check the `User Configurable` section of the `asetenv` file to ensure the variable is defined and is in proper format.

```
Warning! Duplicate ASET execution scheduled.
Check crontab file.
```

Cause

ASET is scheduled more than once. In other words, scheduling is requested while a schedule is already in effect. This is not necessarily an error if more than one schedule is indeed desired, just a warning that normally this is unnecessary since you should use the `crontab`(1) scheduling format if you want more than one schedule.

Action

Verify, through the `command`, that the correct schedule is in effect. Ensure that no unnecessary `crontab` entries for ASET are in place.

CHAPTER 25

Managing System Resources Topics

This section provides instructions for managing system resources in the Solaris environment. This section contains these chapters.

Chapter 26	Provides overview information about Solaris commands and utilities that help you manage system resources by using disk quotas, accounting programs, and `cron` and `at` commands.
Chapter 27	Provides step-by-step instructions for examining and changing system information.
Chapter 28	Provides step-by-step instructions for optimizing disk space by locating unused files and large directories.
Chapter 29	Provides step-by-step instructions for setting up and administering disk quotas.
Chapter 30	Provides step-by-step instructions for scheduling routine or one-time system events using `crontab` and `at` features.
Chapter 31	Provides step-by-step instructions for setting up and maintaining system accounting.
Chapter 32	Provides reference information for system accounting software.

CHAPTER 26

Managing System Resources (Overview)

This chapter contains overview information about miscellaneous features offered by the Solaris operating environment and other UNIX® software products to help you manage system resources by displaying general system information, monitoring disk space, setting disk quotas, using accounting programs, and scheduling `crontab` and `at` commands that automatically run routine commands.

This is a list of the overview information in this chapter.

- "Displaying and Changing System Information" on page 448
- "What Are Quotas?" on page 448
- "Executing Routine Tasks Automatically" on page 449
- "What is System Accounting?" on page 450

Where to Find System Resource Tasks

Use these references to find step-by-step instructions for managing system resources.

- Chapter 27
- Chapter 28
- Chapter 29
- Chapter 30
- Chapter 31

What's New in Managing System Resources?

In this Solaris release, pseudo terminals are allocated dynamically. This means it is unnecessary to set the `pt_cnt` variable in the `/etc/system` file to increase the number of pseudo terminals in the system.

Displaying and Changing System Information

Chapter 27 describes how to find general system information such as the Solaris release the system is running, the amount of memory on a system, and the amount of available disk space.

Setting a system's date and time and increasing some system resources are also covered in this chapter.

What Are Quotas?

Quotas enable system administrators to control the size of UFS file systems by limiting the amount of disk space and the number of inodes (which roughly corresponds to the number of files) that individual users can acquire. For this reason, quotas are especially useful on the file systems where user home directories reside. (As a rule, public and `/tmp` file systems probably wouldn't benefit as much from the establishment of quotas.)

Setting up quotas involves these general steps:

1. A series of commands prepares a file system to accept quotas, ensuring that quotas will be enforced each time the system is rebooted and the file system is mounted. Entries must be added to the `/etc/vfstab` file, and a `quotas` file must be created in the top-level directory of the file system.
2. After a quota is created for one user, it can be copied as a prototype to set up other user quotas.

3. Before quotas are actually turned on, another command checks for consistency by comparing the proposed quotas to the current disk usage making sure there are no conflicts.
4. Finally, a command turns the quotas on for one or more entire file systems.

These steps ensure that quotas are automatically activated on a file system each time it is mounted. See Chapter 29 for specific information about these procedures.

Once they are in place, quotas can be changed to adjust the amount of disk space or number of inodes that users can consume. Additionally, quotas can be added or removed as system needs change. See “Changing and Removing Quotas” on page 492 for instructions on how to change quotas, disable individual quotas, or remove quotas from file systems.

In addition, quota status can be monitored. Quota commands enable administrators to display information about quotas on a file system, or search for users who have exceeded their quotas. For procedures that describe how to use these commands, see “Checking Quotas” on page 489.

Executing Routine Tasks Automatically

Many routine system events can be set up to execute automatically. Some of these tasks should occur repetitively, at regular intervals. Other tasks need to run only once, perhaps during off hours such as evenings or weekends.

This section contains overview information about two commands, `crontab` and `at`, which enable you to schedule routine commands to execute automatically, avoiding peak hours or repeating commands according to a fixed schedule. `crontab` schedules repetitive commands, while `at` schedules commands that execute once.

Scheduling Repetitive Jobs: `crontab`

You can schedule routine system administration commands to execute daily, weekly, or monthly by using the `crontab` commands.

Daily `crontab` system administration tasks might include:

- Removing junk files more than a few days old from temporary directories
- Executing accounting summary commands
- Taking snapshots of the system by using `df` and `ps` commands
- Performing daily security monitoring
- Running system backups

Weekly `crontab` system administration tasks might include:

- Rebuilding the `catman` database for use by `man -k`
- Running `fsck -n` to list any disk problems

Monthly `crontab` system administration tasks might include:

- Listing files not used that month
- Producing monthly accounting reports

Additionally, users can schedule `crontab` commands to execute other routine system tasks, such as sending reminders and removing backup files.

For more information about scheduling `crontab` jobs, see Chapter 30.

Scheduling a Single Job: `at`

The `at` command allows you to schedule a job for execution at a later time. The job may consist of a single command or a script.

Like `crontab`, `at` allows you to schedule the automatic completion of routine commands. However, unlike `crontab` files, `at` files execute their commands once, and then are removed from their directory. Therefore, `at` is most useful for running simple commands or scripts that direct output into separate files for later examination.

Submitting an `at` job involves entering a command, following the `at` command syntax to specify options to schedule the time your job will be executed. For more information about submitting `at` jobs, see “`at` Command Description” on page 509.

The `at` command stores the command or script you entered, along with a copy of your current environment variable in the `/var/spool/cron/atjobs` directory. As a file name, your `at` job file is given a long number specifying its location in the `at` queue, followed by the `.a` extension, such as `793962000.a`.

The `cron` daemon periodically executes the `atrun` program, usually at 15-minute intervals. `atrun` then executes `at` jobs at their scheduled times. After your `at` job has been executed, its file is removed from the `atjobs` directory.

For more information on scheduling `at` jobs, see Chapter 27.

What is System Accounting?

The SunOS 5.8 system accounting software is a set of programs that enables you to collect and record data about user connect time, CPU time charged to processes, and disk usage. Once this data is collected, you can generate reports and charge fees for system usage.

The accounting programs can be used for:

- Monitoring system usage
- Troubleshooting
- Locating and correcting performance problems
- Maintaining system security

After they're set up, the system accounting programs run mostly on their own.

Accounting Components

The accounting software provides C language programs and shell scripts that organize data into summary files and reports. These programs reside in the `/usr/adm/acct` and `/usr/lib/acct` directories.

Daily accounting can help you do four types of auditing:

- Connect
- Process
- Disk
- Fee calculations

How Accounting Works

Setting up automatic accounting involves putting the accounting startup script into `crontab` files so it can be started automatically by `cron`.

The following is an overview of how accounting works.

1. Between system startup and shutdown, raw data about system use (such as user logins, running processes, and data storage) are collected in accounting files.
2. Periodically (usually once a day), the `/usr/lib/acct/runacct` program processes the various accounting files and produces both cumulative summary files and daily accounting reports. The daily reports are printed by the `/usr/lib/acct/prdaily` program.
3. Monthly, the administrator can process and print the cumulative summary files generated by `runacct` by executing the `monacct` program. The summary reports produced by `monacct` provide an efficient means for billing users on a monthly or other fiscal basis.

See Chapter 31 for instructions on setting up the accounting software. See Chapter 32 for reference information about the different accounting features.

CHAPTER 27

Examining and Changing System Information (Tasks)

This chapter describes tasks required to examine and change the most common system information. This is a list of the step-by-step instructions in this chapter.

- "How to Determine Whether a System Can Run the 64–bit Solaris Operating Environment" on page 454
- "How to Display General System Information (`uname`)" on page 457
- "How to Display a System's Host ID Number " on page 457
- "How to Display a System's Installed Memory" on page 458
- "How to Display the Date and Time" on page 458
- "How to Set Up an NTP Server" on page 460
- "How to Set Up an NTP Client" on page 460
- "How to Synchronize Date and Time From Another System" on page 461
- "How to Set a System's Date and Time Manually" on page 461
- "How to Set Up a Message of the Day" on page 462
- "How to Set the Number of Processes per User" on page 463
- "How to Increase Shared Memory Segments" on page 464

Using Commands to Display System Information

The following table describes commands that enable you to display general system information.

TABLE 27–1 Commands for Displaying System Information

Command	Enables You to Display a System's ...
`psrinfo`(1M)	Processor type
`isainfo`(1)	Supported applications and it reports the number of bits supported by *native* applications on the running system, which can be passed as a token to scripts
`showrev`(1M)	Hostname, host identification number, release, kernel architecture, application architecture, hardware provider, domain, and kernel version
`uname`(1)	Operating system name, release, and version; node name; hardware name; processor type
`hostid`(1)	Host ID number
`prtconf`(1M)	Installed memory
`date`(1)	Date and time

▼ How to Determine Whether a System Can Run the 64–bit Solaris Operating Environment

Currently, the only platform capable of supporting the 64–bit Solaris operating environment is an UltraSPARC system. You can verify whether a system is an UltraSPARC system by using the following command:

```
$ uname -m
sun4u
```

If the output of the `uname -m` command is `sun4u`, then the machine is an UltraSPARC system.

If you are running the Solaris 8 release, you can verify this by using the `psrinfo` command:

```
# psrinfo -v
Status of processor 0 as of: 07/12/99 09:41:47
  Processor has been on-line since 07/08/99 13:51:11.
  The sparcv9 processor operates at 333 MHz,
        and has a sparcv9 floating point processor.
```

If the processor type is `sparcv9`, the platform is capable of running the 64-bit Solaris operating environment. This test does not work on previous versions of the `psrinfo` command, where all platforms report the less precise `sparc` as the processor type.

▼ How to Determine Whether a System Has 64-bit Solaris Capabilities Enabled

You can use the `isainfo` command to determine whether a system has 64-bit capabilities enabled, which means the system is booted with the 64–bit kernel.

Examples—Determining Whether a System Has 64–bit Solaris Capabilities Enabled

An UltraSPARC system running a 32-bit kernel looks like this:

```
$ isainfo -v
32-bit sparc applications
```

The output means this system is capable of supporting only 32-bit applications.

An UltraSPARC system running a 64-bit kernel looks like this:

```
 $ isainfo -v
64-bit sparcv9 applications
32-bit sparc applications
```

This output means this system is capable of supporting both 32-bit and 64-bit applications.

Use the `isainfo -b` command to display the number of bits supported by native applications on the running system.

The output from a SPARC, IA, or UltraSPARC system running the 32–bit Solaris operating environment looks like this:

```
$ isainfo -b
32
```

The output from a 64–bit UltraSPARC system running the 64–bit Solaris operating environment looks like:

```
$ isainfo -b
64
```

The command returns `64` only. Even though a 64–bit UltraSPARC system is capable of running both types of applications, 64–bit applications are the best kind of applications to run on a 64–bit system.

The `uname -p` output remains `sparc` or `i386` to ensure that existing 32-bit applications continue to run without interruption.

▼ How to Display System and Software Release Information

To display specific system and software release information, use the `showrev` command.

```
$ showrev [-a]
```

–a — Displays all system release information available.

Example—Displaying System and Software Release Information

The following example shows `showrev` command output.

```
$ showrev -a
Hostname: starbug
Hostid: nnnnnnnn
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: solar.com
Kernel version: SunOS 5.8 s28_26 February 2000

OpenWindows version:
OpenWindows Version 3.6.2  9 August 1999

No patches are installed
$
```

▼ How to Display General System Information (uname)

To display system information, use the uname command.

```
$ uname[-a]
```

–a	Displays the operating system name as well as the system node name, operating system release, operating system version, hardware name, and processor type.

Example—Displaying General System Information

The following example shows uname command output.

```
$ uname
SunOS
$ uname -a
SunOS starbug 5.8 Generic sun4u sparc SUNW,Ultra-5_10
$
```

▼ How to Display a System's Host ID Number

To display the host identification number in hexadecimal format, use the hostid command.

```
$ hostid
```

Example—Displaying a System's Host ID Number

The following example shows sample output from the `hostid` command.

```
$ hostid
80a5d34c
```

▼ How to Display a System's Installed Memory

To display the amount of memory installed on your system, use the `prtconf` command.

```
$ prtconf [| grep Memory]
```

`grep Memory`	Focuses output from this command to display memory information only.

Example—Displaying a System's Installed Memory

The following example shows sample output from the `prtconf` command.

```
# prtconf | grep Memory
Memory size: 128 Megabytes
```

▼ How to Display the Date and Time

To display the current date and time according to your system clock, use the `date` command.

```
$ date
```

Example—Displaying the Date and Time

The following example shows sample output from the `date` command.

```
$ date
Thu Sep 16 14:06:44 MDT 1999
$
```

Using Commands to Change System Information

The table below shows man page references and descriptions for some commands that enable you to change general system information.

TABLE 27–2 Commands for Changing System Information

Command	Enables You to Change a System's ...
`rdate`(1M)	Date and time to match those of another system
`date`(1)	Date and time to match your specifications

Using these commands, you can set a system's date and time to synchronize with the date and time of another system, such as a server. Or you can change a system's date and time by specifying new information.

The message of the day (MOTD) facility, located in `/etc/motd`, enables you to send announcements or inquiries to all users of a system when they log in. Use this facility sparingly, and edit this file regularly to remove obsolete messages.

By editing the `/etc/system` file, you can:

- Change the number of processes per user
- Increase the number of lock requests
- Increase shared memory segments

Using Network Time Protocol (NTP) in Your Network

The Network Time Protocol (NTP) public domain software from the University of Delaware is included in the Solaris software starting with the Solaris 2.6 release.

NTP enables you to manage precise time and network clock synchronization in a network environment. The `xntpd` daemon sets and maintains the system time-of-day. The `xntpd` daemon is a complete implementation of the version 3 standard, as defined by RFC 1305.

The `xntpd` daemon reads the `/etc/inet/ntp.conf` file at system startup. See `xntpd`(1M) for information about configuration options.

Keep the following in mind when using NTP in your network:

- The `xntpd` daemon takes up minimal system resources.
- An NTP client synchronizes automatically with an NTP server when it boots, and if it gets out of sync, it will resync again when it sees a time server.

▼ How to Set Up an NTP Server

1. **Become superuser.**
2. **Change to the `/etc/inet` directory.**
3. **Copy the `ntp.server` file to the `ntp.conf` file.**

```
# cp ntp.server ntp.conf
```

4. **Change to the `/etc/init.d` directory.**
5. **Start the `xntpd` daemon.**

```
# ./xntpd start
```

▼ How to Set Up an NTP Client

1. **Become superuser.**
2. **Change to the `/etc/inet` directory.**
3. **Copy the `ntp.client` file to the `ntp.conf` file.**

```
# cp ntp.client ntp.conf
```

4. **Change to the `/etc/init.d` directory.**
5. **Start the `xntpd` daemon.**

```
# ./xntpd start
```

▼ How to Synchronize Date and Time From Another System

1. **Become superuser.**

2. **To reset the date and time to synchronize with another system, use the `rdate` command.**

```
# rdate another-system
```

another-system	Name of another system.

3. **Verify that you have reset your system's date correctly by checking your system's date and time using the `date` command.**

   The output should show a date and time that matches that of the other system.

Example—Synchronizing Date and Time From Another System

The following example shows how to use `rdate` to synchronize the date and time of one system with another. In this example, the system `earth`, running several hours behind, is reset to match the date and time of the server `starbug`.

```
earth# date
Thu Sep 16 11:08:27 MDT 1999
earth# rdate starbug
Thu Sep 16 14:06:37 1999
earth# date
Thu Sep 16 14:06:40 MDT 1999
```

▼ How to Set a System's Date and Time Manually

1. **Become superuser.**

2. **Enter the new date and time.**

```
# date mmddHHMM[[cc]yy]
```

mm	Month, using two digits.
dd	Day of the month, using two digits.
HH	Hour, using two digits and a 24-hour clock.
MM	Minutes, using two digits.
cc	Century, using two digits.
yy	Year, using two digits.

3. **Verify that you have reset your system's date correctly by checking your system's date and time using the** `date` **command with no options.**

   The output should show a date and time that matches that of the other system.

Example—Setting a System's Date and Time Manually

The following example shows how to use `date` to manually set a system's date and time.

```
# date
Thu Sep 16 14:00:00 MDT 1999
# date 0916141099
Thu Sep 16 14:10:00 MDT 1999
```

▼ How to Set Up a Message of the Day

1. **Become superuser.**

2. **Edit the** `/etc/motd` **file and add a message of your choice.**

   Edit the text to include the message that will be displayed during the user login process, including spaces, Tabs, and Returns.

3. **Verify the changes by displaying the contents of the** `/etc/motd`**.**

```
$ cat /etc/motd
Welcome to the UNIX Universe. Have a nice day.
```

Example—Setting Up a Message of the Day

The default message of the day, provided when you install Solaris software, contains SunOS version information:

```
$ cat /etc/motd
Sun Microsystems Inc.   SunOS 5.8       Generic  February 2000
```

The following example shows an edited `/etc/motd` file that provides information about system availability to each user who logs in.

```
$ cat /etc/motd
The system will be down from 7:00 a.m to 2:00 p.m.on
Saturday, July 10, for upgrades and maintenance.
Do not try to access the system during those hours.
Thank you...
```

▼ How to Set the Number of Processes per User

1. **Become superuser.**

2. **Edit the `/etc/system` file and add the following line.**

```
set maxuprc=value
```

value	Number of processes a user can run at once.

3. **Verify the `maxuprc` value change.**

```
# grep maxuprc /etc/system
set maxuprc=100
```

4. Reboot the system.

Example—Setting the Number of Processes per User

The following example shows the line you would add to the `/etc/system` file to allow users to run 100 processes each.

```
set maxuprc=100
```

▼ How to Increase Shared Memory Segments

1. Become superuser.

2. Edit the `/etc/system` file and add the following variables to increase shared memory segments.

```
set shmsys:shminfo_shmmax=value
set shmsys:shminfo_shmmin=value
set shmsys:shminfo_shmmni=value
set shmsys:shminfo_shmseg=value
set semsys:seminfo_semmap=value
set semsys:seminfo_semmni=value
set semsys:seminfo_semmns=value
set semsys:seminfo_semmsl=value
set semsys:seminfo_semmnu=value
set semsys:seminfo_semume=value
```

`shmsys:shminfo_shmmax`	Maximum shared memory segment size
`shmsys:shminfo_shmmin`	Minimum shared memory segment size
`shmsys:shminfo_shmmni`	Number of shared memory identifiers
`shmsys:shminfo_shmseg`	Number of segments, per process
`semsys:seminfo_semmap`	Number of entries in the semaphore map
`semsys:seminfo_semmni`	Number of semaphore identifiers

`semsys:seminfo_semmns`	Number of semaphores in the system
`semsys:seminfo_semmsl`	Maximum number of semaphores, per ID
`semsys:seminfo_semmnu`	Number of processes using the undo facility
`semsys:seminfo_semume`	Maximum number of undo structures per process

3. Verify the shared memory value changes.

```
# grep shmsys /etc/system
```

4. Reboot the system.

```
# init 6
```

Example—Increasing Shared Memory Segments

The following shared memory values accommodate a system with a large amount of memory (for example, 128 MBytes) that is running a large database application.

```
set shmsys:shminfo_shmmax=268435456
set shmsys:shminfo_shmmin=200
set shmsys:shminfo_shmmni=200
set shmsys:shminfo_shmseg=200
set semsys:seminfo_semmap=250
set semsys:seminfo_semmni=500
set semsys:seminfo_semmns=500
set semsys:seminfo_semmsl=500
set semsys:seminfo_semmnu=500
set semsys:seminfo_semume=100
```

CHAPTER 28

Managing Disk Use (Tasks)

This chapter describes how to optimize disk space by locating unused files and large directories. This is a list of the step-by-step instructions in this chapter.

- "How to Display Information About Blocks, Files, and Disk Space" on page 467
- "How to Display the Size of Files" on page 470
- "How to Find Large Files" on page 471
- "How to Find Files That Exceed a Given Size Limit" on page 472
- "How to Display the Size of Directories, Subdirectories, and Files" on page 473
- "How to Display the User Allocation of Local UFS File Systems" on page 474
- "How to List the Newest Files" on page 476
- "How to Find and Remove Old or Inactive Files " on page 476
- "How to Clear Out Temporary Directories" on page 478
- "How to Find and Delete `core` Files" on page 478
- "How to Delete Crash Dump Files" on page 479

Displaying Blocks and Files Used

Use the `df` command and its options to report the number of free disk blocks and files. For more information, see `df`(1M).

▼ How to Display Information About Blocks, Files, and Disk Space

Display information about how disk space is used by using the `df` command.

```
$ df [directory] [-F fstype] [-g] [-k] [-t]
```

`df`	With no options, lists all mounted file systems and their device names, the number of total 512-byte blocks used, and the number of files.
directory	Directory whose file system you want to check. The device name, blocks used, and number of files are displayed.
`–F` *fstype*	Displays a list of unmounted file systems, their device names, the number of 512-byte blocks used, and the number of files on file systems of type *fstype*.
`–g`	Displays the `statvfs` structure for all mounted file systems.
`–k`	Displays a list of file systems, kilobytes used, free kilobytes, percent capacity used, and mount points.
`–t`	Displays total blocks as well as blocks used for all mounted file systems.

Examples—Displaying Information About Blocks, Files, and Disk Space

In the following example, all the file systems listed are locally mounted except for `/usr/dist`, which is mounted remotely from the system `venus`.

```
$ df
/                (/dev/dsk/c0t0d0s0 ):  287530 blocks    92028 files
/usr             (/dev/dsk/c0t0d0s6 ): 1020214 blocks   268550 files
/proc            (/proc             ):       0 blocks      878 files
/dev/fd          (fd                ):       0 blocks        0 files
/etc/mnttab      (mnttab            ):       0 blocks        0 files
/var/run         (swap              ):  396016 blocks     9375 files
/tmp             (swap              ):  396016 blocks     9375 files
/opt             (/dev/dsk/c0t0d0s5 ):  381552 blocks    96649 files
/export/home     (/dev/dsk/c0t0d0s7 ):  434364 blocks   108220 files
/usr/dist        (venus:/usr/dist   ):14750510 blocks  2130134 files
```

In the following example, the file system, total Kbytes, used Kbytes, available Kbytes, percent of capacity used, and mount point are displayed.

```
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0     192807   49042  124485    29%    /
/dev/dsk/c0t0d0s6    1190551  680444  450580    61%    /usr
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
mnttab                     0       0       0     0%    /etc/mnttab
```

```
swap                  198056       0  198056     0%    /var/run
swap                  198064       8  198056     1%    /tmp
/dev/dsk/c0t0d0s5     192807    2031  171496     2%    /opt
/dev/dsk/c0t0d0s7     217191       9  195463     1%    /export/home
venus:/usr/dist     20612581 13237326 6963005   66%    /usr/dist
```

The following example shows information about the same system as the previous example, but only UFS file system information is displayed.

```
$ df -F ufs
/                  (/dev/dsk/c0t0d0s0 ):  287530 blocks    92028 files
/usr               (/dev/dsk/c0t0d0s6 ): 1020214 blocks   268550 files
/opt               (/dev/dsk/c0t0d0s5 ):  381552 blocks    96649 files
/export/home       (/dev/dsk/c0t0d0s7 ):  434364 blocks   108220 files
```

Note - Although `/proc` and `/tmp` are local file systems, they are not UFS file systems (`/proc` is a PROCFS file system, `/var/run` and `/tmp` are TMPFS file systems, and `/etc/mnttab` is a MNTFS file system).

The following example shows a list of all mounted file systems, device names, total 512-byte blocks used, and number of files. The second line of each two-line entry displays the total number of blocks and files allocated for the file system.

```
$ df -t
/                  (/dev/dsk/c0t0d0s0 ):   287530 blocks    92028 files
                                  total:   385614 blocks    96832 files
/usr               (/dev/dsk/c0t0d0s6 ):  1020214 blocks   268550 files
                                  total:  2381102 blocks   300288 files
/proc              (/proc             ):        0 blocks      879 files
                                  total:        0 blocks      924 files
/dev/fd            (fd                ):        0 blocks        0 files
                                  total:        0 blocks       72 files
/etc/mnttab        (mnttab            ):        0 blocks        0 files
                                  total:        0 blocks        1 files
/var/run           (swap              ):   396112 blocks     9375 files
                                  total:   396112 blocks     9395 files
/tmp               (swap              ):   396112 blocks     9375 files
                                  total:   396128 blocks     9395 files
/opt               (/dev/dsk/c0t0d0s5 ):   381552 blocks    96649 files
                                  total:   385614 blocks    96832 files
/export/home       (/dev/dsk/c0t0d0s7 ):   434364 blocks   108220 files
                                  total:   434382 blocks   108224 files
/usr/dist          (venus:/usr/dist   ): 14750510 blocks  2130134 files
                                  total: 41225162 blocks  2482176 files
```

Checking the Size of Files

You can check the size of files and sort them by using the `ls` command. You can find files that exceed a size limit by using the `find` command. For more information, see `ls`(1) and `find`(1).

▼ How to Display the Size of Files

1. **Change the directory to where the files you want to check are located.**

2. **Display the size of the files.**

```
$ ls [-l] [-s]
```

`–l`	Displays a list of files and directories in long format, showing the sizes in bytes.
`–s`	Displays a list of the files and directories, showing the sizes in blocks.

Examples—Displaying the Size of Files

The following example shows that `lastlog` and `messages` are larger than the other files in the `/var/adm` directory.

```
$ cd /var/adm
$ ls -l
total 144
drwxrwxr-x   5 adm      adm          512 Sep  1 14:11 acct/
-rw-------   1 uucp     bin            0 Sep  1 14:08 aculog
-r--r--r--   1 root     root      350700 Sep  3 10:37 lastlog
drwxr-xr-x   2 adm      adm          512 Sep  1 14:08 log/
-rw-r--r--   1 root     root       14619 Sep  2 16:11 messages
-rw-r--r--   1 adm      adm         8200 Sep  3 14:35 pacct
-rw-r--r--   1 adm      adm          920 Sep  3 10:47 pacct1
drwxr-xr-x   2 adm      adm          512 Sep  1 14:08 passwd/
drwxrwxr-x   2 adm      sys          512 Sep  1 14:11 sa/
drwxr-xr-x   2 root     sys          512 Sep  1 14:36 sm.bin/
-rw-rw-rw-   1 root     bin            0 Sep  1 14:08 spellhist
-rw-------   1 root     root         420 Sep  3 14:17 sulog
-rw-r--r--   1 root     bin         4092 Sep  3 10:37 utmpx
-rw-r--r--   1 root     root         122 Sep  1 15:39 vold.log
```

(continued)

(Continuation)

```
-rw-r--r--   1 adm      adm        11904 Sep  3 10:47 wtmpx
```

The following example shows that `lpsched.1` uses two blocks.

```
$ cd /var/lp/logs
$ ls -s
total 2            0 lpsched       2 lpsched.1
```

▼ How to Find Large Files

1. **Change directory to the location you want to search.**
2. **Display the size of files in blocks from largest to smallest.**

```
$ ls -s | sort -nr | more
```

`sort -nr`	Sorts the list of files by block size from smallest to largest.

Example—Finding Large Files

In the following example, `lastlog` and `messages` are the largest files in the `/var/adm` directory.

```
$ cd /var/adm
$ ls -s | sort -nr | more
  48 lastlog
  30 messages
  24 wtmpx
  18 pacct
   8 utmpx
   2 vold.log
   2 sulog
   2 sm.bin/
   2 sa/
   2 passwd/
   2 pacct1
   2 log/
   2 acct/
```

(continued)

```
   0 spellhist
   0 aculog
total 144
```

▼ How to Find Files That Exceed a Given Size Limit

To locate and display the names of files that exceed a specified size, use the `find` command.

```
$ find directory -size +nnn
```

directory	Directory you want to search.
–size +*nnn*	Is a number of 512-byte blocks. Files that exceed the size indicated are listed.

Example—Finding Files That Exceed a Given Size Limit

The following example shows how to find files with more than 400 blocks in the current working directory.

```
$ find . -size +400 -print
./Howto/howto.doc
./Howto/howto.doc.backup
./Howto/howtotest.doc
./Routine/routineBackupconcepts.doc
./Routine/routineIntro.doc
./Routine/routineTroublefsck.doc
./.record
./Mail/pagination
./Config/configPrintadmin.doc
./Config/configPrintsetup.doc
./Config/configMailappx.doc
./Config/configMailconcepts.doc
./snapshot.rs
```

Checking the Size of Directories

You can display the size of directories by using the du command and its options. Additionally, you can find the amount of disk space taken up by user accounts on local UFS file systems by using the quot command. For more information about these commands, see du(1M)and quot(1M).

▼ How to Display the Size of Directories, Subdirectories, and Files

Display the size of one or more directories, subdirectories, and files by using the du command. Sizes are displayed in 512-byte blocks.

```
$ du [-as] [directory ...]
```

du	Displays the size of each directory you specify, including each subdirectory beneath it.
–a	Displays the size of each file and subdirectory, and the total number of blocks contained in the specified directory.
–s	Displays only the total number of blocks contained in the specified directory.
directory ...	Specifies one or more directories you want to check.

Examples—Displaying the Size of Directories, Subdirectories, and Files

The following example displays the total sizes of two directories and all the subdirectories they contain.

```
$ du -s /var/adm /var/spool/lp
130     /var/adm
40      /var/spool/lp
```

The following example displays the sizes of two directories, all of the subdirectories and files they contain, and the total number of blocks contained in each directory.

```
$ du /var/adm /var/spool/lp
2       /var/adm/log
2       /var/adm/passwd
2       /var/adm/acct/fiscal
2       /var/adm/acct/nite
2       /var/adm/acct/sum
8       /var/adm/acct
2       /var/adm/sa
2       /var/adm/sm.bin
130     /var/adm
4       /var/spool/lp/admins
2       /var/spool/lp/fifos/private
2       /var/spool/lp/fifos/public
6       /var/spool/lp/fifos
2       /var/spool/lp/requests/starbug
4       /var/spool/lp/requests
2       /var/spool/lp/system
2       /var/spool/lp/tmp/starbug
2       /var/spool/lp/tmp/.net/tmp/starbug
4       /var/spool/lp/tmp/.net/tmp
2       /var/spool/lp/tmp/.net/requests/starbug
4       /var/spool/lp/tmp/.net/requests
10      /var/spool/lp/tmp/.net
14      /var/spool/lp/tmp
40      /var/spool/lp
```

▼ How to Display the User Allocation of Local UFS File Systems

1. **Become superuser.**

2. **Display users, directories, or file systems, and the number of 1024-byte blocks used.**

```
# quot [-a] [filesystem]
```

–a	Lists all users of each mounted UFS file system and the number of 1024-byte blocks used.
filesystem	Is a UFS file system. Users and the number of blocks used are displayed.

Note - The `quot` command works only on local UFS file systems.

Example—Displaying the User Allocation of Local UFS File Systems

In the following example, users of the root (/) file system are displayed, then users of all mounted UFS file systems are displayed.

```
# quot /
/dev/rdsk/c0t0d0s0:
43340   root
 3142   rimmer
   47   uucp
   35   lp
   30   adm
    4   bin
    4   daemon
# quot -a
/dev/rdsk/c0t0d0s0 (/):
43340   root
 3150   rimmer
   47   uucp
   35   lp
   30   adm
    4   bin
    4   daemon
/dev/rdsk/c0t0d0s6 (/usr):
460651  root
206632  bin
  791   uucp
   46   lp
    4   daemon
    1   adm
/dev/rdsk/c0t0d0s7 (/export/home):
    9   root
```

Finding and Removing Old and Inactive Files

Part of the job of cleaning up heavily loaded file systems involves locating and removing files that have not been used recently. You can locate unused files using the `ls` or `find` commands. For more information, see `ls`(1) and `find`(1).

Other ways to conserve disk space include emptying temporary directories such as the ones located in `/var/tmp` or `/var/spool`, and deleting `core` and crash dump files. For more information about these files, refer to Chapter 39.

▼ How to List the Newest Files

List files, displaying the most recently created or changed files first, by using the `ls -t` command.

```
$ ls -t [directory]
```

`–t`	Sorts listings by latest time stamp first.
directory	Directory you want to search.

Example—Listing the Newest Files

The following example shows how to use `ls -tl` to locate the most recent files within the `/var/adm` directory. The `sulog` file was created or edited most recently.

```
$ ls -tl /var/adm
total 134
-rw-------   1 root     root         315 Sep 24 14:00 sulog
-r--r--r--   1 root     other     350700 Sep 22 11:04 lastlog
-rw-r--r--   1 root     bin         4464 Sep 22 11:04 utmpx
-rw-r--r--   1 adm      adm        20088 Sep 22 11:04 wtmpx
-rw-r--r--   1 root     other          0 Sep 19 03:10 messages
-rw-r--r--   1 root     other          0 Sep 12 03:10 messages.0
-rw-r--r--   1 root     root       11510 Sep 10 16:13 messages.1
-rw-r--r--   1 root     root           0 Sep 10 16:12 vold.log
drwxr-xr-x   2 root     sys          512 Sep 10 15:33 sm.bin
drwxrwxr-x   5 adm      adm          512 Sep 10 15:19 acct
drwxrwxr-x   2 adm      sys          512 Sep 10 15:19 sa
-rw-------   1 uucp     bin            0 Sep 10 15:17 aculog
-rw-rw-rw-   1 root     bin            0 Sep 10 15:17 spellhist
drwxr-xr-x   2 adm      adm          512 Sep 10 15:17 log
drwxr-xr-x   2 adm      adm          512 Sep 10 15:17 passwd
```

▼ How to Find and Remove Old or Inactive Files

1. **Become superuser.**

2. **Find files that have not been accessed for a specified number of days and list them in a file.**

```
# find directory -type f[-atime + nnn] [-mtime + nnn] -print > filename
```

directory	Directory you want to check. Directories below this also will be checked.
–atime +*nnn*	Finds files that have not been accessed within the number of days you specify.
–mtime +*nnn*	Finds files that have not been modified within the number of days you specify.
filename	File containing the list of inactive files.

3. Remove the inactive files that you listed in the previous step.

```
# rm ‘cat filename‘
```

filename	File created in previous step which contains the list of inactive files.

Example—Finding and Removing Old or Inactive Files

The following example locates regular files in `/var/adm` and its directories that have not been accessed in the last 60 days and saves the list of inactive files in `/var/tmp/deadfiles`. These files are then removed with the `rm` command.

```
# find /var/adm -type f -atime +60 -print > /var/tmp/deadfiles &
# more /var/tmp/deadfiles
/var/adm/log/asppp.log
/var/adm/aculog
/var/adm/spellhist
/var/adm/wtmpx
/var/adm/sa/sa13
/var/adm/sa/sa27
/var/adm/sa/sa11
/var/adm/sa/sa23
/var/adm/sulog
/var/adm/vold.log
/var/adm/messages.1
/var/adm/messages.2
/var/adm/messages.3
# rm ‘cat /var/tmp/deadfiles‘
#
```

▼ How to Clear Out Temporary Directories

1. **Become superuser.**

2. **Change to the** `/var/tmp` **directory.**

```
# cd /var/tmp
```

Caution - Be sure you are in the right directory before completing the following step. The next step deletes all files in the current directory.

3. **Delete the files and subdirectories in the current directory.**

```
# rm -r *
```

4. **Change to other directories containing unnecessary temporary or obsolete subdirectories and files, and delete them by repeating Step 3 above.**

Example—Clearing Out Temporary Directories

The following example shows how to clear out the `/var/tmp` directory, and verifies that all files and subdirectories were removed.

```
# cd /var/tmp
# ls
deadfiles           wxconAAAa0003r:0.0  wxconAAAa000NA:0.0
test_dir            wxconAAAa0003u:0.0  wxconAAAa000cc:0.0
wxconAAAa000zs:0.0
# rm -r *
# ls
#
```

▼ How to Find and Delete `core` Files

1. **Become superuser.**

2. **Change the directory to where you want to start the search.**

3. **Find and remove any** `core` **files in this directory and its subdirectories.**

```
# find . -name core -exec rm {} \;
```

Example—Finding and Deleting `core` Files

The following example shows how to find and remove `core` files from the user account belonging to `jones` using the `find` command.

```
# cd /home/jones
# find . -name core -exec rm {} \;
```

▼ How to Delete Crash Dump Files

Crash dump files can be very large, so if you have enabled your system to store these files, do not retain them for longer than necessary.

1. **Become superuser.**

2. **Change to the directory where crash dump files are stored.**

```
# cd /var/crash/system
```

system — System that created the crash dump files.

Caution - Be sure you are in the right directory before completing the following step. The next step deletes all files in the current directory.

3. **Remove the crash dump files.**

```
# rm *
```

4. **Verify the crash dump files are removed.**

```
# ls
```

Example—Deleting Crash Dump Files

The following example shows how to remove crash dump files from the system venus, and how to verify that the crash dump files were removed.

```
# cd /var/crash/venus
# rm *
# ls
```

CHAPTER 29

Managing Quotas (Tasks)

This chapter describes how to set up and administer quotas for disk space and inodes. This is a list of the step-by-step instructions in this chapter.

- "How to Configure File Systems for Quotas" on page 485
- "How to Set Up Quotas for a User" on page 486
- "How to Set Up Quotas for Multiple Users" on page 487
- "How to Check Quota Consistency" on page 487
- "How to Turn Quotas On" on page 488
- "How to Check for Exceeded Quotas" on page 489
- "How to Check Quotas on a File System" on page 490
- "How to Change the Soft Time Limit Default" on page 492
- "How to Change Quotas for a User" on page 493
- "How to Disable Quotas for a User" on page 494
- "How to Turn Quotas Off" on page 495

Using Quotas

Using quotas enable system administrators to control the size of UFS file systems by limiting the amount of disk space and the number of inodes (which roughly corresponds to the number of files) that individual users can acquire. For this reason, quotas are especially useful on the file systems where user home directories reside.

Once they are in place, quotas can be changed to adjust the amount of disk space or number of inodes that users can consume. Additionally, quotas can be added or

removed as system needs change. See “Changing and Removing Quotas” on page 492 for instructions on changing quotas or the amount of time that quotas can be exceeded, disabling individual quotas, or removing quotas from file systems.

In addition, quota status can be monitored. Quota commands enable administrators to display information about quotas on a file system, or search for users who have exceeded their quotas. For procedures that describe how to use these commands, see “Checking Quotas” on page 489.

Soft Limits and Hard Limits

You can set both soft and hard limits. The system will not allow a user to exceed his or her hard limit. However, a system administrator may set a soft limit (sometimes referred to as a quota) which can be temporarily exceeded by the user. The soft limit must be less than the hard limit.

Once the user exceeds the soft limit, a timer begins. While the timer is ticking, the user is allowed to operate above the soft limit but cannot exceed the hard limit. Once the user goes below the soft limit, the timer gets reset. However, if the user’s usage remains above the soft limit when the timer expires, the soft limit is enforced as a hard limit. By default, the soft limit timer is seven days.

The value of the timer is shown by the `timeleft` field in the `repquota` and `quota` commands.

For example, let’s say a user has a soft limit of 10,000 blocks and a hard limit of 12,000 blocks. If the user’s block usage exceeds 10,000 blocks and the timer is also exceeded (more than seven days), the user will not be able to allocate more disk blocks on that file system until his or her usage drops below the soft limit.

Difference Between Disk Block and File Limits

There are two resources that a file system provides to the user: blocks (for data) and inodes (for files). Each file consumes one inode. File data is stored in data blocks (usually made of up 1 kilobyte blocks.)

Assuming there are no directories, it is possible for a user to exceed his or her inode quota without using any blocks by creating all empty files. It is also possible for a user to use only one inode yet exceed his or her block quota by simply creating one file large enough to consume all the data blocks in the user’s quota.

Setting Up Quotas

You can set up quotas to limit the amount of disk space and number of inodes (roughly equivalent to the number of files) available to users. These quotas are activated automatically each time a file system is mounted. This section describes how to configure file systems for quotas, and how to set up and activate quotas.

Setting up quotas involves these general steps:

1. A series of commands prepares a file system to accept quotas, ensuring that quotas will be enforced each time the system is rebooted and the file system is mounted. Entries must be added to the `/etc/vfstab` file, and a `quotas` file must be created in the top-level directory of the file system.
2. After a quota is created for one user, it can be copied as a prototype to set up other user quotas.
3. Before quotas are actually turned on, another command checks for consistency by comparing the proposed quotas with the current disk usage to make sure that there are no conflicts.
4. Finally, a command turns the quotas on for one or more entire file systems.

These steps ensure that quotas are automatically activated on a file system each time it is mounted. For specific information about these procedures, see “Setting Up Quotas Task Map” on page 484.

The following table describes the commands you use to set up disk quotas.

TABLE 29–1 Commands for Setting Up Quotas

Command	Enables You To ...
`edquota`(1M)	Set the hard and soft limits on the number of inodes and disk space for each user
`quotacheck`(1M)	Examine each mounted UFS file system, comparing against information stored in the file system’s disk quota file, and resolve inconsistencies
`quotaon`(1M)	Activate the quotas for the specified file systems
`quota`(1M)	Display user’s quotas on mounted file systems to verify that quotas have been correctly set up

Guidelines for Setting Up Quotas

Before you set up quotas, you need to determine how much space and how many inodes to allocate to each user. If you want to be sure the total file system space is never exceeded, you can divide the total size of the file system between the number of users. For example, if three users share a 100-Mbyte slice and have equal disk space needs, you could allocate 33 Mbytes to each. In environments where not all users are likely to push their limits, you may want to set individual quotas so that they add up to more than the total size of the file system. For example, if three users share a 100-Mbyte slice, you could allocate 40 Mbytes to each.

When you have established a quota for one user by using the `edquota` command, you can use this quota as a prototype to set the same quota for other users on the same file system.

After you have configured UFS file systems for quotas and established quotas for each user, run the `quotacheck` command to check consistency between current disk usage and quota files before you actually turn quotas on. Also, if systems are rebooted infrequently, it is a good idea to periodically run `quotacheck`.

The quotas you set up with `edquota` are not enforced until you turn them on by using the `quotaon` command. If you have properly configured the quota files, quotas will be turned on automatically each time a system is rebooted and the file system is mounted.

Setting Up Quotas Task Map

TABLE 29–2 Setting Up Quotas Task Map

Task	Description	For Instructions, Go To ...
1. Configure a File System for Quotas	Edit `/etc/vfstab` so that quotas are activated each time the file system is mounted, and create a `quotas` file.	"How to Configure File Systems for Quotas" on page 485
2. Set Up Quotas for a User	Use the `edquota` command to create disk and inode quotas for a single user account.	"How to Set Up Quotas for a User" on page 486
3. Set Up Quotas for Multiple Users	*Optional.* Use `edquota` to apply prototype quotas to other user accounts.	"How to Set Up Quotas for Multiple Users" on page 487

TABLE 29–2 Setting Up Quotas Task Map *(continued)*

Task	Description	For Instructions, Go To ...
4. Check for Consistency	Use the `quotacheck` command to compare quotas to current disk usage for consistency on one or more file systems.	"How to Check Quota Consistency" on page 487
5. Turn Quotas On	Use the `quotaon` command to initiate quotas on one or more file systems.	"How to Turn Quotas On" on page 488

▼ How to Configure File Systems for Quotas

1. **Become superuser.**
2. **Edit the `/etc/vfstab` file and add `rq` to the `mount options` field for each UFS file system that will have quotas.**
3. **Change directory to the top of the file system that will have quotas.**
4. **Create a file named `quotas`.**

```
# touch quotas
```

5. **Change permissions to read/write for root only.**

```
# chmod 600 quotas
```

Examples—Configuring File Systems for Quotas

The following example from `/etc/vfstab` shows that the `/export/home` directory from the system `pluto` is mounted as an NFS file system on the local system with quotas enabled, signified by the `rq` entry under the `mount options` column.

```
#device           device   mount       FS    fsck   mount   mount
#to mount         to fsck  point       type  pass   at boot options
#
pluto:/export/home -       /export/home nfs    -     yes    rq
```

The following example line from `/etc/vfstab` shows that the local `/work` directory is mounted with quotas enabled, signified by the `rq` entry under the `mount options` column.

```
#device           device            mount  FS   fsck mount   mount
#to mount         to fsck           point  type pass at boot options
#
/dev/dsk/c0t4d0s0 /dev/rdsk/c0t4d0s0 /work ufs  3    yes     rq
```

▼ How to Set Up Quotas for a User

1. **Become superuser.**

2. **Use the quota editor to create a temporary file containing one line of quota information for each mounted UFS file system that has a `quotas` file in its top-level directory.**

```
# edquota username
```

username	User for whom you want to set up quotas.

3. **Change the number of 1-Kbyte disk blocks, both soft and hard, and the number of inodes, both soft and hard, from 0 (the default) to the quotas you specify for each file system.**

4. **Verify the user's quota by using the `quota` command.**

```
# quota -v username
```

–v	Display's user's quota information on all mounted file systems where quotas exist.
username	Specifies user name to view quota limits.

Examples—Setting Up Quotas for a User

The following example shows the contents of the temporary file opened by `edquota` on a system where `/files` is the only mounted file system containing a `quotas` file in its top-level directory.

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

The following example shows the same line in the temporary file after quotas have been set up.

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

▼ How to Set Up Quotas for Multiple Users

1. **Become superuser.**

2. **Use the quota editor to apply the quotas you already established for a prototype user to the additional users you specify.**

```
# edquota -p prototype-user username ...
```

prototype-user	User name of the account for which you have set up quotas.
username ...	Specifies one or more user names of additional accounts.

Example—Setting Up Prototype Quotas for Multiple Users

The following example applies the quotas established for user `bob` to users `mary` and `john`.

```
# edquota -p bob mary john
```

▼ How to Check Quota Consistency

Note - To ensure accurate disk data, the file systems being checked should be quiescent when you run the `quotacheck` command manually. The `quotacheck` command is run automatically when a system is rebooted.

---

1. **Become superuser.**

2. **Run a consistency check on UFS file systems.**

```
# quotacheck [-v] filesystem
```

–v	(Optional) Identifies the disk quotas for each user on a particular file system.
–a	Checks all file systems with an `rq` entry in the `/etc/vfstab` file.
filesystem	Specifies a file system to check.

See `quotacheck`(1M) for more information.

Example—Checking Quota Consistency

The following example checks quotas for the `/export/home` file system on the `/dev/rdsk/c0t0d0s7` slice. The `/export/home` file system is the only file system with an `rq` entry in the `/etc/vfstab` file.

```
# quotacheck -va
*** Checking quotas for /dev/rdsk/c0t0d0s7 (/export/home)
```

▼ How to Turn Quotas On

1. **Become superuser.**

2. **Turn file system quotas on by using the `quotaon` command.**

```
# quotaon [-v] -a filesystem ...
```

–v	Verbose option.
–a	Turns quotas on for all file systems with an `rq` entry in the `/etc/vfstab` file.
filesystem ...	Turns quotas on for one or more file systems that you specify.

Example—Turning Quotas On

The following example turns quotas on for the file systems on the `/dev/dsk/c0t4d0s7` and `/dev/dsk/c0t3d0s7` slices.

```
# quotaon -v /dev/dsk/c0t4d0s7 /dev/dsk/c0t3d0s7
/dev/dsk/c0t4d0s7: quotas turned on
/dev/dsk/c0t3d0s7: quotas turned on
```

Checking Quotas

After you have set up and turned on disk and inode quotas, you can check for users who exceed their quotas. In addition, you can check quota information for entire file systems.

The table below describes the commands you use to check quotas.

TABLE 29–3 Commands for Checking Quotas

Command	Task
`quota`(1M)	Display user quotas and current disk use, and information about users who are exceeding their quotas
`repquota`(1M)	Display quotas, files, and amount of space owned for specified file systems

▼ How to Check for Exceeded Quotas

You can display the quotas and disk use for individual users on file systems on which quotas have been activated by using the `quota` command.

1. **Become superuser.**

2. **Display user quotas for mounted file systems where quotas are enabled.**

```
# quota [-v] username
```

`–v`	Displays users' quotas on all mounted file systems that have quotas.
username	Is the login name or UID of a user's account.

Example—Checking for Exceeded Quotas

The following example shows that the user account identified by UID 301 has a quota of one Kbyte but has not used any disk space.

```
# quota -v 301
Disk quotas for bob (uid 301):
Filesystem  usage  quota limit timeleft files quota  limit timeleft
/export/home   0      1     2             0      2      3
```

`Filesystem`	Is the mount point for the file system.
`usage`	Is the current block usage.
`quota`	Is the soft block limit.
`limit`	Is the hard block limit.
`timeleft`	Is the amount of time (in days) left on the quota timer.
`files`	Is the current inode usage.
`quota`	Is the soft inode limit.
`limit`	Is the hard inode limit.
`timeleft`	Is the amount of time (in days) left on the quota timer.

▼ How to Check Quotas on a File System

Display the quotas and disk use for all users on one or more file systems by using the `repquota` command.

1. Become superuser.

2. Display all quotas for one or all file systems, even if there is no usage.

```
# repquota [-v]  -a filesystem
```

–v	Reports on quotas for all users-even those who do not consume resources.
–a	Reports on all file systems.
filesystem	Reports on the specified file system.

Example—Checking Quotas on a File System

The following example shows output from the `repquota` command on a system that has quotas enabled on only one file system (`/export/home`).

```
# repquota -va
/dev/dsk/c0t3d0s7 (/export/home):
                  Block limits              File limits
User      used   soft   hard  timeleft  used  soft   hard  timeleft
#301  --            0      1   2.0 days         0      2      3
#341  --    57     50     60   7.0 days  2      90    100
```

`Block Limits`	
`used`	Is the current block usage.
`soft`	Is the soft block limit.
`hard`	Is the hard block limit.
`timeleft`	Is the amount of time (in days) left on the quota timer.
`File Limits`	
`used`	Is the current inode usage.
`soft`	Is the soft inode limit.

`hard`	Is the hard inode limit.
`timeleft`	Is the amount of time (in days) left on the quota timer.

Changing and Removing Quotas

You can change quotas to adjust the amount of disk space or number of inodes users can consume. You can also remove quotas for individual users or from entire file systems as needed.

The following table describes the commands you use to change or remove quotas.

TABLE 29–4 Commands for Changing and Removing Quotas

Command	Task
`edquota`(1M)	Change the hard and soft limits on the number of inodes or disk space for each user. Also, change the soft quota time limit for each file system with a quota.
`quotaoff`(1M)	Turn off quotas for specified file systems.

▼ How to Change the Soft Time Limit Default

Users can exceed the soft time limits for their quotas for one week, by default. This means that after a week of repeated violations of the soft time limits of either disk space or inode quotas, the system prevents users from using any more inodes or disk blocks.

You can change the length of time that users may exceed their disk space or inode quotas by using the `edquota` command.

1. **Become superuser.**

2. **Use the quota editor to create a temporary file containing soft time limits.**

```
# edquota -t
```

3. **Change the time limits from 0 (the default) to the time limits you specify by numbers and the keywords** `month`**,** `week`**,** `day`**,** `hour`**,** `min`**, or** `sec`**.**

Note - This procedure doesn't affect current quota violators.

Examples—Changing the Soft Time Limit Default

The following example shows the contents of the temporary file opened by `edquota` on a system where `/export/home` is the only mounted file system with quotas. The `0` (default) value means that the default time limit of one week is used.

```
fs /export/home blocks time limit = 0 (default), files time limit = 0 (default)
```

The following example shows the same temporary file after the time limit for exceeding the blocks quota has been changed to two weeks, and the time limit for exceeding the number of files has been changed to 16 days.

```
fs /export/home blocks time limit = 2 weeks, files time limit = 16 days
```

▼ How to Change Quotas for a User

1. **Become superuser.**

2. **Use the quota editor to open a temporary file containing one line for each mounted file system that has a** `quotas` **file in its top-level directory.**

```
# edquota username
```

username — User name whose quota you want to change.

Caution - Although you can specify multiple users as arguments to the `edquota` command, the information displayed does not show which user it belongs to, which could create some confusion.

---

3. **Enter the number of 1-Kbyte disk blocks, both soft and hard, and the number of inodes, both soft and hard.**

4. **Verify that a user's quota has been correctly changed by using the** `quota` **command.**

```
# quota -v username
```

–v	Displays user quota information on all mounted file systems with quotas enabled.
username	User name whose quota you want to check.

Examples—Changing Quotas for a User

The following example shows the contents of the temporary file opened by `edquota` on a system where `/files` is the only mounted file system containing a `quotas` file in its top-level directory.

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

The following example shows the same temporary file after quotas have been changed.

```
fs /files blocks (soft = 0, hard = 500) inodes (soft = 0, hard = 100)
```

The following example shows how to verify that the hard quotas for user `smith` have been changed to 500 1-Kbyte blocks, and 100 inodes.

```
# quota -v smith
Disk quotas for smith (uid 12):
Filesystem  usage  quota  limit  timeleft  files  quota  limit  timeleft

  /files     1       0     500              1       0     100
```

▼ How to Disable Quotas for a User

1. **Become superuser.**

2. **Use the quota editor to create a temporary file containing one line for each mounted file system that has a `quotas` file in its top-level directory.**

```
# edquota username
```

username	User name whose quota you want to disabled.

Caution - Although you can specify multiple users as arguments to the `edquota` command, the information displayed does not show which user it belongs with, which could create some confusion.

3. **Change the number of 1-Kbyte disk blocks, both soft and hard, and the number of inodes, both soft and hard, to 0.**

   Note - Be sure you change the values to zero. Do *not* delete the line from the text file.

4. **Verify that you have disabled a user's quota by using the `quota` command.**

```
# quota -v username
```

–v	Displays user quota information on all mounted file systems with quotas enabled.
username	User name (UID) whose quota you want to check.

Examples—Disabling Quotas for a User

The following example shows the contents of the temporary file opened by `edquota` on a system where `/files` is the only mounted file system containing a quotas file in its top-level directory.

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

The following example shows the same temporary file after quotas have been disabled.

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

▼ How to Turn Quotas Off

1. **Become superuser.**
2. **Turn file system quotas off.**

```
# quotaoff [-v] -a filesystem ...
```

–v	Displays a message from each file system when quotas are turned off.
–a	Turns quotas off for all file systems.
filesystem	Turns quotas off for one or more file systems you specify.

Example—Turning Quotas Off

The following example turns the quotas off for the `/export/home` file system.

```
# quotaoff -v /export/home
/export/home: quotas turned off
```

CHAPTER 30

Scheduling System Events (Tasks)

This chapter describes how to schedule routine or one-time system events by using the `crontab` and `at` commands. It also explains how to control access to these commands by using `cron.deny`, `cron.allow`, and `at.deny` files.

This is a list of the step-by-step instructions in this chapter.

- "How to Create or Edit a `crontab` File" on page 501
- "How to Display a `crontab` File" on page 503
- "How to Remove a `crontab` File" on page 504
- "How to Deny `crontab` Access" on page 506
- "How to Limit `crontab` Access to Specified Users" on page 507
- "How to Create an `at` Job" on page 509
- "How to Display the `at` Queue" on page 511
- "How to Display `at` Jobs" on page 511
- "How to Remove `at` Jobs" on page 512
- "How to Deny `at` Access" on page 513

Commands for Scheduling System Events

You can schedule system events to execute repetitively, at regular intervals, by using the `crontab` command.You can schedule a single system event for execution at a specified time by using the `at` command. The following table summarizes `crontab` and `at`, as well as the files that enable you to control access to these commands.

TABLE 30–1 Command Summary: Scheduling System Events

Command	What It Schedules	Location of Files	Files That Control Access
`crontab`	Multiple system events at regular intervals	`/var/spool/cron/crontabs`	`/etc/cron.d/cron.allow` and `/etc/cron.d/cron.deny`
`at`	A single system event	`/var/spool/cron/atjobs`	`/etc/cron.d/at.deny`

Scheduling a Repetitive System Event (`cron`)

The following sections describe how to create, edit, display, and remove `crontab` files, as well as how to control access to them.

Inside a `crontab` File

The `cron` daemon schedules system events according to commands found within each `crontab` file. A `crontab` file consists of commands, one per line, that will be executed at regular intervals. The beginning of each line contains date and time information that tells the `cron` daemon when to execute the command.

For example, a `crontab` file named `root` is supplied during SunOS software installation. Its contents include these command lines:

```
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
```

The first command line instructs the system to run `logchecker` at 3:10 on Sundays and Thursdays nights. The second command line schedules the system to run `newsyslog` at 3:10 every Sunday morning. The third command line orders the

system to execute `nfsfind` Sundays at 3:15 in the morning. The fourth command line instructs the system to check daily for daylight savings time and make corrections if necessary. If there is no RTC time zone nor an `/etc/rtc_config` file, this entry will do nothing. The fifth command line instructs the system to check for and remove duplicate entries in the Generic Security Service table, `/etc/gss/gsscred_db`.

For more information about the syntax of lines within a `crontab` file, see “Syntax of `crontab` File Entries” on page 500.

The `crontab` files are stored in `/var/spool/cron/crontabs`. Several `crontab` files besides `root` are provided during SunOS software installation (see the following table).

TABLE 30–2 Default `crontab` Files

`crontab` **File**	**Function**
`adm`	Accounting
`lp`	Printing
`root`	General system functions and file system cleanup
`sys`	Performance collection
`uucp`	General `uucp` cleanup

Besides the default `crontab` file, users can create `crontab` files to schedule their own system events.

Other `crontab` files are named after the user accounts in which they are created, such as `bob`, `mary`, `smith`, or `jones`.

To access `crontab` files belonging to root or other users, superuser privileges are required.

Procedures explaining how to create, edit, display, and remove `crontab` files are described in “Commands for Scheduling System Events” on page 497.

How the `cron` Daemon Handles Scheduling

The `cron` daemon handles the automatic scheduling of `crontab` commands. Its function is to check the `/var/spool/cron/crontab` directory for the presence of `crontab` files, normally every 15 minutes. It checks for new `crontab` files or

changes to existing ones, reads the execution times listed within the files, and submits the commands for execution at the proper times.

In much the same way, the `cron` daemon controls the scheduling of `at` files, which are stored in the `/var/spool/cron/atjobs` directory.

Syntax of `crontab` File Entries

A `crontab` file consists of commands, one per line, that execute automatically at the time specified by the first five fields at the beginning of each command line. These first five fields, described in the following table, are separated by spaces. They indicate when the command will be executed.

TABLE 30–3 Values for `crontab` Time Fields

Time Field	Values
Minute	0-59
Hour	0-23
Day of month	1-31
Month	1-12
Day of week	0-6 (0 = Sunday)

Follow these guidelines to use special characters in `crontab` time fields:

- Use a space to separate each field.
- Use a comma to separate multiple values.
- Use a hyphen to designate a range of values.
- Use an asterisk as a wildcard to include all possible values.
- Use a comment mark (#) at the beginning of a line to indicate a comment or a blank line.

For example, the following sample `crontab` command entry displays a reminder in the user's console window at 4 p.m. on the first and fifteenth of every month.

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

Each command within a `crontab` file must consist of one line, even if it is very long, because `crontab` does not recognize extra carriage returns. For more detailed information about `crontab` entries and command options, refer to `crontab`(1).

Creating and Editing `crontab` Files

The simplest way to create a `crontab` file is to use the `crontab -e` command to invoke the text editor set up for your system environment, defined by the `EDITOR` environment variable. If this variable has not been set, `crontab` uses the default editor `ed`. Define your `EDITOR` environment to be an editor you are familiar with. The following example shows how to check to see whether an editor has been defined, and how to set up `vi` as the default.

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

When you create a `crontab` file, it is automatically placed in the `/var/spool/cron/crontabs` directory and is given your user name. You can create or edit a `crontab` file for another user, or root, if you have superuser privileges.

Enter `crontab` command entries as described in "Syntax of `crontab` File Entries" on page 500.

▼ How to Create or Edit a `crontab` File

1. **(Optional) Become superuser to create or edit a `crontab` file belonging to root or another user.**

2. **Create a new `crontab` file, or edit an existing one.**

```
$ crontab -e [username]
```

username	Name of another user's account, requires root privileges to create or edit.

Caution - If you accidentally enter the `crontab` command with no option, press the interrupt character for your editor. This allows you to quit without saving changes. Exiting the file and saving changes at this point would overwrite an existing `crontab` file with an empty file.

3. **Add command lines to the file, following the syntax described in "Syntax of** `crontab` **File Entries" on page 500.**

   The `crontab` file will be placed in `/var/spool/cron/crontabs`.

4. **Verify the** `crontab` **file by using the** `crontab -l` **command.**

```
# crontab -l [username]
```

Example—Creating or Editing a `crontab` File

The following example shows how to create a `crontab` file for another user.

```
# crontab -e jones
```

The following command entry added to a new `crontab` file will automatically remove any log files from the user's home directory at 1:00 am every Sunday morning. Because the command entry does not redirect output, redirect characters are added to the command line after `*.log` to make sure that the command executes properly.

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/jones/*.log > /dev/null 2>&1
```

▼ How to Verify a `crontab` File

To verify that a `crontab` file exists for a user, use the `ls -l` command in the `/var/spool/cron/crontabs` directory. For example, the following display shows that `crontab` files exist for users `smith` and `jones`.

```
$ ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Feb 26 16:23 adm
-rw-------  1 root     staff        225 Mar  1  9:19 jones
-rw-r--r--  1 root     root        1063 Feb 26 16:23 lp
-rw-r--r--  1 root     sys          441 Feb 26 16:25 root
-rw-------  1 root     staff         60 Mar  1  9:15 smith
-rw-r--r--  1 root     sys          308 Feb 26 16:23 sys
```

Verify the contents of user's `crontab` file by using `crontab -l` as described in "How to Display a `crontab` File" on page 503.

Displaying crontab Files

The crontab -l command displays the contents of your crontab file much the way the cat command displays the contents of other types of files. You do not have to change directories to /var/spool/cron/crontabs (where crontab files are located) to use this command.

By default, the crontab -l command displays your own crontab file. To display crontab files belonging to other users, you must be superuser.

▼ How to Display a crontab File

1. **(Optional) Become superuser to display a** crontab **file belonging to root or another user.**

2. **Display the** crontab **file.**

```
$ crontab -l [username]
```

username	Name of another user's account, and requires superuser privileges to create or edit.

Caution - If you accidentally enter the crontab command with no option, press the interrupt character for your editor. This allows you to quit without saving changes. Exiting the file and saving changes at this point would overwrite an existing crontab file with an empty file.

Example—Displaying a crontab File

The following example shows how to use crontab -l to display the contents of the default user's crontab file, the default root crontab file, and the crontab file belonging to another user.

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
$ su
Password:
# crontab -l
#ident  "@(#)root       1.19    98/07/06 SMI"   /* SVr4.0 1.1.3.1       */
```

(continued)

```
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null
2>&1
```

Removing `crontab` Files

By default, `crontab` file protections are set up so that you cannot inadvertently delete a `crontab` file by using the `rm` command. Instead, use the `crontab -r` command to remove `crontab` files.

By default, `crontab -r` removes your own `crontab` file. You must be superuser to remove `crontab` files belonging to superuser or other users.

You do not have to change directories to `/var/spool/cron/crontabs` (where `crontab` files are located) to use this command.

▼ How to Remove a `crontab` File

1. **(Optional) Become superuser to remove a `crontab` file belonging to root or another user.**

2. **Remove the `crontab` file.**

```
$ crontab -r [username]
```

username	Name of another user's account, and requires superuser privileges to create or edit.

Caution - If you accidentally enter the `crontab` command with no option, press the interrupt character for your editor. This allows you to quit without saving changes. Exiting the file and saving changes at this point would overwrite an existing `crontab` file with an empty file.

3. Verify the crontab file is removed.

```
# ls /var/spool/cron/crontabs
```

Example—Removing a `crontab` File

The following example shows how user `smith` uses the `crontab -r` command to remove his `crontab` file.

```
$ ls /var/spool/cron/crontabs
adm     jones     lp     root    smith    sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm     jones     lp     root    sys    uucp
```

Controlling Access to `crontab`

You can control access to `crontab` by using two files in the `/etc/cron.d` directory: `cron.deny` and `cron.allow`. These files permit only specified users to perform `crontab` tasks such as creating, editing, displaying, or removing their own `crontab` files.

The `cron.deny` and `cron.allow` files consist of a list of user names, one per line. These access control files work together like this:

- If `cron.allow` exists, only the users listed in this file can create, edit, display, or remove `crontab` files.
- If `cron.allow` doesn't exist, all users may submit `crontab` files, except for users listed in `cron.deny`.
- If neither `cron.allow` nor `cron.deny` exists, superuser privileges are required to run `crontab`.

Superuser privileges are required to edit or create the `cron.deny` and `cron.allow` files.

During SunOS software installation, a default `cron.deny` file is provided:

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

None of these user names can access `crontab` commands. You can edit this file to add other user names who will be denied access to the `crontab` command.

No default `cron.allow` file is supplied. This means that, after Solaris software installation, all users (except the ones listed in the default `cron.deny` file) can access `crontab`. If you create a `cron.allow` file, only these users can access `crontab` commands.

▼ How to Deny `crontab` Access

1. **Become superuser.**

2. **Edit the `/etc/cron.d/cron.deny` file and add user names, one per line, who will be prevented from using `crontab` commands.**

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

3. **Verify the `/etc/cron.d/cron.deny` file.**

```
# cat /etc/cron.d/cron.deny
```

▼ How to Limit `crontab` Access to Specified Users

1. **Become superuser.**

2. **Create the `/etc/cron.d/cron.allow` file.**

3. **Enter the user names, one per line, who will be allowed to use `crontab` commands.**

```
root
username1
username2
username3
.
.
.
```

Be sure to add `root` to this list. If you do not, superuser access to `crontab` commands will be denied.

Examples—Limiting `crontab` Access to Specified Users

The following example shows a `cron.deny` file that prevents user names `visitor`, `jones`, and `temp` from accessing `crontab`.

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

The following example shows a `cron.allow` file. The users `smith`, `jones`, `lp`, and `root` are the only ones who may access `crontab`.

```
$ cat /etc/cron.d/cron.allow
root
jones
lp
smith
```

▼ How to Verify Limited `crontab` Accesss

To verify whether or not a specific user can access `crontab`, use the `crontab -l` command while logged into the user account.

```
$ crontab -l
```

If the user can access `crontab`, and already has created a `crontab` file, it will be displayed. Otherwise, if the user can access `crontab` but no `crontab` file exists, a message like the following will be displayed:

```
crontab: can't open your crontab file
```

This user either is listed in `cron.allow` (if it exists), or is not listed in `cron.deny`.

If the user cannot access `crontab`, the following message is displayed whether or not a previous `crontab` file exists:

```
crontab: you are not authorized to use cron. Sorry.
```

This means either that the user is not listed in `cron.allow` (if it exists), or the user is listed in `cron.deny`.

Scheduling a Single System Event (`at`)

The following sections describe how to use `at`(1) to schedule jobs (commands and scripts) for execution at a later time, how to display and remove these jobs, and how to control access to the `at` command.

By default, users can create, display, and remove their own `at` job files. To access `at` files belonging to root or other users, you must have superuser privileges.

When you submit an `at` job, it is assigned a job identification number along with the `.a` extension that becomes its file name.

at Command Description

Submitting an at job file includes:

1. Invoking the at utility, specifying a command execution time.
2. Entering a command or script to execute later.

Note - If output from this command or script is important, be sure to direct it to a file for later examination.

For example, the following at job removes core files from the user account smith near midnight on the last day of July.

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Sat Jul 31 23:45:00 1999
```

at Command Security

You can set up a file to control access to the at command, permitting only specified users to create, remove, or display queue information about their at jobs. The file that controls access to at, /etc/cron.d/at.deny, consists of a list of user names, one per line. The users listed in this file cannot access at commands.

The at.deny file, created during SunOS software installation, contains the following user names:

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

With superuser privileges, you can edit this file to add other user names whose at access you want to restrict.

▼ How to Create an at Job

1. **Start the** at **utility, specifying the time you want your job executed, and press Return.**

```
$ at [-m] time [date]
```

`-m`	Sends you mail after the job is completed.
time	Hour that you want to schedule the job. Add `am` or `pm` if you do not specify the hours according to a 24-hour clock. `midnight`, `noon`, and `now` are acceptable keywords. Minutes are optional.
date	First three or more letters of a month, a day of the week, or the keywords `today` or `tomorrow`.

2. **At the `at` prompt, enter the commands or scripts you want to execute, one per line. You may enter more than one command by pressing Return at the end of each line.**

3. **Exit the `at` utility and save the `at` job by pressing Control-d.**

   Your `at` job is assigned a queue number, which is also its file name. This number is displayed when you exit the `at` utility.

Examples—Creating an `at` Job

The following example shows the `at` job that user `jones` created to remove her backup files at 7:30 at night. She used the –`m` option so that she would receive a mail message after her job completed.

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-d
job 897355800.a at Mon Jul  12 19:30:00 1999
```

She received a mail message which confirmed the execution of her `at` job.

```
Your ‘‘at’’ job ‘‘rm /home/jones/*.backup’’
completed.
```

The following example shows how `jones` scheduled a large `at` job for 4:00 Saturday morning. The output of which was directed to `big.file`.

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

▼ How to Display the at Queue

To check your jobs that are waiting in the at queue, use the atq command. This command displays status information about the at jobs that you created.

```
$ atq
```

▼ How to Verify an at Job

To verify that you have created an at job, use the atq command. The atq command confirms that at jobs belonging to jones have been submitted to the queue.

```
$ atq
Rank   Execution Date     Owner     Job          Queue   Job Name
  1st   Jul 12, 1999 19:30   jones  897355800.a     a     stdin
  2nd   Jul 14, 1999 23:45   jones  897543900.a     a     stdin
  3rd   Jul 17, 1999 04:00   jones  897732000.a     a     stdin
```

▼ How to Display at Jobs

To display information about the execution times of your at jobs, use the at -l command.

```
$ at -l [job-id]
```

–l *job-id*	Identification number of the job whose status you want to examine.

Example—Displaying at Jobs

The following example shows output from the at -l command, used to get status information on all jobs submitted by a user.

```
$ at -l
897543900.a Wed Jul 14 23:45:00 1999
897355800.a Mon Jul 12 19:30:00 1999
897732000.a Sat Jul 17 04:00:00 1999
```

The following example shows output displayed when a single job is specified with the `at -l` command.

```
$ at -l 897732000.a
897732000.a  Sat Jul 17 04:00:00 1999
```

▼ How to Remove `at` Jobs

1. **(Optional) Become superuser to remove an `at` job belonging to root or another user.**

2. **Remove the `at` job from the queue before it is executed.**

```
$ at -r [job-id]
```

–r *job-id*	Identification number of the job you want to remove.

3. **Verify the `at` job is removed by using the `at -l` (or the `atq`) command to display the jobs remaining in the `at` queue. The job whose identification number you specified should not appear.**

```
$ at -l [job-id]
```

Example—Removing `at` Jobs

In the following example, a user wants to remove an `at` job that was scheduled to execute at 4 am on July 17th. First, the user displays the `at` queue to locate the job identification number. Next, the user removes this job from the `at` queue. Finally, the user verifies that this job has been removed from the queue.

```
$ at -l
897543900.a Wed Jul 14 23:45:00 1999
897355800.a Mon Jul 12 19:30:00 1999
897732000.a Sat Jul 17 04:00:00 1999
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

Controlling Access to at

Users listed in the at.deny file cannot use at to schedule jobs or to check the at queue status.

The at.deny file is placed in the /etc/cron.d directory during Solaris software installation. At that time, the same users are listed in both this file and the default cron.deny file.

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

Root permissions are required to edit this file.

▼ How to Deny at Access

1. **Become superuser.**

2. **Edit the** /etc/cron.d/at.deny **file and add the names of users, one per line, who will be prevented from using** at **commands.**

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

Example—Denying `at` Access

The following example shows an `at.deny` file that has been edited so that the users `smith` and `jones` may not access the `at` command.

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

▼ How to Verify `at` Access Is Denied

To verify whether or not a user's name was added correctly to `/etc/cron.d/at.deny`, use the `at -l` command while logged in as the user. If the user cannot access `at` commands, the following message is displayed.

```
# su smith
Password:
$ at -l
at: you are not authorized to use at.  Sorry.
```

Likewise, if the user tries to submit an `at` job, the following message is displayed:

```
$ at 2:30pm
at: you are not authorized to use at.  Sorry.
```

This confirms that the user is listed in the `at.deny` file.

If `at` access is allowed, the `at -l` command returns nothing.

CHAPTER 31

Managing System Accounting (Tasks)

This section contains some simple procedures for setting up and maintaining system accounting.

This is a list of the step-by-step instructions in this chapter.

- "How to Set Up System Accounting" on page 516
- "How to Bill Users" on page 519
- "How to Fix a `wtmpx` File" on page 520
- "How to Fix `tacct` Errors" on page 520
- "How to Restart `runacct`" on page 521
- "How to Set Up System Accounting" on page 516
- "How to Permanently Disable System Accounting" on page 523

Setting Up System Accounting

You can set up system accounting to run while the system is in multiuser mode (system state 2). Generally, this involves:

1. Creating `/etc/rc0.d/K22acct` and `/etc/rc2.d/S22acct`
2. Modifying `/var/spool/cron/crontabs/adm` and `/var/spool/cron/crontabs/root`

Most of the accounting scripts are added to the `/var/spool/cron/crontabs/adm` database file. The following table describes the default accounting scripts.

TABLE 31–1 Default Accounting Scripts

Accounting Script ...	Is Used To ...	And Runs ...
`ckpacct`(1M)	Check the size of the `/usr/adm/pacct` log file	Periodically
`runacct`(1M)	Process connect, disk, and fee accounting information	Daily
`monacct`(1M)	Generate fiscal reports and is run once per period	On a fiscal basis

You can change these defaults. After these entries have been added to the database and the accounting programs have been installed, accounting should run automatically.

▼ How to Set Up System Accounting

1. **Become superuser.**

2. **If necessary, install the `SUNWaccr` and `SUNWaccu` packages on your system by using the `pkgadd` command.**

3. **Install `/etc/init.d/acct` as the startup script for Run Level 2.**

```
# ln /etc/init.d/acct /etc/rc2.d/S22acct
```

4. **Install `/etc/init.d/acct` as the stop script for Run Level 0.**

```
# ln /etc/init.d/acct /etc/rc0.d/K22acct
```

5. **Add the following lines to the `adm` `crontab` file to start the `ckpacct`, `runacct`, and `monacct` programs automatically.**

```
# EDITOR=vi; export EDITOR
# crontab -e adm
0 * * * * /usr/lib/acct/ckpacct
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
```

(continued)

```
30 7 1 * * /usr/lib/acct/monacct
```

6. **Add the following line to the** `root` `crontab` **file to start the** `dodisk` **program automatically.**

```
# crontab -e
30 22 * * 4 /usr/lib/acct/dodisk
```

7. **Edit** `/etc/acct/holidays` **to include national and local holidays.**

8. **Reboot the system, or type**:

```
# /etc/init.d/acct start
```

Examples—Setting Up Accounting

The following example shows how the `crontab` entries that run `/usr/lib/acct/ckpacct`, `/usr/lib/acct/runacct`, and `/usr/lib/acct/monacct` have been added to `/var/spool/cron/crontabs/adm`.

```
#ident  "@(#)adm        1.5     92/07/14 SMI"    /* SVr4.0 1.2   */
#
# The adm crontab file should contain startup of performance
# collection if the profiling and performance feature has been
# installed.
0 * * * * /usr/lib/acct/ckpacct
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
30 7 1 * * /usr/lib/acct/monacct
```

The following example shows how the `crontab` entry that runs `/usr/lib/acct/dodisk` has been added to `/var/spool/cron/crontabs/root`.

```
#ident  "@(#)root       1.16    98/04/28 SMI"   /* SVr4.0 1.1.3.1       */
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 22 * * 4 /usr/lib/acct/dodisk
```

The following example shows a sample `/etc/acct/holidays` file.

```
* @(#)holidays January 1, 1999
*
* Prime/Nonprime Table for UNIX Accounting System
*
* Curr Prime Non-Prime
* Year Start Start
*
  1999 0800 1800
*
* only the first column (month/day) is significant.
*
* month/day Company
*   Holiday
*
1/1  New Years Day
7/4  Indep. Day
12/25  Christmas
```

Billing Users

If you provide special user services on a request basis, such as restoring files or remote printing, you may want to bill users by running a utility called `chargefee`(1M). `chargefee` records charges in the file `/var/adm/fee`. Each time the `runacct` utility is executed, new entries are merged into the total accounting records.

▼ How to Bill Users

1. **Become superuser.**

2. **Charge a user for special services.**

```
# chargefee username amount
```

username	User account you want to bill.
amount	Number of units to bill the user.

Example—Billing Users

The following example charges the user `print_customer` 10 units.

```
# chargefee print_customer 10
```

Maintaining Accounting Information

This section describes how to maintain accounting information.

Fixing Corrupted Files and `wtmpx` Errors

Unfortunately, the UNIX accounting system is not foolproof. Occasionally, a file will become corrupted or lost. Some of the files can simply be ignored or restored from backup. However, certain files must be fixed to maintain the integrity of the accounting system.

The `wtmpx`(4) files seem to cause the most problems in the day-to-day operation of the accounting system. When the date is changed and the system is in multiuser mode, a set of date change records is written into `/var/adm/wtmpx`. The `wtmpfix`(1M) utility is designed to adjust the time stamps in the `wtmp` records when a date change is encountered. However, some combinations of date changes and reboots will slip through `wtmpfix` and cause `acctcon` to fail. For instructions on correcting `wtmpx` problems, see the following procedure..

▼ How to Fix a `wtmpx` File

1. **Become superuser.**
2. **Change to the** `/var/adm/acct/nite` **directory.**
3. **Convert the binary file** `wtmp.`***MMDD*** **into the ASCII file** `xwtmp`**.**

```
# fwtmp wtmp.MMDD xwtmp
```

MMDD	Pair of two-digit numbers representing the month and day.

4. **Edit** `xwtmp`**. Delete the corrupted files, or delete all records from the beginning up to the date change.**
5. **Convert the ASCII file** `xwtmp` **to a binary file, overwriting the corrupted file.**

```
# fwtmp -ic xwtmp wtmp.MMDD
```

Fixing `tacct` Errors

The integrity of `/var/adm/acct/sum/tacct` is important if you are charging users for system resources. Occasionally, mysterious `tacct` records appear with negative numbers, duplicate user IDs, or a user ID of 65535. First, check `/var/adm/acct/sum/tacctprev`, using `prtacct` to print it. If the contents look all right, patch the latest `/var/adm/acct/sum/tacct.`*MMDD* file, then recreate the `/var/adm/acct/sum/tacct` file. The following steps outline a simple patch procedure.

▼ How to Fix `tacct` Errors

1. **Become superuser.**
2. **Change to the** `/var/adm/acct/sum` **directory.**
3. **Convert the contents of** `tacct.`***MMDD*** **from binary to ASCII format.**

```
# acctmerg -v tacct.MMDD xtacct
```

MMDD Month and day specified by two-digit numbers.

4. **Edit the** `xtacct` **file, removing bad records and writing duplicate records to another file.**

5. **Convert the** `xtacct` **file from ASCII format to binary.**

```
# acctmerg -i xtacct tacct.MMDD
```

MMDD Month and day specified by two-digit numbers.

6. **Merge the files** `tacct.prev` **and** `tacct.`***MMDD*** **into the file** `tacct`**.**

```
# acctmerg tacctprev tacct.MMDD tacct
```

Restarting `runacct`

The `runacct` program can fail for a variety of reasons, the most common being a system crash, `/var` running out of space, or a corrupted `wtmpx` file. If the `active`*MMDD* file exists, check it first for error messages. If the `active` and lock files exist, check `fd2log` for any mysterious messages.

Called without arguments, `runacct` assumes that this is the first invocation of the day. The argument *MMDD* is necessary if `runacct` is being restarted and specifies the month and day for which `runacct` will rerun the accounting. The entry point for processing is based on the contents of `statefile`. To override `statefile`, include the desired state on the command line.

Caution - When running the `runacct` program manually, be sure to run it as user `adm`.

▼ How to Restart `runacct`

1. **Remove the** `lastdate` **file and any** `lock`*** files, if any.**

```
$ cd /var/adm/acct/nite
$ rm lastdate lock*
```

2. **Restart the** `runacct` **program.**

```
$ runacct MMDD [state] 2> /var/adm/acct/nite/fd2log &
```

MMDD	Month and day specified by two-digit numbers.
state	Specifies a state, or starting point, where `runacct` processing should begin.

Stopping and Disabling System Accounting

You can temporarily stop system accounting or disable it permanently.

▼ How to Temporarily Stop System Accounting

1. **Become superuser.**

2. **Edit the** `adm` `crontab` **file to stop the** `ckpacct`**,** `runacct`**, and** `monacct` **programs from running by commenting out the appropriate lines.**

```
# EDITOR=vi; export EDITOR
# crontab -e adm
#0 * * * * /usr/lib/acct/ckpacct
#30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
#30 7 1 * * /usr/lib/acct/monacct
```

3. **Edit the** `crontab` **file for user** `root` **to stop the** `dodisk` **program from running by commenting out the appropriate line.**

```
# crontab -e
#30 22 * * 4 /usr/lib/acct/dodisk
```

4. **Stop the accounting program.**

```
# /etc/init.d/acct stop
```

To re-enable system accounting, remove the newly added comment symbols from the `crontab` files and restart the accounting program.

```
# /etc/init.d/acct start
```

▼ How to Permanently Disable System Accounting

1. **Become superuser.**

2. **Edit the** `adm crontab` **file and delete the entries for the** `ckpacct`**,** `runacct`**, and** `monacct` **programs.**

```
# EDITOR=vi; export EDITOR
# crontab -e adm
```

3. **Edit the** `root crontab` **file and delete the entries for the** `dodisk` **program.**

```
# crontab -e
```

4. **Remove the startup script for Run Level 2.**

```
# unlink /etc/rc2.d/S22acct
```

5. **Remove the stop script for Run Level 0.**

```
# unlink /etc/rc0.d/K22acct
```

6. **Stop the accounting program.**

```
# /etc/init.d/acct stop
```

CHAPTER 32

System Accounting (Reference)

This is a list of reference information in this chapter.

- "Daily Accounting" on page 525
- "Connect Accounting " on page 525
- "Process Accounting " on page 526
- "Disk Accounting " on page 526
- "Calculating User Fees" on page 527
- "How Daily Accounting Works" on page 527
- "Daily Accounting Reports " on page 529
- "The `runacct` Program" on page 538
- "Accounting Files" on page 540

Daily Accounting

Daily accounting can help you track four types of accounting: *connect accounting*, *process accounting*, *disk accounting*, and *fee calculations*.

Connect Accounting

Connect accounting enables you to determine the following:

- The length of time a user was logged in
- How the `tty` lines are being used

- The number of reboots on your system
- The frequency with which the accounting software was turned off and on

To provide this information, the system stores records of time adjustments, boot times, times the accounting software was turned off and on, changes in run levels, the creation of user processes (`login` processes and `init` processes), and the deaths of processes. These records (produced from the output of system programs such as `date`, `init`, `login`, `ttymon`, and `acctwtmp`) are stored in the `/var/adm/wtmpx` file. Entries in the `wtmpx` file may contain the following information: a user's login name, a device name, a process ID, the type of entry, and a time stamp denoting when the entry was made.

Process Accounting

Process accounting enables you to keep track of the following data about each process run on your system:

- User and group IDs of those using the process
- Beginning and elapsed times of the process
- CPU time for the process (user time and system time)
- Amount of memory used
- Commands run
- The `tty` controlling the process

Every time a process dies, the `exit` program collects this data and writes it to the `/var/adm/pacct` file.

Disk Accounting

Disk accounting enables you to gather and format the following data about the files each user has on disks:

- Name and ID of the user
- Number of blocks used by the user's files

This data is collected by the shell script `/usr/lib/acct/dodisk` at intervals determined by the entry you add to the `/var/spool/cron/crontabs/root` file. In turn, `dodisk` invokes the commands `acctdusg` and `diskusg`, which gather disk usage by login.

See "How to Set Up System Accounting" on page 516 for more information about setting up `dodisk`.

The `acctdusg`(1M) command gathers all the disk accounting information. Each time it is invoked, this command can process a maximum of 3000 users.

Caution - Information gathered by running `dodisk`(1M) is stored in the `/var/adm/acct/nite/disktacct` file. This information is overwritten the next time `dodisk` is run. Therefore, avoid running `dodisk` twice in the same day.

The `diskusg` command may overcharge for files that are written in random access fashion, which may create holes in the files. This is because `diskusg` does not read the indirect blocks of a file when determining its size. Rather, `diskusg` determines the size of a file by looking at the *di_size* value of the inode.

Calculating User Fees

The `chargefee` utility stores charges for special services provided to a user, such as file restoration, in the file `/var/adm/fee`. Each entry in the file consists of a user's login name, user ID, and the fee. This file is checked by the `runacct` program every day and new entries are merged into the total accounting records. For instructions on running `chargefee` to bill users, see "How to Bill Users" on page 519.

How Daily Accounting Works

Here is a step-by-step summary of how daily accounting works:

1. When the system is switched into multiuser mode, the `/usr/lib/acct/startup` program is executed. The `startup` program executes several other programs that invoke accounting.
2. The `acctwtmp` program adds a "boot" record to `/var/adm/wtmpx`. In this record, the system name is shown as the login name in the `wtmpx` record. The following table summarizes how the raw accounting data is gathered and where it is stored.

TABLE 32–1 Raw Accounting Data

File in `/var/adm`	**Information**	**Written By**	**Format**
`wtmpx`	Connect sessions	`login`, `init`	`utmpx.h`
	Changes	`date`	
	Reboots	`acctwtmp`	
	Shutdowns	`shutacct` shell	

TABLE 32–1 Raw Accounting Data *(continued)*

File in /var/adm	Information	Written By	Format
pacctn	Processes	Kernel (when the process ends)	acct.h
		turnacct switch (creates a new file when the old one reaches 500 blocks)	
fee	Special charges	chargefee	acct.h
acct/nite/ disktacct	Disk space used	dodisk	tacct.h

3. The turnacct program, invoked with the –on option, begins process accounting. Specifically, turnacct executes the accton program with the /var/adm/pacct argument.
4. The remove shell script "cleans up" the saved pacct and wtmpx files left in the sum directory by runacct.
5. The login and init programs record connect sessions by writing records into /var/adm/wtmpx. Any date changes (using date with an argument) are also written to /var/adm/wtmpx. Reboots and shutdowns using acctwtmp are also recorded in /var/adm/wtmpx.
6. When a process ends, the kernel writes one record per process, using acct.h format, in the /var/adm/pacct file.

   Every hour, cron executes the ckpacct program to check the size of /var/adm/pacct. If the file grows past 500 blocks (default), the turnacct switch is executed. (The program moves the pacct file and creates a new one.) The advantage of having several smaller pacct files becomes apparent when trying to restart runacct if a failure occurs when processing these records.
7. runacct is executed by cron each night. runacct processes the accounting files: /var/adm/pacct*n*, /var/adm/wtmpx, /var/adm/fee, and /var/adm/acct/nite/disktacct, to produce command summaries and usage summaries by login.
8. The /usr/lib/acct/prdaily program is executed on a daily basis by runacct to write the daily accounting information collected by runacct (in ASCII format) in /var/adm/acct/sum/rprt.*MMDD*.
9. The monacct program should be executed on a monthly basis (or at intervals determined by you, such as the end of every fiscal period). The monacct program creates a report based on data stored in the sum directory that has been

updated daily by `runacct`. After creating the report, `monacct` “cleans up” the `sum` directory to prepare the directory’s files for the new `runacct` data.

What Happens if the System Shuts Down

If the system is shut down using `shutdown`, the `shutacct` program is executed automatically. The `shutacct` program writes a reason record into `/var/adm/wtmpx` and turns off process accounting.

Accounting Reports

This section describes the various reports generated by the accounting software.

Daily Accounting Reports

The `runacct`(1M) shell script generates four basic reports upon each invocation. These reports cover the areas of connect accounting, usage by login on a daily basis, command usage reported by daily and monthly totals, and a report of the last time users were logged in. the following table describes the four basic reports generated.

TABLE 32–2 Daily Accounting Reports

Report Type	Description
Daily Report	Shows line utilization by `tty` number.
Daily Usage Report	Indicates usage of system resources by users (listed in order of UID).
Daily Command Summary	Indicates usage of system resources by commands, listed in descending order of use of memory (in other words, the command that used the most memory is listed first). This same information is reported for the month with the monthly total command summary.
Last Login	Shows the last time each user logged in (arranged in chronological order).

Daily Report

This report gives information about each terminal line used. A sample daily report appears below.

```
Jul  7 02:30:02 1999  DAILY REPORT FOR mercury Page 1


from Wed Jul 07 02:30:02 1999
to   Thu Jul 08 02:30:02 1999
1       system boot
1       run-level 3
1       acctg on
1       runacct
1       acctcon

TOTAL DURATION IS 1384 MINUTES
LINE         MINUTES  PERCENT  # SESS  # ON  # OFF
/dev/pts/5   0        0        0       0     0
/dev/pts/6   0        0        0       0     1
/dev/pts/7   0        0        0       0     0
console      1337     97       1       1     1
pts/3        0        0        0       0     1
pts/4        0        0        0       0     1
pts/5        3        0        2       2     3
pts/6        232      17       5       5     5
pts/7        54       4        1       1     2
pts/8        0        0        0       0     1
pts/9        0        0        0       0     1
TOTALS       1625     --       9       9     16
```

The `from` and `to` lines specify the time period reflected in the report—the period from the time the last accounting report was generated until the time the current accounting report was generated. It is followed by a log of system reboots, shutdowns, power failure recoveries, and any other record dumped into `/var/adm/wtmpx` by the `acctwtmp` program. For more information, see `acct`(1M).

The second part of the report is a breakdown of line utilization. The `TOTAL DURATION` tells how long the system was in multiuser state (accessible through the terminal lines). The columns are described in the following table.

TABLE 32–3 Daily Report Data

Column	Description
`LINE`	The terminal line or access port.
`MINUTES`	The total number of minutes that the line was in use during the accounting period.

TABLE 32–3 Daily Report Data *(continued)*

Column	Description
PERCENT	The total number of MINUTES the line was in use, divided into the TOTAL DURATION.
# SESS	The number of times this port was accessed for a login session.
# ON	Identical to SESS. (This column does not have much meaning anymore. Previously, it listed the number of times that a port was used to log in a user.)
# OFF	This column reflects the number of times a user logs out and any interrupts that occur on that line. Generally, interrupts occur on a port when ttymon is first invoked after the system is brought to multiuser state. If the # OFF exceeds the # ON by a large factor, the multiplexer, modem, or cable is probably going bad, or there is a bad connection somewhere. The most common cause of this is an unconnected cable dangling from the multiplexer.

During real time, you should monitor /var/adm/wtmpx because it is the file from which the connect accounting is geared. If the wtmpx file grows rapidly, execute acctcon -l *file* < /var/adm/wtmpx to see which tty line is the noisiest. If interruption is occurring frequently, general system performance will be affected. Additionally, wtmp may become corrupted. To correct this, see "How to Fix a wtmpx File" on page 520.

Daily Usage Report

The daily usage report gives a breakdown of system resource utilization by user. A sample of this type of report appears below.

```
Jul  7 02:30:02 1999  DAILY USAGE REPORT FOR mercury Page 1

      LOGIN    CPU (MINS) KCORE-MINS  CONNECT (MINS) DISK  # OF  # OF # DISK FEE
UID   NAME   PRIME NPRIME PRIME NPRIME PRIME NPRIME  BLOCKS PROCS SESS SAMPLES
0     TOTAL  1     1      2017  717    785   840    660361  1067  9     7    20
0     root   1     1      1833  499    550   840    400443  408   2     1    0
1     daemon 0     0      0     0      0     0      400     0     0     1    0
2     bin    0     0      0     0      0     0      253942  0     0     1    0
3     sys    0     0      0     0      0     0      2       0     0     1    0
4     adm    0     0      46    83     0     0      104     280   0     1    0
5     uucp   0     0      74    133    0     0      1672    316   0     1    0
71    lp     0     0      0     2      0     0      3798    1     0     1    0
8198  ksm    0     0      8     0      0     0      0       6     1     0    0
```

(continued)

(Continuation)

```
52171 pjm    0     0      56    0      234   0      0       56    6     0    20
```

The data provided in the daily usage report is described in the following table.

TABLE 32–4 Daily Usage Report Data

Column	Description
UID	User identification number.
LOGIN NAME	Login name of the user. Identifies a user who has multiple login names.
CPU-MINS	Amount of time, in minutes, that the user's process used the central processing unit. Divided into PRIME and NPRIME (non-prime) utilization. The accounting system's version of this data is located in the /etc/acct/holidays file.
KCORE-MINS	A cumulative measure of the amount of memory in Kbyte segments per minute that a process uses while running. Divided into PRIME and NPRIME utilization.
CONNECT-MINS	Amount of time a user was logged into the system, or "real time." Divided into PRIME and NPRIME use. If these numbers are high while the # OF PROCS is low, you can conclude that the user logs in first thing in the morning and hardly touches the terminal the rest of the day.
DISK BLOCKS	Output from the acctdusg program, which runs and merges disk accounting programs and total accounting record (daytacct). (For accounting purposes, a block is 512 bytes.)
# OF PROCS	Number of processes invoked by the user. If large numbers appear, a user may have a shell procedure that has run out of control.
# OF SESS	Number of times a user logged on to the system.
# DISK SAMPLES	Number of times disk accounting was run to obtain the average number of DISK BLOCKS.
FEE	Often unused field that represents the total accumulation of units charged against the user by chargefee.

Daily Command Summary

The daily command summary report shows the system resource use by command. With this report, you can identify the most heavily used commands and, based on how those commands use system resources, gain insight on how best to tune the system. The format of the daily and monthly reports are virtually the same; however, the daily summary reports only on the current accounting period while the monthly summary reports on the start of the fiscal period to the current date. In other words, the monthly report is a cumulative summary that reflects the data accumulated since the last invocation of `monacct`.

These reports are sorted by `TOTAL KCOREMIN`, which is an arbitrary gauge but often a good one for calculating drain on a system.

A sample daily command summary appears below.

```
Jul  7 02:30:02 1999  DAILY COMMAND SUMMARY Page 1

                                   TOTAL COMMAND SUMMARY
COMMAND NUMBER    TOTAL     TOTAL     TOTAL    MEAN     MEAN     HOG    CHARS  BLOCKS
NAME     CMDS  KCOREMIN  CPU-MIN  REAL-MIN  SIZE-K  CPU-MIN FACTOR TRNSFD   READ

TOTALS   1067   2730.99    2.01    1649.38 1361.41   0.00  0.00    6253571  2305
sendmail   28   1085.87    0.05       0.24  23865.20  0.00  0.19     101544    39
admintoo    3    397.68    0.12    1132.96  3443.12   0.04  0.00     680220    83
sh        166    204.78    0.31     161.13  651.80    0.00  0.00     598158    20
nroff      12    167.17    0.14       0.24  1205.55   0.01  0.59     709048    22
find       10    151.27    0.27       2.72  563.40    0.03  0.10     877971  1580
acctdusg    3     87.40    0.13       2.74  698.29    0.04  0.05     883845   203
lp         10     74.29    0.05       0.22  1397.38   0.01  0.24     136460    57
expr       20     67.48    0.02       0.06  3213.24   0.00  0.34       6380     1
mail.loc    3     65.83    0.01       0.04  11285.60  0.00  0.15      24709    15
cmdtool     1     37.65    0.02      20.13  2091.56   0.02  0.00     151296     1
uudemon.  105     37.38    0.09       0.32  435.46    0.00  0.27      62130    17
csh         6     35.17    0.05      57.28  756.30    0.01  0.00     209560    13
col        12     31.12    0.06       0.26  523.00    0.00  0.23     309932     0
ntpdate    22     27.55    0.05      11.18  599.00    0.00  0.00      22419     0
uuxqt      44     18.66    0.04       0.06  417.79    0.00  0.74      32604     3
man        12     15.11    0.03       7.05  503.67    0.00  0.00      85266    47
.
.
.
```

The data provided, by column, in the daily command summary is described in the table below.

TABLE 32–5 Daily Command Summary

Column	Description
COMMAND NAME	Name of the command. Unfortunately, all shell procedures are lumped together under the name `sh` because only object modules are reported by the process accounting system. It's a good idea to monitor the frequency of programs called `a.out` or `core` or any other unexpected name. `acctcom` can be used to determine who executed an oddly named command and if superuser privileges were used.
NUMBER CMNDS	Total number of invocations of this particular command during prime time.
TOTAL KCOREMIN	Total cumulative measurement of the Kbyte segments of memory used by a process per minute of run time.
TOTAL CPU-MIN:	Total processing time this program has accumulated during prime time.
TOTAL REAL-MIN	Total real-time (wall-clock) minutes this program has accumulated.
MEAN SIZE-K	Mean of the `TOTAL KCOREMIN` over the number of invocations reflected by `NUMBER CMDS`.
MEAN CPU-MIN	Mean derived between the `NUMBER CMDS` and `TOTAL CPU-MIN`.
HOG FACTOR	Total CPU time divided by elapsed time. Shows the ratio of system availability to system use, providing a relative measure of total available CPU time consumed by the process during its execution.
CHARS TRNSFD	Total count of the number of characters pushed around by the read and write system calls. May be negative due to overflow.
BLOCKS READ	Total count of the physical block reads and writes that a process performed.

Monthly Command Summary

The monthly command summary is similar to the daily command summary. The only difference is that the monthly command summary shows totals accumulated since the last invocation of `monacct`. A sample report appears below.

```
Jul  6 02:30:03 1999  MONTHLY TOTAL COMMAND SUMMARY Page 1

                                    TOTAL COMMAND SUMMARY
COMMAND NUMBER     TOTAL    TOTAL     TOTAL    MEAN   MEAN   HOG    CHARS  BLOCKS
NAME      CMDS  KCOREMIN   CPU-MIN REAL-MIN SIZE-K CPU-MIN FACTOR TRNSFD   READ

TOTALS     771    483.70   0.94    8984.09  515.12   0.00  0.00   2248299   179

sh         105    155.41   0.23     429.58  667.94   0.00  0.00    491870     1
uudemon.    85     29.39   0.07       0.29  434.28   0.00  0.23     49630    14
acctcms      5     27.21   0.04       0.04  752.41   0.01  0.90    218880     1
ntpdate     17     21.30   0.04      14.10  605.73   0.00  0.00     18192     0
dtpad        1     19.69   0.01      10.87 2072.70   0.01  0.00     46992     8
sendmail    17     16.75   0.02       0.02  859.04   0.00  0.91      1965     0
acctprc      1     14.92   0.03       0.03  552.69   0.03  0.95    115584     0
uuxqt       34     14.78   0.03       0.04  426.29   0.00  0.92     25194     0
uusched     34     10.96   0.03       0.03  363.25   0.00  0.91     25194     0
sed         40     10.15   0.03       0.09  315.50   0.00  0.36     64162     2
man          5     10.08   0.02      57.58  555.05   0.00  0.00     25773     2
getent       1      7.68   0.01       0.02  921.60   0.01  0.40     20136     0
in.rlogi     5      7.65   0.01    4331.67  611.73   0.00  0.00     87440     0
cp          37      7.28   0.03       0.05  280.08   0.00  0.50      1739    36
date        27      7.24   0.02       0.03  329.12   0.00  0.65     23443     1
ls          15      7.05   0.01       0.02  503.33   0.00  0.79     14123     0
awk         19      6.94   0.02       0.06  372.04   0.00  0.32       666     0
rm          29      6.83   0.02       0.04  301.32   0.00  0.60      2348    17
```

See “Daily Command Summary ” on page 533 for a description of the data.

Last Login Report

This report gives the date when a particular login was last used. You can use this information to find unused logins and login directories that may be archived and deleted. A sample report appears below.

```
Jul  7 02:30:03 1999  LAST LOGIN Page 1
.
.
.
00-00-00  arimmer       00-00-00  lister        99-06-27  pjm
00-00-00  reception     00-00-00  smithe        99-06-27  ksm
00-00-00  release       00-00-00  smsc          99-06-27  root
00-00-00  resch         00-00-00  datab
```

Looking at the `pacct` File With `acctcom`

At any time, you can examine the contents of the `/var/adm/pacct`*n* files, or any file with records in the `acct.h` format, by using the `acctcom` program. If you don’t specify any files and don’t provide any standard input when you run this command, `acctcom` reads the `pacct` file. Each record read by `acctcom` represents information

about a dead process (active processes may be examined by running the `ps` command). The default output of `acctcom` provides the following information:

- Command name (pound (#) sign if it was executed with superuser privileges)
- User
- `tty` name (listed as `?` if unknown)
- Starting time
- Ending time
- Real time (in seconds)
- CPU time (in seconds)
- Mean size (in Kbytes)

The following information can be obtained by using options to `acctcom`:

- State of the `fork/exec` flag (1 for `fork` without `exec`)
- System exit status
- Hog factor
- Total `kcore` minutes
- CPU factor
- Characters transferred
- Blocks read

The following table describes the `acctcom` options.

TABLE 32–6 `acctcom` Options

Option	Description
`–a`	Shows some average statistics about the processes selected. (The statistics are printed after the output is recorded.)
`–b`	Reads the files backward, showing latest commands first. (This has no effect if reading standard input.)
`–f`	Prints the `fork/exec` flag and system exit status columns. (The output is an octal number.)
`–h`	Instead of mean memory size, shows the hog factor, which is the fraction of total available CPU time consumed by the process during its execution. Hog factor = *total_CPU_time/elapsed_time*.
`–i`	Prints columns containing the I/O counts in the output.

TABLE 32–6 `acctcom` Options *(continued)*

Option	Description
`–k`	Shows total `kcore minutes` instead of memory size.
`–m`	Shows mean core size (this is the default).
`–q`	Prints average statistics, not output records
`–r`	Shows CPU factor: *user_time*/(*system_time* + *user_tim*e).
`–t`	Shows separate system and user CPU times.
`–v`	Excludes column headings from the output.
`–C` *sec*	Shows only processes with total CPU time (system plus user) exceeding *sec* seconds.
`–e` *time*	Shows processes existing at or before *time*, given in the format *hr[:min[:sec]]*.
`–E` *time*	Shows processes starting at or before *time*, given in the format *hr[:min[:sec]]*. Using the same time for both `–S` and `–E`, show processes that existed at the time.
`–g` *group*	Shows only processes belonging to *group*.
`–H` *factor*	Shows only processes that exceed *factor*, where *factor* is the “hog factor” (see the `–h` option).
`–I` *chars*	Shows only processes transferring more characters than the cutoff number specified by *chars*.
`–l` *line*	Show only processes belonging to the terminal `/dev/`*line*.
`–n` *pattern*	Shows only commands matching *pattern* (a regular expression except that “+” means one or more occurrences).
`–o` *ofile*	Instead of printing the records, copys them in `acct.h` format to *ofile*.
`–O` *sec*	Shows only processes with CPU system time exceeding *sec* seconds.

TABLE 32–6 `acctcom` Options *(continued)*

Option	Description
–s *time*	Show processes existing at or after *time*, given in the format *hr*[:*min*[:*sec*]].
–S *time*	Show processes starting at or after *time*, given in the format *hr*[:*min*[:*sec*]].
–u *user*	Shows only processes belonging to *user*.

The `runacct` Program

The main daily accounting shell script, `runacct`, is normally invoked by `cron` outside of prime business hours. The `runacct` shell script processes connect, fee, disk, and process accounting files. It also prepares daily and cumulative summary files for use by `prdaily` and `monacct` for billing purposes.

The `runacct` shell script takes care not to damage files if errors occur. A series of protection mechanisms are used that attempt to recognize an error, provide intelligent diagnostics, and complete processing in such a way that `runacct` can be restarted with minimal intervention. It records its progress by writing descriptive messages into the file `active`. (Files used by `runacct` are assumed to be in the `/var/adm/acct/nite` directory, unless otherwise noted.) All diagnostic output during the execution of `runacct` is written into `fd2log`.

When `runacct` is invoked, it creates the files `lock` and `lock1`. These files are used to prevent simultaneous execution of `runacct`. The `runacct` program prints an error message if these files exist when it is invoked. The `lastdate` file contains the month and day `runacct` was last invoked, and is used to prevent more than one execution per day. If `runacct` detects an error, a message is written to the console, mail is sent to `root` and `adm`, locks may be removed, diagnostic files are saved, and execution is ended. For instructions on how to start `runacct` again, see “How to Restart `runacct`” on page 521.

To allow `runacct` to be restartable, processing is broken down into separate re-entrant states. The file `statefile` is used to keep track of the last state completed. When each state is completed, `statefile` is updated to reflect the next state. After processing for the state is complete, `statefile` is read and the next state is processed. When `runacct` reaches the `CLEANUP` state, it removes the locks and ends. States are executed as shown in the table below.

TABLE 32–7 `runacct` States

State	Description
SETUP	The command `turnacct switch` is executed to create a new `pacct` file. The process accounting files in `/var/adm/pacct`*n* (except for the `pacct` file) are moved to `/var/adm/Spacctn.`*MMDD*. The `/var/adm/wtmpx` file is moved to `/var/adm/acct/nite/wtmp.`*MMDD* (with the current time record added on the end) and a new `/var/adm/wtmp` is created. `closewtmp` and `utmp2wtmp` add records to `wtmp.`*MMDD* and the new `wtmpx` to account for users currently logged in.
WTMPFIX	The `wtmpfix` program checks the `wtmp.`*MMDD* file in the `nite` directory for accuracy. Because some date changes will cause `acctcon` to fail, `wtmpfix` attempts to adjust the time stamps in the `wtmpx` file if a record of a date change appears. It also deletes any corrupted entries from the `wtmpx` file. The fixed version of `wtmp.`*MMDD* is written to `tmpwtmp`.
CONNECT	The `acctcon` program is used to record connect accounting records in the file `ctacct.`*MMDD*. These records are in `tacct.h` format. In addition, `acctcon` creates the `lineuse` and `reboots` files. The `reboots` file records all the boot records found in the `wtmpx` file.
PROCESS	The `acctprc` program is used to convert the process accounting files, `/var/adm/Spacct`*n.MMDD*, into total accounting records in `ptacct`*n.MMDD*. The `Spacct` and `ptacct` files are correlated by number so that if `runacct` fails, the `Spacct` files will not be processed.
MERGE	The `acctmerg` program merges the process accounting records with the connect accounting records to form `daytacct`.
FEES	The `acctmerg` program merges ASCII `tacct` records from the `fee` file into `daytacct`.
DISK	If the `dodisk` procedure has been run, producing the `disktacct` file, the `DISK` program merges the file into `daytacct` and moves `disktacct` to `/tmp/disktacct.`*MMDD*.
MERGETACCT	The `acctmerg` program merges `daytacct` with `sum/tacct`, the cumulative total accounting file. Each day, `daytacct` is saved in `sum/tacct.`*MMDD*, so that `sum/tacct` can be recreated if it is corrupted or lost.

TABLE 32–7 `runacct` States *(continued)*

State	Description
`CMS`	The `acctcms` program is run several times. `acctcms` is first run to generate the command summary using the `Spacct`*n* files and write it to `sum/daycms`. The `acctcms` program is then run to merge `sum/daycms` with the cumulative command summary file `sum/cms`. Finally, `acctcms` is run to produce the ASCII command summary files, `nite/daycms` and `nite/cms`, from the `sum/daycms` and `sum/cms` files, respectively. The `lastlogin` program is used to create the `/var/adm/acct/sum/loginlog` log file, the report of when each user last logged in. (If `runacct` is run after midnight, the dates showing the time last logged in by some users will be incorrect by one day.)
`USEREXIT`	Any installation-dependent (local) accounting program can be included at this point. `runacct` expects it to be called `/usr/lib/acct/runacct.local`.
`CLEANUP`	Cleans up temporary files, runs `prdaily` and saves its output in `sum/rpt.`*MMDD*, removes the locks, then exits.

Caution - When restarting `runacct` in the `CLEANUP` state, remove the last `ptacct` file because it will not be complete.

Accounting Files

The `/var/adm` directory structure contains the active data collection files.

The following table describes the accounting related files in the `/var/adm` directory.

TABLE 32–8 Files in the `/var/adm` Directory

File	Description
`dtmp`	Output from the `acctdusg` program
`fee`	Output from the `chargefee` program, ASCII `tacct` records
`pacct`	Active process accounting file

TABLE 32–8 Files in the /var/adm Directory *(continued)*

File	Description
pacct*n*	Process accounting files switched using turnacct
Spacct*n.MMDD*	Process accounting files for *MMDD* during execution of runacct

The /var/adm/acct directory contains the nite, sum, and fiscal directories, which contain the actual data collection files. For example, the nite directory contains files that are reused daily by the runacct procedure. A brief summary of the files in the /var/adm/acct/nite directory follows.

TABLE 32–9 Files in the /var/adm/acct/nite Directory

File	Description
active	Used by runacct to record progress and print warning and error messages
active*MMDD*	Same as active after runacct detects an error
cms	ASCII total command summary used by prdaily
ctacct.*MMDD*	Connect accounting records in tacct.h format
ctmp	Output of acctcon1 program, connect session records in ctmp.h format (acctcon1 and acctcon2 are provided for compatibility purposes)
daycms	ASCII daily command summary used by prdaily
daytacct	Total accounting records for one day in tacct.h format
disktacct	Disk accounting records in tacct.h format, created by the dodisk procedure
fd2log	Diagnostic output during execution of runacct
lastdate	Last day runacct executed (in date +%m%d format)
lock	Used to control serial use of runacct
lineuse	tty line usage report used by prdaily

TABLE 32–9 Files in the /var/adm/acct/nite Directory *(continued)*

File	Description
log	Diagnostic output from acctcon
log.*MMDD*	Same as log after runacct detects an error
owtmp	Previous day's wtmpx file
reboots	Beginning and ending dates from wtmpx and a listing of reboots
statefile	Used to record current state during execution of runacct
tmpwtmp	wtmpx file corrected by wtmpfix
wtmperror	Place for wtmpfix error messages
wtmperror.*MMDD*	Same as wtmperror after runacct detects an error
wtmp.*MMDD*	runacct's copy of the wtmpx file

The sum directory contains the cumulative summary files updated by runacct and used by monacct. A brief summary of the files in the /var/adm/acct/sum directory is shown in the following table.

TABLE 32–10 Files in the /var/adm/acct/sum Directory

File	Description
cms	Total command summary file for current fiscal period in internal summary format
cmsprev	Command summary file without latest update
daycms	Command summary file for the day's usage in internal summary format
loginlog	Record of last date each user logged on; created by lastlogin and used in the prdaily program
rprt.*MMDD*	Saved output of prdaily program
tacct	Cumulative total accounting file for current fiscal period

TABLE 32–10 Files in the `/var/adm/acct/sum` Directory *(continued)*

File	Description
`tacctprev`	Same as `tacct` without latest update
`tacct.`*MMDD*	Total accounting file for *MMDD*

The fiscal directory contains periodic summary files created by `monacct`. A brief description of the files in the `/var/adm/acct/fiscal` directory is shown in the following table.

TABLE 32–11 Files in the `/var/adm/acct/fiscal` Directory

File	Description
`cms`*n*	Total command summary file for fiscal period *n* in internal summary format
`fiscrpt`*n*	Report similar to `rprt`*n* for fiscal period *n*
`tacct`*n*	Total accounting file for fiscal period *n*

Files Produced by `runacct`

The most useful files produced by `runacct` (found in `/var/adm/acct`) are shown in the following table.

TABLE 32–12 Files Produced by `runacct`

File	Description
`nite/lineuse`	`runacct` calls `acctcon` to gather data on terminal line usage from `/var/adm/acct/nite/tmpwtmp` and writes the data to `/var/adm/acct/nite/lineuse`. `prdaily` uses this data to report line usage. This report is especially useful for detecting bad lines. If the ratio between the number of logouts to logins is greater than about three to one, there is a good possibility that the line is failing.
`nite/daytacct`	This file is the total accounting file for the day in `tacct.h` format.
`sum/tacct`	This file is the accumulation of each day's `nite/daytacct` and can be used for billing purposes. It is restarted each month or fiscal period by the `monacct` procedure.
`sum/daycms`	`runacct` calls `acctcms` to process the data about the commands used during the day. This information is stored in `/var/adm/acct/sum/daycms`. It contains the daily command summary. The ASCII version of this file is `/var/adm/acct/nite/daycms`.
`sum/cms`	This file is the accumulation of each day's command summaries. It is restarted by the execution of `monacct`. The ASCII version is `nite/cms`.
`sum/loginlog`	`runacct` calls `lastlogin` to update the last date logged in for the logins in `/var/adm/acct/sum/loginlog`. `lastlogin` also removes from this file logins that are no longer valid.
`sum/rprt.`*MMDD*	Each execution of `runacct` saves a copy of the daily report that was printed by `prdaily`.

CHAPTER 33

Managing System Performance Topics

This section provides instructions for managing system performance. This section contains these chapters.

Chapter 34	Provides overview information about performance topics.
Chapter 35	Provides step-by-step instructions for using process commands to enhance system performance.
Chapter 36	Provides step-by-step instructions for using `vmstat`, `sar`, and disk utilization commands to monitor performance.

CHAPTER 34

System Performance (Overview)

Getting good performance from a computer or network is an important part of system administration. This chapter is an overview of some of the factors that contribute to maintaining and managing the performance of the computer systems in your care.

This is a list of the overview information in this chapter.

- "Where to Find System Performance Tasks" on page 549
- "System Performance and System Resources" on page 549
- "Processes and System Performance" on page 551
- "About Monitoring Performance" on page 553

What's New in Managing System Performance?

This section describes new Solaris 8 features in the area of managing system performance.

SPARC: `busstat`

A new system monitoring tool, `busstat`, provides command line access to the bus-related hardware performance counters in the system. It enables the gathering of system-wide bus performance statistics directly from the system hardware. The current list of supported hardware is SBus, AC and PCI devices. These are all SPARC system devices. Currently, there are no IA supported devices.

The busstat command enables the measurement of system-wide statistics such as memory bank reads/writes, clock cycles, number of interrupts, streaming DVMA read/write transfers etc.

Superuser can use busstat to program these counters. Ordinary users can only read counters programmed previously by superuser.

The busstat command lists the devices in a system that are found to support these hardware performance counters. If no supported devices are found in the system, the following message is displayed:

```
busstat: No devices available in system.
```

See busstat(1M) for more information on using this monitoring tool.

The cpustat and cputrack Commands

You can use the new cpustat and cputrack commands for monitoring the performance of a system or a process.

The cpustat command gathers system-wide CPU information. This command must be run by the superuser. The cputrack command is similar to the truss command for displaying information about an application or a process. This command can be run by regular users.

Developers can create their own versions of these monitoring tools by using the same library APIs that were used to build the cpustat command.

See cpustat(1M) and cputrack(1) for more information.

prstat

The prstat command displays information about active processes on the system. You can specify whether you want information on specific processes, UIDs, CPU IDs, or processor sets. By default, prstat displays information about all processes sorted by CPU usage.

You can display detailed process microstate accounting information with prstat -m, which provides the percentage of time the process has spent processing system traps, text page faults, data page faults, and waiting for CPU, also known as CPU latency time.

See prstat(1M) for more information.

Obsolete Interprocess Communication Parameters

The Interprocess Communication (IPC) Message facility has been made more scalable in the Solaris 8 release by using `kmem_alloc`(9F) rather than `rmalloc`(9F) to allocate message text.

Therefore, the previously-documented `msginfo_msgssz` and `msginfo_msgseg` tunables, which were artifacts of the `rmalloc`-based implementation, are obsolete in this release.

Where to Find System Performance Tasks

Use these references to find step-by-step instructions for monitoring system performance.

- Chapter 35
- Chapter 36

System Performance and System Resources

The performance of a computer system depends upon how the system uses and allocates its resources. It is important to monitor your system's performance on a regularly so that you know how it behaves under normal conditions. You should have a good idea of what to expect, and be able to recognize a problem when it occurs.

System resources that affect performance are described in the following table.

System Resource	Description
Central processing unit (CPU)	The CPU processes instructions, fetching instructions from memory and executing them.
Input/output (I/O) devices	I/O devices transfer information into and out of the computer. Such a device could be a terminal and keyboard, a disk drive, or a printer.
Memory	Physical (or main) memory is the amount of memory (RAM) on the system.

Chapter 36 describes the tools that display statistics about the activity and the performance of the computer system.

Sources of Performance Tuning Information

Performance is a broad subject that can't be adequately covered in these chapters. Sun provides performance tuning courses, online performance tuning information and several books are available that cover various aspects of improving performance and tuning your system or network.

For ...	Go To ...
Performance tuning classes	`http://suned.sun.com`
Online performance tuning information	`http://www.sun.com/sun-on-net/performance`
Ordering performance tuning documentation by Sun Microsystems Press such as *Resource Management*	`http://www.sun.com/books/blueprints.series.html`

System or network performance tuning is covered in the following books:

- *Resource Management* by Richard McDougall, Adrian Cockcroft, Evert Hoogendoorn, Enrique Vargas, Tom Bialaski, Sun Microsystems Press, ISBN 0-13-025855-5
- *Sun Performance and Tuning: SPARC and Solaris*, by Adrian Cockcroft, Sun Microsystems Press/PRT Prentice Hall, ISBN 0-13-149642-3
- *System Performance Tuning*, by Mike Loukides, O'Reilly & Associates, Inc.
- *Managing NFS and NIS*, by Hal Stern, O'Reilly & Associates, Inc.

Processes and System Performance

Terms related to processes are described in the table below.

TABLE 34–1 Process Terminology

Term	Description
Process	An instance of program in execution.
Lightweight process (LWP)	Is a virtual CPU or execution resource. LWPs are scheduled by the kernel to use available CPU resources based on their scheduling class and priority. LWPs include a kernel thread, which contains information that has to be in memory all the time and an LWP, which contains information that is swappable.
Application thread	A series of instructions with a separate stack that can execute independently in a user's address space. They can be multiplexed on top of LWPs.

A process can consist of multiple LWPs and multiple application threads. The kernel schedules a kernel-thread structure, which is the scheduling entity in the SunOS environment. Various process structures are described in the table below.

TABLE 34–2 Process Structures

Structure	Description
`proc`	Contains information that pertains to the whole process and has to be in main memory all the time.
`kthread`	Contains information that pertains to one LWP and has to be in main memory all the time.
`user`	Contains the per process information that is swappable.
`klwp`	Contains the per LWP process information that is swappable.

The figure below illustrates the relationship of these structures.

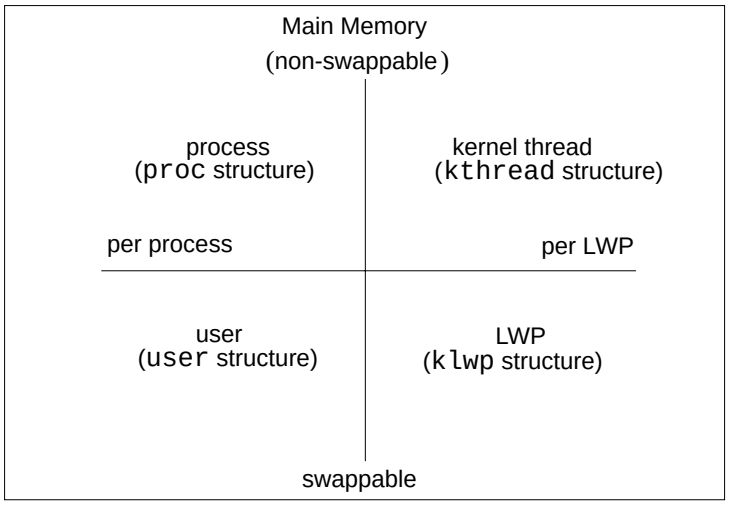


Figure 34–1 Process Structures

Most process resources are accessible to all the threads in the process. Almost all process virtual memory is shared. A change in shared data by one thread is available to the other threads in the process.

Commands for Managing Processes

The table below describes commands for managing processes.

TABLE 34–3 Commands for Managing Processes

Use This Command ...	To ...
`ps`(1), `pgrep`(1), and `prstat`(1M)	Check the status of active processes on a system, as well as display detailed information about the processes
`dispadmin`(1M)	List default scheduling policies
`priocntl`(1)	Assign processes to a priority class and manage process priorities
`nice`(1)	Change the priority of a timesharing process

TABLE 34–3 Commands for Managing Processes *(continued)*

Another feature enables the control of process groups over processor sets. Using processor sets means process groups can bind to a group of processors rather than to just a single processor. The `/usr/sbin/psrset` command gives a system administrator control over the creation and management of processor sets. See `psrset`(1M) for more information.

See Chapter 35 for more information about commands for managing processes.

About Monitoring Performance

While your computer is running, counters in the operating system are incremented to keep track of various system activities. System activities that are tracked are:

- Central processing unit (CPU) utilization
- Buffer usage
- Disk and tape input/output (I/O) activity
- Terminal device activity
- System call activity
- Context switching
- File access
- Queue activity
- Kernel tables
- Interprocess communication
- Paging
- Free memory and swap space
- Kernel Memory Allocation (KMA)

Monitoring Tools

The Solaris software provides several tools to help you keep track of how your system is performing. These include:

TABLE 34–4 Performance Monitoring Tools

The ...	Enable(s) You To ...	For More Information, See ...
`sar` and `sadc` utilities	Collect and report on system activity data	Chapter 36
`ps` and `prstat` commands	Display information about active processes	Chapter 35
`vmstat` and `iostat` commands	Summarize system activity data, such as virtual memory statistics, disk usage, and CPU activity	Chapter 36
`swap` command	Display information about available swap space on your system	"Configuring Additional Swap Space (Tasks)" in *System Administration Guide, Volume 1*
`netstat` and `nfsstat` commands	Display information about network performance	
Sun Enterprise SyMON	Collect system activity data on Sun's Enterprise™ level systems	*Sun Enterprise SyMON 2.0.1 Software User's Guide*

CHAPTER 35

Managing Processes (Tasks)

This chapter describes the procedures for managing system processes. This is a list of the step-by-step instructions in this chapter.

- "How to List Processes" on page 557
- "How to Display Information About Processes" on page 560
- "How to Control Processes" on page 563
- "How to Kill a Process" on page 565
- "How to Display Basic Information About Process Classes" on page 566
- "How to Display the Global Priority of a Process" on page 567
- "How to Designate a Process Priority" on page 567
- "How to Change Scheduling Parameters of a Timeshare Process" on page 568
- "How to Change the Class of a Process" on page 569
- "How to Change the Priority of a Process" on page 571

Displaying Information About Processes

This section describes commands used to manage process information.

The ps Command

The ps command enables you to check the status of active processes on a system, as well as display technical information about the processes. This data is useful for such administrative tasks as determining how to set process priorities.

Depending on which options you use, ps reports the following information:

- Current status of the process
- Process ID
- Parent process ID
- User ID
- Scheduling class
- Priority
- Address of the process
- Memory used
- CPU time used

The table below describes some of the fields reported by the ps command. The fields displayed depend on which option you choose. See ps(1) for a description of all available options.

TABLE 35–1 Summary of Fields in ps Reports

Field	Description
UID	The effective user ID of the process's owner.
PID	The process ID.
PPID	The parent process's ID.
C	The processor utilization for scheduling. This field is not displayed when the –c option is used.
CLS	The scheduling class to which the process belongs: real-time, system, or timesharing. This field is included only with the –c option.
PRI	The kernel thread's scheduling priority. Higher numbers mean higher priority.
NI	The process's nice number, which contributes to its scheduling priority. Making a process "nicer" means lowering its priority.
ADDR	The address of the proc structure.

TABLE 35–1 Summary of Fields in ps Reports *(continued)*

Field	Description
SZ	The virtual address size of the process.
WCHAN	The address of an event or lock for which the process is sleeping.
STIME	The starting time of the process (in hours, minutes, and seconds).
TTY	The terminal from which the process (or its parent) was started. A question mark indicates there is no controlling terminal.
TIME	The total amount of CPU time used by the process since it began.
CMD	The command that generated the process.

▼ How to List Processes

To list all the processes being executed on a system, use the ps command.

```
$ ps [-ef]
```

ps	Displays only the processes associated with your login session.
–ef	Displays full information about all the processes being executed on the system.

Example—Listing Processes

The following example shows output from the ps command when no options are used.

```
$ ps
   PID TTY      TIME COMD
  1664 pts/4    0:06 csh
  2081 pts/4    0:00 ps
```

The following example shows output from ps –ef. This shows that the first process executed when the system boots is sched (the swapper) followed by the init process, pageout, and so on.

```
$ ps -ef
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   May 05 ?       0:04 sched
    root     1     0  0   May 05 ?      10:48 /etc/init -
    root     2     0  0   May 05 ?       0:00 pageout
    root     3     0  0   May 05 ?      43:21 fsflush
    root   238     1  0   May 05 ?       0:00 /usr/lib/saf/sac -t 300
    root   115     1  0   May 05 ?       0:10 /usr/sbin/rpcbind
    root   158     1  0   May 05 ?       0:00 /usr/lib/autofs/autom...
    root   134     1  0   May 05 ?       0:12 /usr/sbin/inetd -s
    root   107     1  0   May 05 ?      11:49 /usr/sbin/in.routed -q
    root   117     1  5   May 05 ?     899:32 /usr/sbin/keyserv
    root   125     1  0   May 05 ?       0:00 /usr/sbin/kerbd
    root   123     1  0   May 05 ?       4:17 /usr/sbin/nis_cachemgr
    daemon 137     1  0   May 05 ?       0:00 /usr/lib/nfs/statd
    root   139     1  0   May 05 ?       0:02 /usr/lib/nfs/lockd
    root   159     1 50   May 05 ?    8243:36 /usr/sbin/automount
    root   162     1  0   May 05 ?       0:07 /usr/sbin/syslogd
    root   181     1  0   May 05 ?       0:03 /usr/sbin/nscd...
    root   169     1  0   May 05 ?       5:09 /usr/sbin/cron
    root   191     1  0   May 05 ?       0:00 /usr/lib/lpsched
    root   210     1  0   May 05 ?       0:01 /usr/sbin/vold
    root   200     1  0   May 05 ?       0:08 /usr/lib/sendmail -bd -q1h
    root  4942     1  0   May 17 console 0:00 /usr/lib/saf/ttymon...
    root   208     1  0   May 05 ?       0:00 /usr/lib/utmpd
    root   241   238  0   May 05 ?       0:00 /usr/lib/saf/ttymon
    root  5748   134  0 17:09:49 ?       0:01 in.rlogind
    root  5750  5748  0 17:09:52 pts/0   0:00 -sh
    root  5770  5750  2 17:23:39 pts/0   0:00 ps -ef
```

The /proc File System and Commands

In addition, process tools are available in the /usr/proc/bin directory that display highly detailed information about the processes listed in the /proc directory, also known as the process file system (PROCFS). Images of active processes are stored here by their process ID number.

The process tools are similar to some options of the ps command, except that the output provided by the tools is more detailed. In general, the process tools:

- Display more details about processes, such as fstat and fcntl information, working directories, and trees of parent and child processes
- Provide control over processes, allowing users to stop or resume them

Displaying Information About Processes (/proc Tools)

You can display detailed, technical information about active processes by using some of the process tool commands contained in /usr/proc/bin. The table below lists these process tools. Refer to proc(1) for more information.

TABLE 35–2 /usr/proc/bin Process Tools That Display Information

Process Tool	What It Displays
pcred	Credentials
pfiles	fstat and fcntl information for open files in a process
pflags	/proc tracing flags, pending and held signals, and other status information
pldd	Dynamic libraries linked into a process
pmap	Address space map
psig	Signal actions
pstack	Hex+symbolic stack trace
ptime	Process time using microstate accounting
ptree	Process trees that contain the process
pwait	Status information after a process terminates
pwdx	Current working directory for a process

Note - To avoid typing long command names, add the process tool directory to your PATH variable. This enables you to run process tools by entering only the last part of each file name (for example, pwdx instead of /usr/proc/bin/pwdx).

▼ How to Display Information About Processes

1. **(Optional) Use output from the** `pgrep` **command to obtain the identification number of the process you want to display more information about.**

```
# pgrep process
```

process	Name of the process you want to display more information about.

The process identification number is in the first column of the output.

2. **Use the appropriate** `/usr/bin/proc` **command to display the information you need.**

```
# /usr/proc/bin/pcommand pid
```

pcommand	Process tool command you want to run. Table 35–2 lists these commands.
pid	Identification number of a process.

Examples—Displaying Information About Processes

The following example shows how to use process tool commands to display more information about an `lpsched` process. First, the `/usr/proc/bin` path is defined to avoid typing long process tool commands. Next, the identification number for `lpsched` is obtained. Finally, output from three process tool commands is shown.

```
# PATH=$PATH:/usr/proc/bin
# export PATH 1
# ps -e | grep lpsched 2
207 ?        0:00 /usr/lib/lpsched
# pwdx 191 3
207:    /
# ptree 191 4
207   /usr/lib/lpsched
# pfiles 191 5
207:    /usr/lib/lpsched
  Current rlimit: 4096 file descriptors
   0: S_IFIFO mode:0000 dev:179,0 ino:70 uid:0 gid:0 size:0
      O_RDWR
   1: S_IFIFO mode:0000 dev:179,0 ino:70 uid:0 gid:0 size:0
```

(continued)

```
      O_RDWR
   3: S_IFCHR mode:0666 dev:32,8 ino:11446 uid:0 gid:3 rdev:21,0
      O_WRONLY FD_CLOEXEC
   4: S_IFDOOR mode:0444 dev:183,0 ino:59515 uid:0 gid:0 size:0
      O_RDONLY|O_LARGEFILE FD_CLOEXEC  door to nscd[201]
   5: S_IFREG mode:0664 dev:32,9 ino:1330 uid:71 gid:8 size:0
      O_WRONLY
```

1. Adds the `/usr/proc/bin` directory to the `PATH` variable.
2. Obtains the process identification number for `lpsched`.
3. Displays the current working directory for `lpsched`.
4. Displays the process tree containing `lpsched`.
5. Displays `fstat` and `fcntl` information.

The following example shows output from the `pwait` command, which waits until a process terminates, then displays information about what happened. The following example shows output from the `pwait` command after a Command Tool window was exited.

```
$ ps -e | grep cmdtool
  273 console 0:01 cmdtool
  277 console 0:01 cmdtool
  281 console 0:01 cmdtool
$ pwait -v 281
281: terminated, wait status 0x0000
```

Controlling Processes (`/proc` Tools)

You can control some aspects of processes by using some of the process tools contained in `/usr/proc/bin`. The table below lists these process tools. Refer to `proc`(1) for detailed information about process tools.

TABLE 35–3 Process Tools

Tools That Control Processes	**What the Tools Do**
`/usr/proc/bin/pstop` *pid*	Stops the process
`/usr/proc/bin/prun` *pid*	Restarts the process
`/usr/proc/bin/ptime` *pid*	Times the process using microstate accounting
`/usr/proc/bin/pwait` [–v] *pid*	Waits for specified processes to terminate
Tools That Display Process Details	What the Tools Display
`/usr/proc/bin/pcred` *pid*	Credentials
`/usr/proc/bin/pfiles` *pid*	`fstat` and `fcntl` information for open files
`/usr/proc/bin/pflags` *pid*	`/proc` tracing flags, pending and held signals, and other status information for each `lwp`
`/usr/proc/bin/pldd` *pid*	Dynamic libraries linked into each process
`/usr/proc/bin/pmap` *pid*	Address space map
`/usr/proc/bin/psig` *pid*	Signal actions
`/usr/proc/bin/pstack` *pid*	Hex+symbolic stack trace for each `lwp`
`/usr/proc/bin/ptree` *pid*	Process trees containing specified pids
`/usr/proc/bin/pwdx` *pid*	Current working directory

In these commands, *pid* is a process identification number. You can obtain this number by using the `ps -ef` command.

Chapter 35 describes how to use the process tool commands to perform selected system administration tasks, such as displaying details about processes, and starting

and stopping them. A more detailed description of the process tools can be found in `proc`(1).

If a process becomes trapped in an endless loop, or if it takes too long to execute, you may want to stop (kill) the process. See Chapter 35 for more information about stopping processes using the `pkill` command.

The previous flat `/proc` file system has been restructured into a directory hierarchy that contains additional sub-directories for state information and control functions.

It also provides a watchpoint facility that is used to remap read/write permissions on the individual pages of a process's address space. This facility has no restrictions and is MT-safe.

The new `/proc` file structure provides complete binary compatibility with the old `/proc` interface except that the new watchpoint facility cannot be used with the old interface.

Debugging tools have been modified to use `/proc`'s new watchpoint facility, which means the entire watchpoint process is faster.

The following restrictions have been removed when setting watchpoints using the `dbx` debugging tool:

- Setting watchpoints on local variables on the stack due to SPARC register windows
- Setting watchpoints on multi-threaded processes

See `proc`(4), `core`(4), and `adb`(1) for more information.

Note - To avoid typing long command names, add the process tool directory to your `PATH` variable. This allows you to run process tools by entering only the last part of each file name (for example, `prun` instead of `/usr/proc/bin/prun`).

▼ How to Control Processes

1. **(Optional) Use output from the `ps` command to obtain the identification number of the process you want to display more information about.**

```
# pgrep process
```

process	Name of the process you want to display more information about.

The process identification number is in the first column of the output.

2. **Use the appropriate `/usr/proc/bin` command to control the process.**

```
# /usr/proc/bin/pcommand PID
```

pcommand	Process tool command you want to run. Table 35–3 lists these commands.
PID	Identification number of a process.

3. Verify the process status using the `ps` command.

```
# pgrep PID
```

Example—Controlling Processes

The following example shows how to use process tools to stop and restart Print Tool.

```
# PATH=$PATH:/usr/proc/bin
# export PATH 1
# ps -e | grep print* 2
264 console 0:03 printtool
# pstop 264 3
# prun 264 4
# ps | grep 264
264 console 0:03 printtool
#
```

1. Adds the `/usr/proc/bin` directory to the `PATH` variable.
2. Obtains the process identification number for Print Tool.
3. Stops the Print Tool process.
4. Restarts the Print Tool process.

Killing a Process (`pkill`)

Sometimes it is necessary to stop (kill) a process. The process may be in an endless loop, or you may have started a large job that you want to stop before it is completed. You can kill any process that you own, and superuser can kill any processes in the system except for those with process IDs `0`, `1`, `2`, `3`, and `4`.

Refer to `pkill`(1) for more detailed information.

▼ How to Kill a Process

1. **(Optional) To kill a process belonging to another user, become superuser.**

2. **(Optional) Use output from the** `pgrep` **command to obtain the identification number of the process you want to display more information about.**

```
$ pgrep process
```

process	Name of the process you want to display more information about.

The process identification number is in the first column of the output.

3. **Use the** `pkill` **command to stop the process.**

```
$ pkill [-9] PID ...
```

–9	Ensures that the process terminates promptly.
PID . . .	ID of the process or processes to stop.

4. **Use the** `pgrep` **command to verify that the process has been stopped.**

```
$ pgrep PID ...
```

Managing Process Class Information

The listing below shows which classes are configured on your system, and the user priority range for the timesharing class. The possible classes are:

- System (SYS)
- Interactive (IA)
- Real-time (RT)
- Timesharing (TS)

- The user-supplied priority ranges from -20 to +20.
- The priority of a process is inherited from the parent process. This is referred to as the *user-mode* priority.
- The system looks up the user-mode priority in the timesharing dispatch parameter table and adds in any `nice` or `priocntl` (user-supplied) priority and ensures a 0-59 range to create a *global* priority.

Changing the Scheduling Priority of Processes With `priocntl`

The scheduling priority of a process is the priority it is assigned by the process scheduler, according to scheduling policies. The `dispadmin` command lists the default scheduling policies.

The `priocntl`(1) command can be used to assign processes to a priority class and to manage process priorities. See the section called "How to Designate a Process Priority" on page 567 for instructions on using the `priocntl` command to manage processes.

▼ How to Display Basic Information About Process Classes

You can display process class and scheduling parameters with the `priocntl -l` command.

```
$ priocntl -l
```

Example—Getting Basic Information About Process Classes

The following example shows output from the `priocntl -l` command.

```
# priocntl -l
CONFIGURED CLASSES
==================

SYS (System Class)

TS (Time Sharing)
        Configured TS User Priority Range: -60 through 60

IA (Interactive)
        Configured IA User Priority Range: -60 through 60

RT (Real Time)
```

(continued)

```
        Maximum Configured RT Priority: 59
```

▼ How to Display the Global Priority of a Process

You can display the global priority of a process by using the `ps` command.

```
$ ps -ecl
```

The global priority is listed under the `PRI` column.

Example—Displaying the Global Priority of a Process

The following example shows output from `ps -ecl`. Data in the `PRI` column show that `pageout` has the highest priority, while `sh` has the lowest.

```
$ ps -ecl
 F S UID PID  PPID CLS PRI  ADDR      SZ  WCHAN    TTY      TIME   COMD
19 T 0   0    0    SYS 96   f00d05a8   0           ?        0:03  sched
 8 S 0   1    0    TS  50   ff0f4678 185  ff0f4848 ?       36:51   init
19 S 0   2    0    SYS 98   ff0f4018   0  f00c645c ?        0:01 pageout
19 S 0   3    0    SYS 60   ff0f5998   0  f00d0c68 ?      241:01 fsflush
 8 S 0   269  1    TS  58   ff0f5338 303  ff49837e ?        0:07    sac
 8 S 0   204  1    TS  43   ff2f6008  50  ff2f606e console  0:02     sh
```

▼ How to Designate a Process Priority

1. **Become superuser.**

2. **Start a process with a designated priority.**

```
# priocntl -e -c class -m userlimit -p pri command_name
```

–e	Executes the command.
–c *class*	Specifies the class within which to run the process. The default classes are TS (timesharing) or RT (real-time).
–m *userlimit*	Specifies the maximum amount you can raise or lower your priority, when using the –p option.
–p *pri command_name*	Lets you specify the relative priority in the RT class, for a real-time thread. For a timesharing process, the –p option lets you specify the user-supplied priority which ranges from -20 to +20.

3. Verify the process status by using the `ps -ecl` command.

```
# ps -ecl | grep command_name
```

Example—Designating a Priority

The following example starts the `find` command with the highest possible user-supplied priority.

```
# priocntl -e -c TS -m 20 -p 20 find . -name core -print
# ps -ecl | grep find
```

▼ How to Change Scheduling Parameters of a Timeshare Process

1. Become superuser.

2. Change the scheduling parameter of a running timeshare process.

```
# priocntl -s -m userlimit [-p userpriority] -i idtype idlist
```

–s	Lets you set the upper limit on the user priority range and change the current priority.
–m *userlimit*	Specifies the maximum amount you can raise or lower your priority, when using the –p option.
–p *userpriority*	Allows you to designate a priority.
–i*idtype idlist*	Uses a combination of *idtype* and *idlist* to identify the process. The *idtype* specifies the type of ID, such as PID or UID.

3. **Verify the process status by using the** `ps -ecl` **command.**

```
# ps -ecl | grep idlist
```

Example—Changing Scheduling Parameters of a Timeshare Process

The following example executes a command with a 500-millisecond time slice, a priority of 20 in the RT class, and a global priority of 120.

```
# priocntl -e -c RT -t 500 -p 20 myprog
# ps -ecl | grep myprog
```

▼ How to Change the Class of a Process

1. **(Optional) Become superuser.**

Note - You must be superuser or working in a real-time shell to change processes from, or to, real-time processes.

---

2. **Change the class of a process.**

```
# priocntl -s -c class -i idtype idlist
```

`–s`	Lets you set the upper limit on the user priority range and change the current priority.
`–c` *class*	Specifies the class, `TS` or `RT`, to which you are changing the process.
`–i` *idtype idlist*	Uses a combination of *idtype* and *idlist* to identify the process. The *idtype* specifies the type of `ID`, such as `PID` or `UID`.

3. Verify the process status by using the `ps -ecl` command.

```
# ps -ecl | grep idlist
```

Example—Changing the Class of a Process

The following example changes all the processes belonging to user `15249` to real-time processes.

```
# priocntl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

Note - If, as superuser, you change a user process to the real-time class, the user cannot subsequently change the real-time scheduling parameters (using `priocntl -s`).

Changing the Priority of a Timesharing Process With `nice`

The `nice`(1) command is only supported for backward compatibility to previous Solaris releases. The `priocntl` command provides more flexibility in managing processes.

The priority of a process is determined by the policies of its scheduling class, and by its *nice number*. Each timesharing process has a global priority which is calculated by adding the user-supplied priority, which can be influenced by the `nice` or `priocntl` commands, and the system-calculated priority.

The execution priority number of a process is assigned by the operating system, and is determined by several factors, including its schedule class, how much CPU time it has used, and (in the case of a timesharing process) its `nice` number.

Each timesharing process starts with a default `nice` number, which it inherits from its parent process. The `nice` number is shown in the `NI` column of the `ps` report.

A user can lower the priority of a process by increasing its user-supplied priority. But only the superuser can lower a `nice` number to increase the priority of a process. This is to prevent users from increasing the priorities of their own processes, thereby monopolizing a greater share of the CPU.

Nice numbers range between 0 and +40, with 0 representing the highest priority. The default value is 20. Two versions of the command are available, the standard version, `/usr/bin/nice`, and a version that is part of the C shell.

▼ How to Change the Priority of a Process

You can raise or lower the priority of a command or a process by changing the `nice` number. To lower the priority of a process:

`/usr/bin/nice` *command_name*	Increase the `nice` number by four units (the default)
`/usr/bin/nice +4` *command_name*	Increase the `nice` number by four units
`/usr/bin/nice -10` *command_name*	Increase the `nice` number by ten units

The first and second commands increase the `nice` number by four units (the default); and the third command increases the `nice` by ten units, lowering the priority of the process.

The following commands raise the priority of the command by lowering the `nice` number.

To raise the priority of a process:

`/usr/bin/nice -10` *command_name*	Raises the priority of the command by lowering the `nice` number
`/usr/bin/nice - -10` *command_name*	Raises the priority of the command by lowering the `nice` number. The first minus sign is the option sign, and the second minus sign indicates a negative number.

The above commands raise the priority of the command, *command_name*, by lowering the `nice` number. Note that in the second case, the two minus signs are required.

Process Troubleshooting

Here are some tips on obvious problems you may find:

- Look for several identical jobs owned by the same user. This may come as a result of running a script that starts a lot of background jobs without waiting for any of the jobs to finish.
- Look for a process that has accumulated a large amount of CPU time. You'll see this by looking at the `TIME` field. Possibly, the process is in an endless loop.
- Look for a process running with a priority that is too high. Type `ps -c` to see the `CLS` field, which displays the scheduler class of each process. A process executing as a real-time (`RT`) process can monopolize the CPU. Or look for a timeshare (`TS`) process with a high `nice` value. A user with superuser privileges may have bumped up the priorities of this process. The system administrator can lower the priority by using the `nice` command.
- Look for a runaway process—one that progressively uses more and more CPU time. You can see it happening by looking at the time when the process started (`STIME`) and by watching the cumulation of CPU time (`TIME`) for awhile.

CHAPTER 36

Monitoring Performance (Tasks)

This chapter describes procedures for monitoring system performance by using the `vmstat`, `iostat`, `df`, and `sar` commands. This is a list of the step-by-step instructions in this chapter.

- "How to Display Virtual Memory Statistics (`vmstat`)" on page 574
- "How to Display System Event Information (`vmstat -s`)" on page 576
- "How to Display Swapping Statistics (`vmstat -S`)" on page 577
- "How to Display Cache Flushing Statistics (`vmstat -c`)" on page 578
- "How to Display Interrupts Per Device (`vmstat -i`)" on page 578
- "How to Display Disk Utilization Information (`iostat`)" on page 579
- "How to Display Extended Disk Statistics (`iostat -xtc`)" on page 581
- "How to Display File System Information (`df`)" on page 582
- "How to Check File Access (`sar -a`)" on page 584
- "How to Check Buffer Activity (`sar -b`)" on page 584
- "How to Check System Call Statistics (`sar -c`)" on page 586
- "How to Check Disk Activity (`sar -d`)" on page 587
- "How to Check Page-Out and Memory (`sar -g`)" on page 588
- "How to Check Kernel Memory Allocation (`sar -k`)" on page 590
- "How to Check Interprocess Communication (`sar -m`)" on page 592
- "How to Check Page-In Activity (`sar -p`)" on page 593
- "How to Check Queue Activity (`sar -q`)" on page 594
- "How to Check Unused Memory (`sar -r`)" on page 595
- "How to Check CPU Utilization (`sar -u`)" on page 596
- "How to Check System Table Status (`sar -v`)" on page 598

- “How to Check Swap Activity (sar -w)” on page 599
- “How to Check Terminal Activity (sar -y)” on page 600
- “How to Check Overall System Performance (sar -A)” on page 602
- “How to Set Up Automatic Data Collection ” on page 604

Displaying Virtual Memory Statistics (vmstat)

You can use the vmstat command to report virtual memory statistics and such information about system events as CPU load, paging, number of context switches, device interrupts, and system calls. The vmstat command can also display statistics on swapping, cache flushing, and interrupts.

Refer to vmstat(1M) for a more detailed description of this command.

▼ How to Display Virtual Memory Statistics (vmstat)

Collect virtual memory statistics using the vmstat command with a time interval.

```
$ vmstat n
```

n Interval in seconds between reports.

The table below describes the fields in the vmstat output.

TABLE 36–1 Output From the vmstat Command

Category	Field Name	Description
procs		Reports the following states:
	r	The number of kernel threads in the dispatch queue
	b	Blocked kernel threads waiting for resources
	w	Swapped out LWPs waiting for processing resources to finish

TABLE 36–1 Output From the `vmstat` Command *(continued)*

Category	Field Name	Description
`memory`		Reports on usage of real and virtual memory:
	`swap`	Available swap space
	`free`	Size of the free list
`page`		Reports on page faults and paging activity, in units per second:
	`re`	Pages reclaimed
	`mf`	Minor and major faults
	`pi`	Kbytes paged in
	`po`	Kbytes paged out
	`fr`	Kbytes freed
	`de`	Anticipated memory needed by recently swapped-in processes
	`sr`	Pages scanned by page daemon (not currently in use). If `sr` does not equal zero, the page daemon has been running.
`disk`		Reports the number of disk operations per second, showing data on up to four disks
`faults`		Reports the trap/interrupt rates (per second):
	`in`	Interrupts per second
	`sy`	System calls per second
	`cs`	CPU context switch rate
`cpu`		Reports on the use of CPU time:
	`us`	User time

TABLE 36–1 Output From the `vmstat` Command *(continued)*

Category	Field Name	Description
	`sy`	System time
	`id`	Idle time

Example—Displaying Virtual Memory Statistics

The following example shows the `vmstat` display of statistics gathered at five-second intervals.

```
$ vmstat 5
 procs    memory            page            disk      faults      cpu
r b w   swap free re  mf  pi  po  fr de sr f0 s3 -- --  in   sy  cs us sy  id
0 0 8 28312  668  0   9   2   0   1  0  0  0  1  0  0  10  61  82  1  2  97
0 0 3 31940  248  0  10  20   0  26  0 27  0  4  0  0  53 189 191  6  6  88
0 0 3 32080  288  3  19  49   6  26  0 15  0  9  0  0  75 415 277  6 15  79
0 0 3 32080  256  0  26  20   6  21  0 12  1  6  0  0 163 110 138  1  3  96
0 1 3 32060  256  3  45  52  28  61  0 27  5 12  0  0 195 191 223  7 11  82
0 0 3 32056  260  0   1   0   0   0  0  0  0  0  0  0   4  52  84  0  1  99
```

▼ How to Display System Event Information (`vmstat -s`)

Run `vmstat -s` to show the total of various system events that have taken place since the system was last booted.

```
$ vmstat -s
        0 swap ins
        0 swap outs
        0 pages swapped in
        0 pages swapped out
   392182 total address trans. faults taken
    20419 page ins
      923 page outs
    30072 pages paged in
     9194 pages paged out
    65167 total reclaims
    65157 reclaims from free list
        0 micro (hat) faults
   392182 minor (as) faults
    19383 major faults
```

(continued)

```
    85775 copy-on-write faults
    66637 zero fill page faults
    46309 pages examined by the clock daemon
        6 revolutions of the clock hand
    15578 pages freed by the clock daemon
     4398 forks
      352 vforks
     4267 execs
 12926285 cpu context switches
109029866 device interrupts
   499296 traps
 22461261 system calls
   778068 total name lookups (cache hits 97%)
    18739 user   cpu
    34662 system cpu
 52051435 idle   cpu
    25252 wait   cpu
```

▼ How to Display Swapping Statistics (`vmstat -S`)

Run `vmstat -S` to show swapping statistics.

```
$ vmstat -S
 procs     memory            page            disk          faults      cpu
 r b w   swap  free  si  so pi po fr de sr f0 s0 s6 --   in   sy   cs us sy id
 0 0 0 200968 17936   0   0  0  0  0  0  0  0  0  0  0  109   43   24  0  0 100
```

The fields are described in the table below.

TABLE 36–2 Output From the `vmstat -S` Command

Field Name	Description
`si`	Average number of LWPs swapped in per second
`so`	Number of whole processes swapped out

Note - The `vmstat` command truncates the output of both of these fields. Use the `sar` command to display a more accurate accounting of swap statistics.

▼ How to Display Cache Flushing Statistics (`vmstat -c`)

Run `vmstat -c` to show cache flushing statistics for a virtual cache.

```
$ vmstat -c
usr     ctx     rgn     seg     pag     par
  0   60714       5  134584 4486560 4718054
```

It shows the total number of cache flushes since the last boot. The cache types are described in the table below.

TABLE 36–3 Output From the `vmstat -c` Command

Cache Name	Cache Type
`usr`	User
`ctx`	Context
`rgn`	Region
`seg`	Segment
`pag`	Page
`par`	Partial-page

▼ How to Display Interrupts Per Device (`vmstat -i`)

Run `vmstat -i` to show interrupts per device.

```
$ vmstat -i
```

Example—Displaying Interrupts Per Device

The following example shows output from the `vmstat -i` command.

```
$ vmstat -i
interrupt         total     rate
-------------------------------
clock          52163269      100
esp0            2600077        4
zsc0              25341        0
zsc1              48917        0
cgsixc0             459        0
lec0             400882        0
fdc0                 14        0
bppc0                 0        0
audiocs0              0        0
-------------------------------
Total          55238959      105
```

Displaying Disk Utilization Information (`iostat` *n*)

Use the `iostat` command to report statistics about disk input and output, and produces measures of throughput, utilization, queue lengths, transaction rates, and service time. For a detailed description of this command, refer to `iostat`(1M).

▼ How to Display Disk Utilization Information (`iostat`)

You can display disk activity information by using the `iostat` command with a time interval.

```
$ iostat 5
     tty          fd0           sd3          nfs1         nfs31          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0    1   0   0  410    3   0   29    0   0    9    3   0   47   4  2  0 94
```

The first line of output shows the statistics since the last boot. Each subsequent line shows the interval statistics. The default is to show statistics for the terminal (`tty`), disks (`fd` and `sd`), and CPU (`cpu`).

The table below describes the fields in the `iostat` command output.

TABLE 36–4 Output From the `iostat` *n* Command

For Each ...	Field Name	Description
Terminal		
	`tin`	Number of characters in the terminal input queue
	`tout`	Number of characters in the terminal output queue
Disk		
	`bps`	Blocks per second
	`tps`	Transactions per second
	`serv`	Average service time, in milliseconds
CPU		
	`us`	In user mode
	`sy`	In system mode
	`wt`	Waiting for I/O
	`id`	Idle

Example—Displaying Disk Utilization Information

The following example shows disk statistics gathered every five seconds.

```
$ iostat 5
tty        sd0           sd6           nfs1          nfs49           cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
 0    0   1   0   49    0   0    0    0   0    0    0   0   15   0  0  0 100
 0   47   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16  44   6  132    0   0    0    0   0    0    0   0    0   0  0  1 99
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
```

(continued)

```
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   3   1   23    0   0    0    0   0    0    0   0    0   0  0  1 99
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
```

▼ How to Display Extended Disk Statistics (`iostat -xtc`)

Run `iostat -xtc` to get extended disk statistics.

```
$ iostat -xtc
                 extended device statistics                      tty         cpu
device       r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b  tin tout  us sy wt id
fd0          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0    0    0   0  0  0 100
sd0          0.0    0.0    0.4    0.4  0.0  0.0   49.5   0   0
sd6          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs1         0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs49        0.0    0.0    0.0    0.0  0.0  0.0   15.1   0   0
nfs53        0.0    0.0    0.4    0.0  0.0  0.0   24.5   0   0
nfs54        0.0    0.0    0.0    0.0  0.0  0.0    6.3   0   0
nfs55        0.0    0.0    0.0    0.0  0.0  0.0    4.9   0   0
```

This command displays a line of output for each disk. The output fields are described in the table below.

TABLE 36–5 Output From the `iostat -xtc` Command

Field Name	Description
`r/s`	Reads per second
`w/s`	Writes per second
`Kr/s`	Kbytes read per second
`Kw/s`	Kbytes written per second
`wait`	Average number of transactions waiting for service (queue length)

TABLE 36–5 Output From the `iostat -xtc` Command *(continued)*

Field Name	Description
`actv`	Average number of transactions actively being serviced
`svc_t`	Average service time, in milliseconds
`%w`	Percentage of time the queue is not empty
`%b`	Percentage of time the disk is busy

Displaying Disk Usage Statistics (`df`)

Use the `df` command to show the amount of free disk space on each mounted disk. The *usable* disk space reported by `df` reflects only 90% of full capacity, as the reporting statistics leave a 10% head room above the total available space. This head room normally stays empty for better performance.

The percentage of disk space actually reported by `df` is used space divided by usable space.

If the file system is above 90% capacity, transfer files to a disk that is not as full by using `cp`, or to a tape by using `tar` or `cpio`; or remove the files.

For a detailed description of this command, refer to the `df`(1M) man page.

▼ How to Display File System Information (`df`)

Use the `df -k` command to display file system information in Kbytes.

```
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0     192807   40231  133296    24%    /
```

The table below describes the `df -k` command output.

TABLE 36–6 Output From the `df -k` Command

Field Name	Description
`kbytes`	Total size of usable space in the file system
`used`	Amount of space used
`avail`	Amount of space available for use
`capacity`	Amount of space used, as a percent of the total capacity
`mounted on`	Mount point

Example—Displaying File System Information

The following example shows output of the `df -k` command.

```
$ df -k
/dev/dsk/c0t0d0s0     192807   49043  124484    29%    /
/dev/dsk/c0t0d0s6    1190551  680444  450580    61%    /usr
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
mnttab                     0       0       0     0%    /etc/mnttab
swap                  198056       0  198056     0%    /var/run
swap                  198064       8  198056     1%    /tmp
/dev/dsk/c0t0d0s5     192807    2031  171496     2%    /opt
/dev/dsk/c0t0d0s7     217191       9  195463     1%    /export/home
venus:/usr/dist    20612581 13237316 6963015    66%    /usr/dist
```

Monitoring System Activities (`sar`)

Use the `sar` command to:

- Organize and view data about system activity
- Access system activity data on a special request basis
- Generate automatic reports to measure and monitor system performance, and special request reports to pinpoint specific performance problems. “Collecting System Activity Data Automatically (`sar`)” on page 602 describes these tools.

For a detailed description of this command, refer to `sar`(1).

▼ How to Check File Access (`sar -a`)

Display file access operation statistics with the `sar -a` command.

```
$ sar -a
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  iget/s namei/s dirbk/s
01:00:00       0       0       0
02:00:02       0       0       0
03:00:00       0       1       0
04:00:00       0       0       0
05:00:01       0       0       0
06:00:00       0       0       0

Average        0       1       0
```

The operating system routines reported are described in the following table.

TABLE 36–7 Output from the `sar -a` Command

Field Name	Description
`iget/s`	The number of requests made for inodes that were not in the directory name lookup cache (`dnlc`).
`namei/s`	This is the number of file system path searches per second. If `namei` does not find a directory name in the `dnlc`, it calls `iget` to get the inode for either a file or directory. Hence, most `igets` are the result of `dnlc` misses.
`dirbk/s`	This is the number of directory block reads issued per second.

The larger the values reported, the more time the kernel is spending to access user files. The amount of time reflects how heavily programs and applications are using the file systems. The `-a` option is helpful for viewing how disk-dependent an application is.

▼ How to Check Buffer Activity (`sar -b`)

Display buffer activity statistics with the `sar -b` command.

The buffer is used to cache metadata, which includes inodes, cylinder group blocks, and indirect blocks.

```
$ sar -b
00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      55       0       0
```

The buffer activities displayed by the –b option are described in the table below. The most important entries are the cache hit ratios %rcache and %wcache, which measure the effectiveness of system buffering. If %rcache falls below 90, or if %wcache falls below 65, it may be possible to improve performance by increasing the buffer space.

TABLE 36–8 Output from the sar -b Command

Field Name	Description
bread/s	Average number of reads per second submitted to the buffer cache from the disk
lread/s	Average number of logical reads per second from the buffer cache
%rcache	Fraction of logical reads found in the buffer cache (100% minus the ratio of bread/s to lread/s)
bwrit/s	Average number of physical blocks (512 blocks) written from the buffer cache to disk, per second
lwrite/s	Average number of logical writes to the buffer cache, per second
%wcache	Fraction of logical writes found in the buffer cache(100% minus the ratio of bwrit/s to lwrit/s)
pread/s	Average number of physical reads, per second, using character device interfaces
pwrit/s	Average number of physical write requests, per second, using character device interfaces

Example—Checking Buffer Activity

The following abbreviated example of sar -b output shows that the %rcache and %wcache buffers are not causing any slowdowns, because all the data is within acceptable limits.

```
$ sar -b
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      55       0       0
02:00:02       0       0     100       0       0      55       0       0
03:00:00       0       0     100       0       0      72       0       0
04:00:00       0       0     100       0       0      56       0       0
05:00:01       0       0     100       0       0      55       0       0
06:00:00       0       0     100       0       0      55       0       0

Average        0       0      94       0       0      64       0       0
```

▼ How to Check System Call Statistics (`sar -c`)

Display system call statistics by using the `sar -c` command.

```
$ sar -c
00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38       2       2    0.00    0.00     149     120
```

The table below describes the following system call categories reported by the `-c` option. Typically, `reads` and `writes` account for about half of the total system calls, although the percentage varies greatly with the activities that are being performed by the system.

TABLE 36–9 Output from the `sar -c` Command

Field Name	Description
`scall/s`	All types of system calls per second (generally about 30 per second on a busy four- to six-user system)
`sread/s`	`read` system calls per second
`swrit/s`	`write` system calls per second
`fork/s`	`fork` system calls per second (about 0.5 per second on a four- to six-user system); this number will increase if shell scripts are running
`exec/d`	`exec` system calls per second; if `exec/s` divided by `fork/s` is greater than three, look for inefficient `PATH` variables

TABLE 36–9 Output from the `sar -c` Command *(continued)*

Field Name	Description
`rchar/s`	Characters (bytes) transferred by `read` system calls per second
`wchar/s`	Characters (bytes) transferred by `write` system calls per second

Example—Checking System Call Statistics

The following abbreviated example shows output from the `sar -c` command.

```
$ sar -c
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38       2       2    0.00    0.00     149     120
02:00:02      38       2       2    0.00    0.00     149     120
03:00:00      42       2       2    0.05    0.05     218     147
04:00:00      39       2       2    0.01    0.00     155     123
05:00:01      38       2       2    0.00    0.00     150     120
06:00:00      38       2       2    0.01    0.00     149     120

Average       50       4       3    0.02    0.02     532     238
```

▼ How to Check Disk Activity (`sar -d`)

Display disk activity statistics with the `sar -d` command.

```
$ sar -d
00:00:00   device        %busy   avque   r+w/s  blks/s  avwait  avserv

01:00:00   fd0               0     0.0       0       0     0.0     0.0
```

The table below describes the disk devices activities reported by the –`d` option. Note that queue lengths and wait times are measured when there is something in the queue. If `%busy` is small, large queues and service times probably represent the periodic efforts by the system to ensure that altered blocks are written to the disk in a timely fashion.

TABLE 36–10 Output from the `sar -d` Command

Field Name	Description
`device`	Name of the disk device being monitored
`%busy`	Percentage of time the device spent servicing a transfer request
`avque`	The sum of the average wait time plus the average service time
`r+w/s`	Number of read and write transfers to the device per second
`blks/s`	Number of 512-byte blocks transferred to the device per second
`avwait`	Average time, in milliseconds, that transfer requests wait idly in the queue (measured only when the queue is occupied)
`avserv`	Average time, in milliseconds, for a transfer request to be completed by the device (for disks, this includes seek, rotational latency, and data transfer times)

Examples—Checking Disk Activity

This abbreviated example illustrates the `sar -d` output.

```
$ sar -d
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00   device        %busy   avque   r+w/s  blks/s  avwait  avserv

01:00:00   fd0               0     0.0       0       0     0.0     0.0
           nfs1              0     0.0       0       0     0.0     0.0
           sd0               0     0.0       0       0     0.0    39.6
           sd0,a             0     0.0       0       0     0.0    39.6
           sd0,b             0     0.0       0       0     0.0     0.0
           sd0,c             0     0.0       0       0     0.0     0.0
           sd0,f             0     0.0       0       0     0.0     0.0
           sd0,g             0     0.0       0       0     0.0     0.0
           sd0,h             0     0.0       0       0     0.0     0.0
           sd6               0     0.0       0       0     0.0     0.0
```

▼ How to Check Page-Out and Memory (`sar -g`)

Use the `sar -g` option reports page-out and memory freeing activities (in averages).

```
$ sar -g
00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
```

The output displayed by `sar -g` is a good indicator of whether more memory may be needed. Use the `ps -elf` command to show the number of cycles used by the page daemon. A high number of cycles, combined with high values for `pgfree/s` and `pgscan/s` indicates a memory shortage.

The `sar -g` command also shows whether inodes are being recycled too quickly, causing a loss of reusable pages.

Output from the –`g` option is described in the following table.

TABLE 36–11 Output From the `sar -g` Command

Field Name	Description
`pgout/s`	The number of page-out requests per second.
`ppgout/s`	The actual number of pages that are paged-out, per second. (A single page-out request may involve paging-out multiple pages.)
`pgfree/s`	The number of pages, per second, that are placed on the free list.
`pgscan/s`	The number of pages, per second, scanned by the page daemon. If this value is high, the page daemon is spending a lot of time checking for free memory. This implies that more memory may be needed.
`%ufs_ipf`	The percentage of `ufs` inodes taken off the free list by `iget` that had reusable pages associated with them. These pages are flushed and cannot be reclaimed by processes. Thus, this is the percentage of `igets` with page flushes. A high value indicates that the free list of inodes is page-bound and the number of `ufs` inodes may need to be increased.

Example—Checking Page-Out and Memory

The following abbreviated example shows output from the `sar -g` command.

```
$ sar -g
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
02:00:02     0.00     0.00     0.00     0.00     0.00
03:00:00     0.00     0.01     0.01     0.00     0.00
04:00:00     0.00     0.00     0.00     0.00     0.00
05:00:01     0.00     0.00     0.00     0.00     0.00
06:00:00     0.00     0.00     0.00     0.00     0.00

Average      0.01     0.12     0.21     0.66     0.00
```

▼ How to Check Kernel Memory Allocation (`sar -k`)

Use the `sar -k` command to report on the following activities of the Kernel Memory Allocator (KMA).

The KMA allows a kernel subsystem to allocate and free memory as needed. Rather than statically allocating the maximum amount of memory it is expected to require under peak load, the KMA divides requests for memory into three categories: small (less than 256 bytes), large (512 to 4 Kbytes), and oversized (greater than 4 Kbytes). It keeps two pools of memory to satisfy small and large requests. The oversized requests are satisfied by allocating memory from the system page allocator.

If you are investigating a system that is being used to write drivers or STREAMS that use KMA resources, then `sar -k` will likely prove useful. Otherwise, you will probably not need the information it provides. Any driver or module that uses KMA resources, but does not specifically return the resources before it exits, can create a memory leak. A memory leak causes the amount of memory allocated by KMA to increase over time. Thus, if the `alloc` fields of `sar -k` increase steadily over time, there may be a memory leak. Another indication of a memory leak is failed requests. If this occurs, a memory leak has probably caused KMA to be unable to reserve and allocate memory.

If it appears that a memory leak has occurred, you should check any drivers or STREAMS that may have requested memory from KMA and not returned it.

```
$ sar -k
00:00:00 sml_mem   alloc  fail  lg_mem   alloc   fail  ovsz_alloc  fail
01:00:00 2523136 1866512     0 18939904 14762364    0      360448     0
02:00:02 2523136 1861724     0 18939904 14778748    0      360448     0
```

Output from the –k option is described in the table below.

TABLE 36–12 Output From the `sar -k` Command

Field Name	Description
`sml_mem`	The amount of memory, in bytes, that the KMA has available in the small memory request pool (a small request is less than 256 bytes)
`alloc`	The amount of memory, in bytes, that the KMA has allocated from its small memory request pool to small memory requests
`fail`	The number of requests for small amounts of memory that failed
`lg_mem`	The amount of memory, in bytes, that the KMA has available in the large memory request pool (a large request is from 512 bytes to 4 Kbytes)
`alloc`	The amount of memory, in bytes, that the KMA has allocated from its large memory request pool to large memory requests
`fail`	The number of failed requests for large amounts of memory
`ovsz_alloc`	The amount of memory allocated for oversized requests (those greater than 4 Kbytes); these requests are satisfied by the page allocator—thus, there is no pool
`fail`	The number of failed requests for oversized amounts of memory

Example—Checking Kernel Memory Allocation (`sar`)

The following is an abbreviated example of `sar -k` output.

```
$ sar -k
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
01:00:00 2523136 1866512     0 18939904 14762364     0      360448     0
02:00:02 2523136 1861724     0 18939904 14778748     0      360448     0
03:00:00 2523136 1865664     0 18939904 14745884     0      360448     0
04:00:00 2523136 1867692     0 18939904 14746616     0      360448     0
05:00:01 2523136 1867208     0 18939904 14763700     0      360448     0
06:00:00 2523136 1867772     0 18939904 14779444     0      360448     0

Average  2724096 1791806     0 20089344 15434591     0      360448     0
```

▼ How to Check Interprocess Communication (`sar -m`)

Use the `sar -m` command to report interprocess communication activities.

```
$ sar -m
00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
```

These figures will usually be zero (0.00), unless you are running applications that use messages or semaphores.

The output from the –`m` option is described in the table below.

TABLE 36–13 Output From the `sar -m` Command

Field Name	Description
`msg/s`	The number of message operations (sends and receives) per second.
`sema/s`	The number of semaphore operations per second.

Example—Checking Interprocess Communication

The following abbreviated example shows output from the `sar -m` command.

```
$ sar -m
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
02:00:02    0.00    0.00
03:00:00    0.00    0.00
04:00:00    0.00    0.00
05:00:01    0.00    0.00
06:00:00    0.00    0.00

Average     0.00    0.00
```

▼ How to Check Page-In Activity (sar -p)

Use the `sar -p` command to report page-in activity which includes protection and translation faults.

```
$ sar -p
00:00:00  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.07    0.00    0.00    0.21    0.39    0.00
```

The reported statistics from the –p option are described in the table below.

TABLE 36–14 Output from the `sar -p` Command

Field Name	Description
`atch/s`	The number of page faults, per second, that are satisfied by reclaiming a page currently in memory (attaches per second). Instances of this include reclaiming an invalid page from the free list and sharing a page of text currently being used by another process (for example, two or more processes accessing the same program text).
`pgin/s`	The number of times, per second, that file systems receive page-in requests.
`ppgin/s`	The number of pages paged in, per second. A single page-in request, such as a soft-lock request (see `slock/s`), or a large block size, may involve paging-in multiple pages.
`pflt/s`	The number of page faults from protection errors. Instances of protection faults are illegal access to a page and "copy-on-writes." Generally, this number consists primarily of "copy-on-writes."
`vflt/s`	The number of address translation page faults, per second. These are known as validity faults, and occur when a valid process table entry does not exist for a given virtual address.
`slock/s`	The number of faults, per second, caused by software lock requests requiring physical I/O. An example of the occurrence of a soft-lock request is the transfer of data from a disk to memory. The system locks the page that is to receive the data, so that it cannot be claimed and used by another process.

Example—Checking Page-In Activity

The following abbreviated example shows output from `sar -p`.

```
$ sar -p
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.07    0.00    0.00    0.21    0.39    0.00
02:00:02    0.07    0.00    0.00    0.21    0.39    0.00
03:00:00    0.32    0.00    0.00    1.10    2.48    0.00
04:00:00    0.09    0.00    0.00    0.32    0.57    0.00
05:00:01    0.07    0.00    0.00    0.21    0.39    0.00
06:00:00    0.07    0.00    0.00    0.21    0.39    0.00

Average     0.26    0.20    0.30    0.92    1.78    0.00
```

▼ How to Check Queue Activity (`sar -q`)

Use the `sar -q` command to report the average queue length while the queue is occupied, and the percentage of time that the queue is occupied.

```
$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00
```

Note - The number of LWPs swapped out may greater than zero even if the system has an abundance of free memory. This happens when a sleeping LWP is swapped out and has not been awakened (for example, a process or LWP sleeping, waiting for the keyboard or mouse input).

Output from the –`q` option is described in the table below.

TABLE 36–15 Output From the `sar -q` Command

Field Name	Description
`runq-sz`	The number of kernel threads in memory waiting for a CPU to run. Typically, this value should be less than 2. Consistently higher values mean that the system may be CPU-bound.
`%runocc`	The percentage of time the dispatch queues are occupied.

TABLE 36–15 Output From the `sar -q` Command *(continued)*

Field Name	Description
`swpq-sz`	The average number of swapped out LWPs.
`%swpocc`	The percentage of time LWPs are swapped out.

Example—Checking Queue Activity

The following abbreviated example shows output from the `sar -q` command. If `%runocc` is high (greater than 90 percent) and `runq-sz` is greater than 2, the CPU is heavily loaded and response is degraded. In this case, additional CPU capacity may be required to obtain acceptable system response.

```
$ sar -q
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00
02:00:02
03:00:00     1.0       0
04:00:00
05:00:01     1.0       0
06:00:00

Average      1.3       0
```

▼ How to Check Unused Memory (`sar -r`)

Use the `sar -r` command to report the number of memory pages and swap-file disk blocks that are currently unused.

```
$ sar -r
00:00:00 freemem freeswap
01:00:00    2135   401922
```

Output from the –`r` option is described in the table below.

TABLE 36–16 Output From the `sar -r` Command

Field Name	Description
`freemem`	The average number of memory pages available to user processes over the intervals sampled by the command. Page size is machine-dependent.
`freeswap`	The number of 512-byte disk blocks available for page swapping.

Example—Checking Unused Memory

The following example shows output from the `sar -r` command.

```
$ sar -r
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 freemem freeswap
01:00:00    2135   401922
02:00:02    2137   401949
03:00:00    2137   402006
04:00:00    2139   401923
05:00:01    2138   402033
06:00:00    2137   401919

Average     2500   399914
```

▼ How to Check CPU Utilization (`sar -u`)

Display CPU utilization with the `sar -u` command.

```
$ sar -u
00:00:00    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
```

(The `sar` command without any options is equivalent to `sar -u`.) At any given moment, the processor is either busy or idle. When busy, the processor is in either user or system mode. When idle, the processor is either waiting for I/O completion or “sitting still” with no work to do.

Output from the –`u` option is described in the table below.

TABLE 36–17 Output From the `sar -u` Command

Field Name	Description
`%sys`	Lists the percentage of time that the processor is in system mode
`%user`	Lists the percentage of time that the processor is in user mode
`%wio`	Lists the percentage of time the processor is idle and waiting for I/O completion
`%idle`	Lists the percentage of time the processor is idle and is not waiting for I/O

A high `%wio` generally means a disk slowdown has occurred.

Example—Checking CPU Utilization

The following example shows output from the `sar -u` command.

```
$ sar -u
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
02:00:02       0       0       0     100
03:00:00       0       0       0     100
04:00:00       0       0       0     100
05:00:01       0       0       0     100
06:00:00       0       0       0     100
07:00:00       0       0       0     100
08:00:01       0       0       0     100
08:20:00       0       0       0     100
08:40:00       0       0       0     100
09:00:00       0       0       0     100
09:20:00       0       0       0     100
09:40:00       0       0       0     100
10:00:00       0       0       0     100
10:20:00       0       0       0     100
10:40:01       0       0       0     100
11:00:00       5       2      10      82

Average        0       0       0     100
```

▼ How to Check System Table Status (`sar -v`)

Use the `sar -v` command to report the status of the process table, inode table, file table, and shared memory record table.

```
$ sar -v
00:00:00  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   43/922     0 2984/4236    0  322/322     0    0/0
```

Output from the –`v` option is described in the table below.

TABLE 36–18 Output From the `sar -v` Command

Field Name	Description
`proc-sz`	The number of process entries (`proc` structs) currently being used, or allocated in the kernel.
`inod-sz`	The total number of inodes in memory verses the maximum number of inodes allocated in the kernel. This is not a strict high water mark; it can overflow.
`file-sz`	The size of the open system file table. The `sz` is given as `0`, since space is allocated dynamically for the file table.
`ov`	The number of shared memory record table entries currently being used or allocated in the kernel. The `sz` is given as `0` because space is allocated dynamically for the shared memory record table.
`lock-sz`	The number of shared memory record table entries currently being used or allocated in the kernel. The `sz` is given as `0` because space is allocated dynamically for the shared memory record table.

Example—Checking System Table Status

The following abbreviated example shows output from the `sar -v` command. This example shows that all tables are large enough to have no overflows. These tables are all dynamically allocated based on the amount of physical memory.

```
$ sar -v
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   43/922     0 2984/4236    0  322/322     0    0/0
02:00:02   43/922     0 2984/4236    0  322/322     0    0/0
03:00:00   43/922     0 2986/4236    0  323/323     0    0/0
04:00:00   43/922     0 2987/4236    0  322/322     0    0/0
05:00:01   43/922     0 2987/4236    0  322/322     0    0/0
06:00:00   43/922     0 2987/4236    0  322/322     0    0/0
```

▼ How to Check Swap Activity (`sar -w`)

Use the `sar -w` command to report swapping and switching activity.

```
$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0      22
```

Target values and observations are described in the table below.

TABLE 36–19 Output From the `sar -w` Command

Field Name	Description
`swpin/s`	The number of LWP transfers into memory per second.
`bswin/s`	The average number of processes swapped out of memory per second. If the number is greater than 1, you may need to increase memory.
`swpot/s`	The average number of processes swapped out of memory per second. If the number is greater than 1, you may need to increase memory.
`bswot/s`	The number of blocks transferred for swap-outs per second.
`pswch/s`	The number of kernel thread switches per second.

Note - All process swap-ins include process initialization.

Example—Checking Swap Activity

The following example shows output from the `sar -w` command.

```
$ sar -w
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0      22
02:00:02    0.00     0.0    0.00     0.0      22
03:00:00    0.00     0.0    0.00     0.0      22
04:00:00    0.00     0.0    0.00     0.0      22
05:00:01    0.00     0.0    0.00     0.0      22
06:00:00    0.00     0.0    0.00     0.0      22
07:00:00    0.00     0.0    0.00     0.0      22
08:00:01    0.00     0.0    0.00     0.0      22
08:20:00    0.00     0.0    0.00     0.0      22
08:40:00    0.00     0.0    0.00     0.0      22
09:00:00    0.00     0.0    0.00     0.0      22
09:20:00    0.00     0.0    0.00     0.0      22
09:40:00    0.00     0.0    0.00     0.0      22
10:00:00    0.00     0.0    0.00     0.0      22
10:20:00    0.00     0.0    0.00     0.0      22
10:40:01    0.00     0.0    0.00     0.0      23
11:00:00    0.00     0.0    0.00     0.0     144

Average     0.00     0.0    0.00     0.0      24
```

▼ How to Check Terminal Activity (`sar -y`)

Use the `sar -y` command to monitor terminal device activities.

```
$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
```

If you have a lot of terminal I/O, you can use this report to determine if there are any bad lines. The activities recorded are defined in the table below.

TABLE 36–20 Output From the `sar -y` Command

Field Name	Description
`rawch/s`	Input characters (raw queue), per second
`canch/s`	Input characters processed by canon (canonical queue) per second

TABLE 36–20 Output From the `sar -y` Command *(continued)*

Field Name	Description
`outch/s`	Output characters (output queue) per second
`rcvin/s`	Receiver hardware interrupts per second
`xmtin/s`	Transmitter hardware interrupts per second
`mdmin/s`	Modem interrupts per second

The number of modem interrupts per second (`mdmin/s`) should be close to zero, and the receive and transmit interrupts per second (`xmtin/s` and `rcvin/s`) should be less than or equal to the number of incoming or outgoing characters, respectively. If this is not the case, check for bad lines.

Example—Checking Terminal Activity

The following abbreviated example shows output from the `sar -y` command.

```
$ sar -y
SunOS venus 5.8 Generic sun4u    09/07/99

00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
02:00:02       0       0       0       0       0       0
03:00:00       0       0       0       0       0       0
04:00:00       0       0       0       0       0       0
05:00:01       0       0       0       0       0       0
06:00:00       0       0       0       0       0       0
07:00:00       0       0       0       0       0       0
08:00:01       0       0       0       0       0       0
08:20:00       0       0       0       0       0       0
08:40:00       0       0       0       0       0       0
09:00:00       0       0       0       0       0       0
09:20:00       0       0       0       0       0       0
09:40:00       0       0       0       0       0       0
10:00:00       0       0       0       0       0       0
10:20:00       0       0       0       0       0       0
10:40:01       0       0      20       0       0       0

Average        0       0       3       0       0       0
```

▼ How to Check Overall System Performance (sar -A)

Use the `sar -A` command to display a view of overall system performance.

This provides a more global perspective. If data from more than one time segment is shown, the report includes averages.

Collecting System Activity Data Automatically (sar)

Three commands are involved in automatic system activity data collection: `sadc`, `sa1`, and `sa2`.

The `sadc` data collection utility periodically collects data on system activity and saves it in a file in binary format—one file for each 24-hour period. You can set up `sadc` to run periodically (usually once each hour), and whenever the system boots to multiuser mode. The data files are placed in the directory `/usr/adm/sa`. Each file is named `sa`*dd*, where *dd* is the current date. The format of the command is as follows:

```
/usr/lib/sa/sadc [t n] [ofile]
```

The command samples *n* times with an interval of *t* seconds (*t* should be greater than 5 seconds) between samples. It then writes, in binary format, to the file *ofile*, or to standard output. If *t* and *n* are omitted, a special file is written once.

Running `sadc` When Booting

The `sadc` command should be run at system boot time in order to record the statistics from when the counters are reset to zero. To make sure that `sadc` is run at boot time, the `/etc/init.d/perf` file must contain a command line that writes a record to the daily data file.

The command entry has the following format:

```
su sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa‘date +5d‘"
```

Running `sadc` Periodically With `sa1`

To generate periodic records, you need to run `sadc` regularly. The simplest way to do this is by putting a line into the `/var/spool/cron/sys` file, which calls the shell script, `sa1`. This script invokes `sadc` and writes to the daily data files, `/var/adm/sa/sa`*dd*. It has the following format:

```
/usr/lib/sa/sa1 [t n]
```

The arguments *t* and *n* cause records to be written *n* times at an interval of *t* seconds. If these arguments are omitted, the records are written only one time.

Producing Reports With sa2

Another shell script, sa2, produces reports rather than binary data files. The sa2 command invokes the sar command and writes the ASCII output to a report file.

Collecting System Activity Data (sar)

The sar command can be used either to gather system activity data itself or to report what has been collected in the daily activity files created by sadc.

The sar command has the following formats:

```
sar [-aAbcdgkmpqruvwy] [-o file] t [n]
```

```
sar [-aAbcdgkmpqruvwy] [-s time] [-e time] [-i sec] [-f file]
```

The sar command below samples cumulative activity counters in the operating system every *t* seconds, *n* times. (*t* should be 5 seconds or greater; otherwise, the command itself may affect the sample.) You must specify a time interval between which to take the samples; otherwise, the command operates according to the second format. The default value of *n* is 1. The following example takes two samples separated by 10 seconds. If the –o option is specified, samples are saved in *file* in binary format.

```
$ sar -u 10 2
```

Other important information about the sar command:

- With no sampling interval or number of samples specified, sar extracts data from a previously recorded file, either the one specified by the –f option or, by default, the standard daily activity file, /var/adm/sa/sa*dd*, for the most recent day.
- The –s and –e options define the starting and ending times for the report. Starting and ending times are of the form *hh*[*:mm*[*:ss*]] (where *h*, *m*, and *s* represent hours, minutes, and seconds).
- The –i option specifies, in seconds, the intervals between record selection. If the –i option is not included, all intervals found in the daily activity file are reported.

The table below lists the sar options and their actions.

TABLE 36–21 Options for `sar` Command

Option	Actions
`–a`	Checks file access operations
`–b`	Checks buffer activity
`–c`	Checks system calls
`–d`	Checks activity for each block device
`–g`	Checks page-out and memory freeing
`–k`	Checks kernel memory allocation
`–m`	Checks interprocess communication
`–p`	Checks swap and dispatch activity
`–q`	Checks queue activity
`–r`	Checks unused memory
`–u`	Checks CPU utilization
`–nv`	Checks system table status
`–w`	Checks swapping and switching volume
`–y`	Checks terminal activity
`–A`	Reports overall system performance (same as entering all options)

If no option is used, it is equivalent to calling the command with the `–u` option.

▼ How to Set Up Automatic Data Collection

1. Become superuser.

2. **Edit the** `/etc/init.d/perf` **file and uncomment all lines:**

   This version of the `sadc` command writes a special record that marks the time when the counters are reset to zero (boot time). The `sadc` output is put into the file `sa`*dd* (where *dd* is the current date), which acts as the daily system activity record.

3. **Edit the** `/var/spool/cron/crontabs/sys` **file (the system** `crontab` **file) and uncomment the following lines:**

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

   The first entry writes a record to `/var/adm/sa/sa`*dd* on the hour, every hour, seven days a week.

   The second entry writes a record to `/var/adm/sa/sa`*dd* twice each hour during peak working hours: at 20 minutes and 40 minutes past the hour, from 8 a.m. to 5 p.m., Monday through Friday.

   Thus, these two `crontab` entries cause a record to be written to `/var/adm/sa/sa`*dd* every 20 minutes from 8 a.m. to 5 p.m., Monday through Friday, and every hour on the hour otherwise. You can change these defaults to meet your needs.

CHAPTER 37

Troubleshooting Solaris Software Topics

This section provides instructions for troubleshooting Solaris software problems. This section contains these chapters.

Chapter 38	Provides overview information about troubleshooting common software problems and instructions for troubleshooting a system crash.
Chapter 39	Provides step-by-step instructions for saving crash dumps and customizing system error logging.
Chapter 40	Provides problem scenarios and possible solutions for general software problems such as a hung system or a system that won't boot.
Chapter 41	Provides solutions for solving common file access problems such as incorrect command search paths and file permissions.
Chapter 42	Provides solutions for solving common printer problems such as no output or incorrect output.
Chapter 43	Provides specific `fsck` error messages and corresponding solutions for solving file system-related problems.
Chapter 44	Provides specific error messages and possible solutions for problems encountered when adding or removing software packages.

CHAPTER 38

Troubleshooting Software Problems (Overview)

This chapter provides a general overview of troubleshooting software problems, including information on troubleshooting system crashes and viewing system messages.

This is a list of information in this chapter.

- "Where to Find Software Troubleshooting Tasks" on page 609
- "Troubleshooting a System Crash" on page 611
- "Troubleshooting a System Crash Checklist" on page 613
- "Viewing System Messages" on page 614
- "Customizing System Message Logging" on page 616

Where to Find Software Troubleshooting Tasks

Use these references to find step-by-step instructions for troubleshooting software problems.

- Chapter 39
- Chapter 40
- Chapter 41
- Chapter 42
- Chapter 43

What's New in System Troubleshooting?

This section describes new system troubleshooting features in the Solaris 8 release.

apptrace

A new application debugging tool, `apptrace`, enables application developers and system support personnel to debug application or system problems by providing call traces to Solaris shared libraries, which may show the series of events leading up to a point of failure.

The `apptrace` tool provides more reliable call-tracing than the previously available `sotruss` command. It also provides better display of function arguments, return values, and error cases for any Solaris library interface.

By default, `apptrace` traces calls directly from the executable object, specified on the command line, to every shared library the executable depends on.

See `apptrace`(1) for more information.

Improved Core File Management

The `coreadm` Command

This release introduces the `coreadm` command, which provides flexible core file naming conventions and better core file retention. For example, you can use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining the core files in a specific directory whenever a Solaris process or daemon terminates abnormally.

Two new configurable `core` file paths, per-process and global, can be enabled or disabled independently of each other. When a process terminates abnormally, it produces a `core` file in the current directory as in previous Solaris releases. But if a global core file path is enabled and set to `/corefiles/core`, for example, then each process that terminates abnormally would produce *two* core files: one in the current working directory and one in the `/corefiles` directory.

By default, the Solaris core paths and core file retention remain the same.

See “Managing Core Files (coreadm)” on page 626 and coreadm(1M) for more information.

Examining Core Files With Proc Tools

Some of the proc tools have been enhanced to examine process core files as well as live processes. The proc tools are utilities that can manipulate features of the /proc file system.

The /usr/proc/bin/pstack, pmap, pldd, pflags, and pcred tools can now be applied to core files by specifying the name of the core file on the command line, similar to the way you specify a process ID to these commands. For example:

```
$ ./a.out
Segmentation Fault(coredump)
$ /usr/proc/bin/pstack ./core
core './core' of 19305: ./a.out
 000108c4 main     (1, ffbef5cc, ffbef5d4, 20800, 0, 0) + 1c
 00010880 _start   (0, 0, 0, 0, 0, 0) + b8
```

For more information on using proc tools to examine core files, see proc(1).

New Remote Console Messaging Features

New remote console features improve your ability to troubleshoot remote systems.

See “Enabling Remote Console Messaging” on page 619 and consadm(1M) for more information.

Troubleshooting a System Crash

If a system running the Solaris operating environment crashes, provide your service provider with as much information as possible—including crash dump files.

What to Do if the System Crashes

The most important things to remember are:

1. Write down the system console messages.

If a system crashes, making it run again might seem like your most pressing concern. However, before you reboot the system, examine the console screen for messages. These messages can provide some insight about what caused the crash. Even if the system reboots automatically and the console messages have disappeared from the screen, you might be able to check these messages by viewing the system error log file that is generated automatically in `/var/adm/messages` (or `/usr/adm/messages`). See "How to View System Messages" on page 615 for more information about viewing system error log files.

If you have frequent crashes and can't determine their cause, gather all the information you can from the system console or the `/var/adm/messages` files, and have it ready for a customer service representative to examine. See "Troubleshooting a System Crash" on page 611 for a complete list of troubleshooting information to gather for your service provider.

See Chapter 40 if the system fails to reboot successfully after a system crash.

2. Synchronize the disks and reboot.

```
ok sync
```

See Chapter 40 if the system fails to reboot successfully after a system crash.

3. Attempt to save the crash information written onto the swap area by running the `savecore` command.

```
# savecore
```

See Chapter 39 for information about saving crash dumps automatically.

Gathering Troubleshooting Data

Answer the following questions to help isolate the system problem. Use "Troubleshooting a System Crash Checklist" on page 613 for gathering troubleshooting data for a crashed system.

TABLE 38–1 Identifying System Crash Data

Question	Description
Can you reproduce the problem?	This is important because a reproducible test case is often essential for debugging really hard problems. By reproducing the problem, the service provider can build kernels with special instrumentation to trigger, diagnose, and fix the bug.
Are you using any third-party drivers?	Drivers run in the same address space as the kernel, with all the same privileges, so they can cause system crashes if they have bugs.
What was the system doing just before it crashed?	If the system was doing anything unusual like running a new stress test or experiencing higher-than-usual load, that may have led to the crash.
Were there any unusual console messages right before the crash?	Sometimes the system will show signs of distress before it actually crashes; this information is often useful.
Did you add any tuning parameters to the `/etc/system` *file*?	Sometimes tuning parameters, such as increasing shared memory segments so that the system tries to allocate more than it has, can cause the system to crash.
Did the problem start recently?	If so, did the onset of problems coincide with any changes to the system, for example, new drivers, new software, different workload, CPU upgrade, or a memory upgrade.

Troubleshooting a System Crash Checklist

Use this checklist when gathering system data for a crashed system.

Item	Your Data
Is a core file available?	
Identify the operating system release and appropriate software application release levels.	
Identify system hardware.	
Include `prtdiag` output from sun4d systems.	
Are patches installed? If so, include `showrev -p` output.	
Is the problem reproducible?	
Does the system have any third-party drivers?	
What was the system doing before it crashed?	
Were there any unusual console messages right before the system crashed?	
Did you add any parameters to the `/etc/system` file?	
Did the problem start recently?	

Viewing System Messages

System messages display on the console device. The text of most system messages look like this:

[ID *msgid facility.priority*]

For example:

```
[ID 672855 kern.notice] syncing file systems...
```

If the message originated in the kernel, the kernel module name is displayed. For example:

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /: file system full
```

When a system crashes, it may display a message on the system console like this:

```
panic: error message
```

where *error message* is one of the panic error messages described in `crash`(1M).

Less frequently, this message may be displayed instead of the panic message:

```
Watchdog reset !
```

The error logging daemon, `syslogd`, automatically records various system warnings and errors in message files. By default, many of these system messages are displayed on the system console and are stored in the `/var/adm` directory. You can direct where these messages are stored by setting up system logging. See "How to Customize System Message Logging" on page 618 for more information. These messages can alert you to system problems, such as a device that is about to fail.

The `/var/adm` directory contains several message files. The most recent messages are in `/var/adm/messages` (and in `messages.0`), and the oldest are in `messages.3`. After a period of time (usually every ten days), a new `messages` file is created. The `messages.0` file is renamed `messages.1`, `messages.1` is renamed `messages.2`, and `messages.2` is renamed `messages.3`. The current `/var/adm/messages.3` is deleted.

Because `/var/adm` stores large files containing messages, crash dumps, and other data, this directory can consume lots of disk space. To keep the `/var/adm` directory from growing too large, and to ensure that future crash dumps can be saved, you should remove unneeded files periodically. You can automate this task by using `crontab`. See "How to Delete Crash Dump Files" on page 479 and Chapter 30 for more information on automating this task.

▼ How to View System Messages

Display recent messages generated by a system crash or reboot by using the `dmesg` command.

```
$ dmesg
```

Or use the `more` command to display one screen of messages at a time.

```
$ more /var/adm/messages
```

For more information, refer to `dmesg`(1M).

Example—Viewing System Messages

The following example shows output from the `dmesg` command.

```
$ dmesg
date starbug genunix: [ID 540533 kern.notice] SunOS Release
5.8 Version 64-bit
date starbug genunix: [ID 223299 kern.notice] Copyright
(c) 1983-1999 by Sun Microsystems, Inc.
date starbug genunix: [ID 678236 kern.info] Ethernet address
= xx:xx:xx:xx:xx:xx
date starbug unix: [ID 389951 kern.info] mem = 131072K
(0x8000000)
date starbug unix: [ID 930857 kern.info] avail mem = 122134528
date starbug rootnex: [ID 466748 kern.info] root nexus
= Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
date starbug rootnex: [ID 349649 kern.info] pcipsy0 at
root: UPA 0x1f 0x0
date starbug genunix: [ID 936769 kern.info] pcipsy0 is
/pci@1f,0
date starbug pcipsy: [ID 370704 kern.info] PCI-device:
pci@1,1, simba0
date starbug genunix: [ID 936769 kern.info] simba0 is /pci@1f,0/pci@1,1
date starbug pcipsy: [ID 370704 kern.info] PCI-device:
pci@1, simba1
date starbug genunix: [ID 936769 kern.info] simba1 is /pci@1f,0/pci@1
date starbug simba: [ID 370704 kern.info] PCI-device: ide@3,
uata0
date starbug genunix: [ID 936769 kern.info] uata0 is /pci@1f,0/pci@1,1/ide@3
.
.
.
```

Customizing System Message Logging

You can capture additional error messages that are generated by various system processes by modifying the `/etc/syslog.conf` file. By default, `/etc/syslog.conf` directs many system process messages to the `/var/adm` message files. Crash and boot messages are stored here as well. To view `/var/adm` messages, see “How to View System Messages” on page 615.

The `/etc/syslog.conf` file has two columns separated by tabs:

facility.level ...	*action*

facility.level	A *facility* or system source of the message or condition. May be a comma-separated listed of facilities. Facility values are listed in Table 38–2. A *level*, indicates the severity or priority of the condition being logged. Priority levels are listed in Table 38–3.
action	The action field indicates where the messages are forwarded.

The following example shows sample lines from a default `/etc/syslog.conf` file.

```
user.err                                        /dev/sysmsg
user.err                                        /var/adm/messages
user.alert                                      'root, operator'
user.emerg                                      *
```

This means the following user messages are automatically logged:

- User errors are printed to the console and also are logged to the `/var/adm/messages` file.
- User messages requiring immediate action (`alert`) are sent to the root and operator users.
- User emergency messages are sent to individual users.

The most common error condition sources are shown in the table below. The most common priorities are shown in Table 38–3 in order of severity.

TABLE 38–2 Source Facilities for `syslog.conf` Messages

Source	Description
`kern`	The kernel
`auth`	Authentication
`daemon`	All daemons
`mail`	Mail system
`lp`	Spooling system
`user`	User processes

Note - Starting in the Solaris 2.6 release, the number of `syslog` facilities that can be activated in the `/etc/syslog.conf` file is unlimited. In previous releases, the number of facilities was limited to 20.

TABLE 38–3 Priority Levels for `syslog.conf` Messages

Priority	Description
`emerg`	System emergencies
`alert`	Errors requiring immediate correction
`crit`	Critical errors
`err`	Other errors
`info`	Informational messages
`debug`	Output used for debugging
`none`	This setting doesn't log output

▼ How to Customize System Message Logging

1. **Become superuser.**
2. **Using the editor of your choice, edit the `/etc/syslog.conf` file, adding or changing message sources, priorities, and message locations according to the syntax described in `syslog.conf`(4) .**
3. **Exit the file, saving the changes.**

Example—Customizing Message System Logging

This sample `/etc/syslog.conf` `user.emerg` facility sends user emergency messages to root *and* individual users.

```
user.emerg                                      'root, *'
```

Enabling Remote Console Messaging

The following new console features improve your ability to troubleshoot remote systems:

- The `consadm` command enables you to select a serial device as an *auxiliary* (or remote) console. Using the `consadm` command, a system administrator can configure one or more serial ports to display redirected console messages and to host `sulogin` sessions when the system transitions between run levels. This feature enables you to dial in to a serial port with a modem to monitor console messages and participate in `init` state transitions. (See `sulogin(1M)` and the step-by-step procedures below for more information.)

  While you can log in to a system using a port configured as an auxiliary console, it is primarily an output device displaying information that is also displayed on the default console. If boot scripts or other applications read and write to and from the default console, the write output displays on all the auxiliary consoles, but the input is only read from the default console. (See "Using the `consadm` Command During an Interactive Login Session" on page 620 for using the `consadm` command during an interactive login session.)

- Console output now consists of kernel and `syslog` messages written to a new pseudo device, `/dev/sysmsg`. In addition, `rc` script startup messages are written to `/dev/msglog`. Previously, all of these messages were written to `/dev/console`.

  Scripts that direct console output to `/dev/console` need to be changed to `/dev/msglog` if you want to see script messages displayed on the auxiliary consoles. Programs referencing `/dev/console` should be explicitly modified to use `syslog()` or `strlog()` if you want messages to be redirected to an auxiliary device.

- The `consadm` command runs a daemon to monitor auxiliary console devices. Any display device designated as an auxiliary console that disconnects—hangs up or loses carrier—is removed from the auxiliary console device list and is no longer active. Enabling one or more auxiliary consoles does not disable message display on the default console; messages continue to display on `/dev/console`.

Using Auxiliary Console Messaging During Run Level Transitions

Keep the following in mind when using auxiliary console messaging during run level transitions:

- Input cannot come from an auxiliary console if user input is expected for an `rc` script that is run when a system is booting. The input must come from the default console.
- The `sulogin` program, invoked by `init` to prompt for the superuser password when transitioning between run levels, has been modified to send the superuser password prompt to each auxiliary device in addition to the default console device.
- When the system is in single-user mode and one or more auxiliary consoles are enabled using the `consadm` command, a console login session runs on the first device to supply the correct superuser password to the `sulogin` prompt. When the correct password is received from a console device, `sulogin` disables input from all other console devices.
- A message is displayed on the default console and the other auxiliary consoles when one of the consoles assumes single-user privileges. This message indicates which device has become the console by accepting a correct superuser password. If there is a loss of carrier on the auxiliary console running the single-user shell, one of two actions may occur:
  - If the auxiliary console represents a system at run level 1, the system proceeds to the default run level.
  - If the auxiliary console represents a system at run level S, the system displays the `ENTER RUN LEVEL (0-6, s or S):` message on the device where the `init s` or `shutdown` command had been entered from the shell. If there isn't any carrier on that device either, you will have to reestablish carrier and enter the correct run level. The `init` or `shutdown` command will not redisplay the run-level prompt.
- If you are logged in to a system using a serial port, and an `init` or `shutdown` command is issued to transition to another run level, the login session is lost whether this device is the auxiliary console or not. This situation is identical to Solaris releases without auxiliary console capabilities.
- Once a device is selected as an auxiliary console using the `consadm` command, it remains the auxiliary console until the system is rebooted or the auxiliary console is unselected. However, the `consadm` command includes an option to set a device as the auxiliary console across system reboots. (See the procedure below for step-by-step instructions.)

Using the `consadm` Command During an Interactive Login Session

If you want to run an interactive login session by logging in to a system using a terminal that is connected to a serial port, and then using the `consadm` command to see the console messages from the terminal, note the following behavior.

- If you use the terminal for an interactive login session while the auxiliary console is active, the console messages are sent to the `/dev/sysmsg` or `/dev/msglog` devices.
- While you issue commands on the terminal, input goes to your interactive session and not to the default console (`/dev/console`).
- If you run the `init` command to change run levels, the remote console software kills your interactive session and runs the `sulogin` program. At this point, input is accepted only from the terminal and is treated like it's coming from a console device. This allows you to enter your password to the `sulogin` program as described in "Using Auxiliary Console Messaging During Run Level Transitions" on page 619.

  Then, if you enter the correct password on the (auxiliary) terminal, the auxiliary console runs an interactive `sulogin` session, locks out the default console and any competing auxiliary console. This means the terminal essentially functions as the system console.
- From here you can change to run level 3 or go to another run level. If you change run levels, `sulogin` runs again on all console devices. If you exit or specify that the system should come up to run level 3, then all auxiliary consoles lose their ability to provide input. They revert to being display devices for console messages.

  As the system is coming up, you must provide information to `rc` scripts on the default console device. After the system comes back up, the `login` program runs on the serial ports and you can log back into another interactive session. If you've designated the device to be an auxiliary console, you will continue to get console messages on your terminal, but all input from the terminal goes to your interactive session.

▼ How to Enable an Auxiliary (Remote) Console

The `consadm` daemon does not start monitoring the port until after you add the auxiliary console with the `consadm` command. As a security feature, console messages are only redirected until carrier drops, or the auxiliary console device is unselected. This means carrier must be established on the port before you can successfully use the `consadm` command.

See `consadm`(1M) for more information on enabling an auxiliary console.

1. **Log in to the system as superuser.**

2. **Enable the auxiliary console.**

```
# consadm -a devicename
```

3. **Verify that the current connection is the auxiliary console.**

```
# consadm
```

Example—Enabling an Auxiliary (Remote) Console

```
# consadm -a /dev/term/a
# consadm
 /dev/term/a
```

▼ How to Display a List of Auxiliary Consoles

1. **Log in to the system as superuser.**

2. **Select one of the following steps:**

   a. **Display the list of auxiliary consoles.**

```
# consadm
/dev/term/a
```

   b. **Display the list of persistent auxiliary consoles.**

```
# consadm -p
/dev/term/b
```

▼ How to Enable an Auxiliary (Remote) Console Across System Reboots

1. **Log in to the system as superuser.**

2. **Enable the auxiliary console across system reboots.**

```
# consadm -a -p devicename
```

   This adds the device to the list of persistent auxiliary consoles.

3. **Verify that the device has been added to the list of persistent auxiliary consoles.**

```
# consadm
```

Example—Enabling an Auxiliary (Remote) Console Across System Reboots

```
# consadm -a -p /dev/term/a
# consadm
/dev/term/a
```

▼ How to Disable an Auxiliary (Remote) Console

1. **Log in to the system as superuser.**

2. **Select one of the following steps**:

   a. **Disable the auxiliary console.**

   ```
   # consadm -d devicename
   ```

   or

   b. **Disable the auxiliary console and remove it from the list of persistent auxiliary consoles.**

   ```
   # consadm -p -d devicename
   ```

3. **Verify that the auxiliary console has been disabled.**

   ```
   # consadm
   ```

Example—Disabling an Auxiliary (Remote) Console

```
# consadm -d /dev/term/a
# consadm
```

CHAPTER 39

Managing System Crash Information

This section contains information about managing system crash information.

This is a list of the step-by-step instructions in this chapter.

- "How to Display the Current Core Dump Configuration" on page 629
- "How to Set a Core File Name Pattern" on page 629
- "How to Display a Core File Name Pattern" on page 630
- "How to Enable a Per-Process Core File Path" on page 630
- "How to Enable a Global Core File Path" on page 630
- "How to Display the Current Crash Dump Configuration" on page 634
- "How to Modify a Crash Dump Configuration" on page 635
- "How to Examine a Crash Dump" on page 637
- "How to Recover From a Full Crash Dump Directory (Optional)" on page 638
- "How to Disable or Enable Saving Crash Dumps (Optional)" on page 638

System Crashes

System crashes can occur due to hardware malfunctions, I/O problems, and software errors. If the system crashes, it will display an error message on the console, and then write a copy of its physical memory to the dump device. The system will then reboot automatically. When the system reboots, the `savecore` command is executed to retrieve the data from the dump device and write the saved crash dump to your `savecore` directory. The saved crash dump files provide invaluable information to your support provider to aid in diagnosing the problem.

System Crash Files and Core Files

The `savecore` command runs automatically after a system crash to retrieve the crash dump information from the dump device and writes a pair of files called `unix.`*X* and `vmcore.`*X*, where *X* identifies the dump sequence number. Together, these files represent the saved system crash dump information.

Crash dump files are sometimes confused with *core* files, which are images of user applications that are written when the application terminates abnormally.

Crash dump files are saved in a predetermined directory, which by default, is `/var/crash/`*hostname*. In previous Solaris releases, crash dump files were overwritten when a system rebooted—unless you manually enabled the system to save the images of physical memory in a crash dump file. Now the saving of crash dump files is enabled by default.

System crash information is managed with the `dumpadm` command. See "Managing System Crash Dump Information (`dumpadm`)" on page 631 for more information.

Core files are managed with the `coreadm` command. See "Managing Core Files (`coreadm`)" on page 626 for more information.

Managing Core Files (`coreadm`)

The `coreadm` command enables you to manage core files. For example, you can use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining the core files in a specific directory whenever a Solaris process or daemon terminates abnormally.

Limitations of the previous Solaris process core dump features are:

- Process `core` files are placed in their current working directory, and thus all Solaris daemons, which typically `chdir` to the root (`/`) directory as part of their initialization, overwrite each other's `core` files.
- Many system daemons, such as `statd`, perform `setuid` operations but do not produce core files in the event of a problem, for security reasons.

Configurable Core File Paths

Two new configurable `core` file paths that can be enabled or disabled independently of each other are:

- A per-process core file path, which defaults to `core` and is enabled by default. If enabled, the per-process core file path causes a `core` file to be produced when the

process terminates abnormally. The per-process path is inherited by a new process from its parent process.

When generated, a per-process core file is owned by the owner of the process with read/write permissions for the owner. Only the owning user can view this file.

- A global core file path, which defaults to `core` and is disabled by default. If enabled, an *additional* core file with the same content as the per-process core file is produced by using the global core file path.

  When generated, a global core file is owned by superuser with read/write permissions for superuser only. Non-privileged users cannot view this file.

When a process terminates abnormally, it produces a `core` file in the current directory as in previous Solaris releases. But if the global core file path is enabled and set to `/corefiles/core`, for example, then each process that expires produce *two* core files: one in the current working directory and one in the `/corefiles` directory.

By default, the Solaris core paths and core file retention remain the same:

- A `setuid` process does not produce core files using either the global or per-process path.
- The global core file path is disabled.
- The per-process core file path is enabled.
- The per-process core file path is set to `core`.

Expanded Core File Names

If a global core file directory is enabled, `core` files can be distinguished from one another by using the variables described in the following table.

Variable Name	**Variable Definition**
`%p`	Process ID
`%u`	Effective user ID
`%g`	Effective group ID
`%f`	Executable file name
`%n`	System node name, equivalent to the `uname -n` output
`%m`	Machine name, equivalent to the `uname -m` output

Variable Name	Variable Definition
%t	Decimal value of time(2) system call
%%	Literal %

For example, if the global core file path is set to:

`/var/core/core.%f.%p`

and a `sendmail` process with PID `12345` terminates abnormally, it produces the following `core` file:

`/var/core/core.sendmail.12345`

Setting the Core File Name Pattern

You can set a core file name pattern on a global basis or a per-process basis, and you can specify whether you want these settings saved across a system reboot.

For example, the following `coreadm` command sets the global core file pattern for all processes started by the `init` process. This pattern will persist across system reboots.

```
$ coreadm -i /var/core/core.%f.%p
```

Global core values are stored in the `/etc/coreadm.conf` file, which means these settings are saved across a system reboot.

This `coreadm` command sets the per-process core file name pattern for all processes:

```
$ coreadm -p /var/core/core.%f.%p $$
```

The `$$` symbols represent a placeholder for the process ID of the currently running shell. The per-process core file name pattern is inherited by all child processes.

Once a global or per-process core file name pattern is set, it must be enabled with the `coreadm -e` command. See the procedures below for more information.

You can set the core file name pattern for all processes run during a user's login session by putting the command in a user's `$HOME/.profile` or `.login` file.

Enabling `setuid` Programs to Produce Core Files

You can use the `coreadm` command to enable or disable `setuid` programs to produce core files for all system processes or on a per-process basis by setting the following paths:

- If the global `setuid` option is enabled, a global core file path allows all `setuid` programs on a system to produce `core` files.
- If the per-process `setuid` option is enable, a per-process core file path allows specific `setuid` processes to produce `core` files.

By default, both flags are disabled. For security reasons, the global core file path must be a full pathname, starting with a leading `/`. If superuser disables per-process core files, individual users cannot obtain core files.

The `setuid` core files are owned by superuser with read/write permissions for superuser only. Ordinary users cannot access them even if the process that produced the `setuid` core file was owned by an ordinary user.

See `coreadm`(1M) for more information.

▼ How to Display the Current Core Dump Configuration

Use the `coreadm` command without any options to display the current core dump configuration.

```
$ coreadm
               global core file pattern: /var/core/core.%f.%p
                 init core file pattern: core
                      global core dumps: enabled
                 per-process core dumps: enabled
                global setid core dumps: enabled
           per-process setid core dumps: disabled
               global core dump logging: disabled
```

▼ How to Set a Core File Name Pattern

1. **Determine whether you want to set a per-process or global core file and select one of the following:**

   a. **Set a per-process file name pattern.**

   ```
   # coreadm -p $HOME/corefiles/%f.%p $$
   ```

   b. **Set a global file name pattern.**

   Become superuser first.

   ```
   # coreadm -g /var/corefiles/%f.%p
   ```

▼ How to Display a Core File Name Pattern

Use the following `coreadm` command to inquire about the core file settings of the current process. The `$$` symbols represent a placeholder for the process ID of the running shell.

```
$ coreadm $$
278:    core.%f.%p
```

Superuser can inquire about any user's core file settings by using `coreadm` *process ID*. Ordinary users can only inquire about the core file settings of their own processes.

▼ How to Enable a Per-Process Core File Path

1. **Become superuser.**
2. **Enable a per-process core file path.**

```
# coreadm -e process
```

3. **Display the current process core file path to verify the configuration.**

```
$ coreadm $$
1180:   /home/kryten/corefiles/%f.%p
```

▼ How to Enable a Global Core File Path

1. **Become superuser.**
2. **Enable a global core file path.**

```
# coreadm -e global -g /var/core/core.%f.%p
```

3. **Display the current process core file path to verify the configuration.**

```
# coreadm
      global core file pattern: /var/core/core.%f.%p
        init core file pattern: core
             global core dumps: enabled
       per-process core dumps: enabled
      global setid core dumps: disabled
  per-process setid core dumps: disabled
      global core dump logging: disabled
```

Troubleshooting Core File Problems

Error Message

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

Cause

You have an obsolete parameter that allows `setuid` core files in your `/etc/system` file.

Solution

Remove `allow_setid_core=1` from the `/etc/system` file. Then use the `coreadm` command to enable global `setuid` core file paths.

Managing System Crash Dump Information (`dumpadm`)

This section describes how to manage system crash information in the Solaris environment.

System Crash Dump Features

This section describes how to manage system crash dump information in the Solaris environment.

- The new `dumpadm` command, which allows system administrators to configure crash dumps of the operating system. The `dumpadm` configuration parameters include the dump content, dump device, and the directory in which crash dump files are saved. See "The `dumpadm` Command" on page 632 for more information about the `dumpadm` command.
- Dump data is now stored in compressed format on the dump device. Kernel crash dump images can be as big as 4 Gbytes or more. Compressing the data means faster dumping and less disk space needed for the dump device.
- Saving crash dump files is run in the background when a dedicated dump device—not the swap area—is part of the dump configuration. This means a booting system does not wait for the `savecore` command to complete before going to the next step. On large memory systems, the system can be available before `savecore` completes.
- System crash dump files, generated by the `savecore` command, are now saved by default.
- The `savecore -L` command is a new feature which enables you to get a crash dump of the live running Solaris operating environment. This command is intended for troubleshooting a running system by taking a snapshot of memory during some bad state—such as a transient performance problem or service outage. If the system is up and you can still run some commands, you can execute the `savecore -L` to save a snapshot of the system to the dump device, and then immediately write out the crash dump files to your `savecore` directory. Because the system is still running, you may only use `savecore -L` if you have configured a dedicated dump device.

The `dumpadm` Command

The `/usr/sbin/dumpadm` command manages a system's crash dump configuration parameters. The following table describes `dumpadm`'s configuration parameters.

Dump Parameter	**Description**
dump device	The device that stores dump data temporarily as the system crashes. When the dump device is not the swap area, `savecore` runs in the background, which speeds up the boot process.
savecore directory	The directory that stores system crash dump files.

Dump Parameter	Description
dump content	Type of data, kernel memory or all of memory, to dump.
minimum free space	Minimum amount of free space required in the `savecore` directory after saving crash dump files. If no minimum free space has been configured, the default is one megabyte.

See `dumpadm`(1M) for more information.

The dump configuration parameters managed by the `dumpadm` command are stored in the `/etc/dumpadm.conf` file.

Note - Do not `/etc/dumpadm.conf` edit manually. This could result in an inconsistent system dump configuration.

How the `dumpadm` Command Works

During system startup, the `dumpadm` command is invoked by the `/etc/init.d/savecore` script to configure crash dumps parameters based on information in the `/etc/dumpadm.conf` file.

Specifically, it initializes the dump device and the dump content through the `/dev/dump` interface.

After the dump configuration is complete, the `savecore` script looks for the location of the crash dump file directory by parsing the content of `/etc/dumpadm.conf` file. Then, `savecore` is invoked to check for crash dumps. It will also check the content of the `minfree` file in the crash dump directory.

Saving Crash Dumps

You can examine the control structures, active tables, memory images of a live or crashed system kernel, and other information about the operation of the kernel by using the `crash` or `adb` utilities. Using `crash` or `adb` to its full potential requires a detailed knowledge of the kernel, and is beyond the scope of this manual. See `crash`(1M) or `adb`(1)for more details on using these utilities.

Additionally, crash dumps saved by `savecore` can be useful to send to a customer service representative for analysis of why the system is crashing. If you will be sending crash dump files to a customer service representative, perform the first two tasks listed in “Managing System Crash Information Task Map” on page 634.

The next section describes how to manage system crash information with the `dumpadm` command.

Managing System Crash Information Task Map

TABLE 39–1 Managing System Crash Information Task Map

Task	Description	For Instructions, Go To
1. Display the Current Crash Dump Configuration	Display the current crash dump configuration by using the `dumpadm`command.	"How to Display the Current Crash Dump Configuration" on page 634
2. Modify the Crash Dump Configuration	Use the `dumpadm` command to specify the type of data to dump, whether or not the system will use a dedicated dump device, the directory for saving crash dump files, and the amount of space that must remain available after crash dump files are written.	"How to Modify a Crash Dump Configuration" on page 635
3. Examine a Crash Dump File	Use the `crash` command to view crash dump files.	"How to Examine a Crash Dump" on page 637
4. Recover From a Full Crash Dump Directory	*Optional.* The system crashes but there is no room in the `savecore` directory, and you want to save some critical system crash dump information.	"How to Recover From a Full Crash Dump Directory (Optional)" on page 638
4. Disable or Enable the Saving of Crash Dump Files	*Optional.* Use the `dumpadm` command to disable or enable the saving the crash dump files. Saving crash dump files is enabled by default.	"How to Disable or Enable Saving Crash Dumps (Optional)" on page 638

▼ How to Display the Current Crash Dump Configuration

1. Become superuser.

2. **Display the current crash dump configuration by using the** `dumpadm` **command without any options.**

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/pluto
  Savecore enabled: yes
```

The above example output means:

- The dump content is kernel memory pages.
- Kernel memory will be dumped on a swap device, `/dev/dsk/c0t3d0s1`. You can identify all your swap areas with the `swap -l` command.
- System crash dump files will be written in the `/var/crash/venus` directory.
- Saving crash dump files is enabled.

▼ How to Modify a Crash Dump Configuration

1. **Become superuser.**

2. **Identify the current crash dump configuration by using the** `dumpadm` **command.**

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
```

This the default dump configuration for a system running the Solaris 8 release.

3. **Modify the crash dump configuration by using the** `dumpadm` **command.**

```
# dumpadm -c content -d dump-device -m nnnk | nnnm | nnn% -n -s savecore-dir
```

–c *content*	Specifies the type of data to dump: kernel memory or all of memory. The default dump content is kernel memory.
–d *dump-device*	Specifies the device that stores dump data temporarily as the system crashes. The primary swap device is the default dump device.
–m *nnnk* | *nnnm* | *nnn%*	Specifies the minimum free disk space for saving crash dump files by creating a `minfree` file in the current `savecore` directory. This parameter can be specified in kilobytes (`nnnk`) , megabytes (`nnnm`) or file system size percentage (`nnn%`). The `savecore` command consults this file prior to writing the crash dump files. If writing the crash dump files, based on their size, would decrease the amount of free space below the `minfree` threshold, the dump files are not written and an error message is logged. See "How to Recover From a Full Crash Dump Directory (Optional)" on page 638 for recovering from this scenario.
–n	Specifies that `savecore` should not be run when the system reboots. This dump configuration is not recommended. If system crash information is written to the swap device, and `savecore` is not enabled, the crash dump information is overwritten when the system begins to swap.
–s	Specifies an alternate directory for storing crash dump files. The default directory is `/var/crash/hostname` where `hostname` is the output of the `uname -n` command.

Example—Modifying a Crash Dump Configuration

In this example, all of memory is dumped to the dedicated dump device, `/dev/dsk/c0t1d0s1`, and the minimum free space that must be available after the crash dump files are saved is 10% of the file system space.

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
 # dumpadm -c all -d /dev/dsk/c0t1d0s1 -m 10%
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
```

▼ How to Examine a Crash Dump

1. **Become superuser.**

2. **Examine a crash dump by using the** `crash` **utility.**

```
# /usr/sbin/crash [-d crashdump-file] [-n name-list] [-w output-file]
```

–d *crashdump-file*	Specifies a file to contain the system memory image. The default crash dump file is `/dev/mem`.
–n *name-list*	Specifies a text file to contain symbol table information if you want to examine symbolic access to the system memory image. The default file name is `/dev/ksyms`.
–w *output-file*	Specifies a file to contain output from a crash session. The default is standard output.

3. **Display crash status information.**

```
# /usr/sbin/crash
dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout
> status
   .
   .
   .
> size buf proc queue
   .
   .
   .
```

Example—Examining a Crash Dump

The following example shows sample output from the `crash` utility. Information about status, and about the buffer, process, and queue size is displayed.

```
# /usr/sbin/crash
dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout
> status
system name: SunOS
release: 5.8
```

(continued)

(Continuation)

```
node name: earth
version: s28_25
machine name: sun4m
time of crash: Wed Jun 30 16:02:31 1999
age of system: 18 min.
panicstr:
panic registers:
 pc: 0      sp: 0
> size buf proc queue
120
1808
96
```

▼ How to Recover From a Full Crash Dump Directory (Optional)

In this scenario, the system crashes but there is no room in the `savecore` directory, and you want to save some critical system crash dump information.

1. **Log in as superuser after the system reboots.**

2. **Clear out the savecore directory, usually** `/var/crash/`***hostname,*** **by removing existing crash dump files that have already been sent to your service provider. Or, run the** `savecore` **command and specify an alternate directory that has sufficient disk space. (See the next step.)**

3. **Manually run the** `savecore` **command and if necessary, specify an alternate savecore directory.**

```
# savecore [ directory ]
```

▼ How to Disable or Enable Saving Crash Dumps (Optional)

1. **Become superuser.**

2. **Disable or enable the saving of crash dumps on your system by using the** `dumpadm` **command.**

Example—Disabling the Saving of Crash Dumps

This example illustrates how to disable the saving of crash dumps on your system.

```
# dumpadm -n
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: no
```

Example—Enabling the Saving of Crash Dumps

This example illustrates how to enable the saving of crash dump on your system.

```
# dumpadm -y
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
```

CHAPTER 40

Troubleshooting Miscellaneous Software Problems

This chapter describes miscellaneous software problems that may occur occasionally and are relatively easy to fix. Troubleshooting miscellaneous software problems includes solving problems that aren't related to a specific software application or topic, such as unsuccessful reboots and full file systems. Resolving these problems are described in the following sections.

This is a list of information in this chapter.

- "What to Do If Rebooting Fails" on page 641
- "What to Do if a System Hangs" on page 643
- "What to Do if a File System Fills Up" on page 644
- "What to Do if File ACLs Are Lost After Copy or Restore" on page 645
- "Troubleshooting Backup Problems" on page 645

What to Do If Rebooting Fails

If the system does not reboot completely, or if it reboots and then crashes again, there may be a software or hardware problem that is preventing the system from booting successfully.

Problem — A System Won't Boot Because ...	How to Fix the Problem
The system can't find `/platform/'uname -m'/kernel/unix`.	You may need to change the `boot-device` setting in the PROM on a SPARC system. See "Booting a System (Tasks)" in *System Administration Guide, Volume 1* for information on changing the default boot device.
There is no default boot device on an IA system. The message displayed is: `Not a UFS filesystem.`	Boot the system using the Configuration Assistant/Boot diskette and select the disk from which to boot.
There's an invalid entry in the `/etc/passwd` file.	See "Shutting Down and Booting a System (Overview)" in *System Administration Guide, Volume 1* for information on recovering from an invalid `passwd` file.
There's a hardware problem with a disk or another device.	Check the hardware connections: ■ Make sure the equipment is plugged in. ■ Make sure all the switches are set properly. ■ Look at all the connectors and cables, including the Ethernet cables. ■ If all this fails, turn off the power to the system, wait 10 to 20 seconds, and then turn on the power again.

If none of the above suggestions solve the problem, contact your local service provider.

SPARC: Troubleshooting 64–bit Solaris Boot Problems

After the 64–bit Solaris release is installed on an UltraSPARC system, the 64–bit kernel will be booted automatically unless any of the following conditions are true:

- A FLASH PROM upgrade may be required on an UltraSPARC system before it can successfully boot the 64–bit kernel. Refer to your hardware manufacturer's documentation to determine whether your UltraSPARC system requires a firmware upgrade.
- The Open Boot PROM `boot-file` parameter is set to `kernel/unix`. If booting the 64–bit kernel fails and this parameter is set, unset it, and reboot the system.
- On some UltraSPARC systems, the 64-bit Solaris kernel is not booted by default, even when the system is completely installed with all the 64-bit Solaris

components and the correct firmware is installed. Without booting the 64-bit Solaris kernel, 64-bit applications are unable to run.

To find out more about this issue, and how to enable booting the 64-bit Solaris kernel by default, see `boot`(1m).

You can always discover which Solaris kernel the system is currently running by using the `isainfo -kv` command.

```
$ isainfo -kv
64-bit sparcv9 kernel modules
```

This output means the system is running the 64–bit Solaris kernel.

You cannot boot the 64-bit Solaris operating environment on a 32-bit Solaris system.

What to Do if a System Hangs

A system can freeze or hang rather than crash completely if some software process is stuck. Follow these steps to recover from a hung system.

1. Determine whether the system is running a window environment and follow the suggestions listed below. If these suggestions don't solve the problem, go to step 2.
   - Make sure the pointer is in the window where you are typing the commands
   - Press Control-q in case the user accidentally pressed Control-s, which freezes the screen. Control-s freezes only the window, not the entire screen. If a window is frozen, try using another window.
   - If possible, log in remotely from another system on the network. Use the `pgrep` command to look for the hung process. If it looks like the window system is hung, identify the process and kill it.
2. Press Control-\ to force a "quit" in the running program and (probably) write out a `core` file.
3. Press Control-c to interrupt the program that might be running.
4. Log in remotely and attempt to identify and kill the process that is hanging the system.
5. Log in remotely, become superuser and reboot the system.
6. If the system still does not respond, force a crash dump and reboot. See Chapter 39 for information on forcing a crash dump and booting.
7. If the system still does not respond, turn the power off, wait a minute or so, then turn the power back on.

8. If you cannot get the system to respond at all, contact your local service provider for help.

What to Do if a File System Fills Up

When the root (/) file system or any other file system fills up, you will see the following message in the console window:

```
.... file system full
```

There are several reasons why a file system fills up. The following sections describe several scenarios for recovering from a full file system. See Chapter 28 for information on routinely cleaning out old and unused files to prevent full file systems.

File System Fills Up Because a Large File or Directory Was Created

Reason Error Occurred	How to Fix the Problem
Someone accidentally copied a file or directory to the wrong location. This also happens when an application crashes and writes a large `core` file into the file system.	Log in as superuser and use the `ls -tl` command in the specific file system to identify which large file is newly created and remove it. See "How to Find and Delete `core` Files" on page 478 to remove `core` files.

A TMPFS File System is Full Because the System Ran Out of Memory

Reason Error Occurred	How to Fix the Problem
This can occur if `TMPFS` is trying to write more than it is allowed or some current processes are using a lot of memory.	See `tmpfs`(7FS) for information on recovering from `tmpfs`-related error messages.

What to Do if File ACLs Are Lost After Copy or Restore

Reason Error Occurred	How to Fix the Problem
If files or directories with ACLs are copied or restored into the `/tmp` directory, the ACL attributes are lost. The `/tmp` directory is usually mounted as a temporary file system, which doesn't support UFS file system attributes such as ACLs.	Copy or restore files into the `/var/tmp` directory instead.

Troubleshooting Backup Problems

This section describes some basic troubleshooting techniques to use when backing up and restoring data.

The root (/) File System Fills Up After You Back Up a File System

You back up a file system, and the root (/) file system fills up. Nothing is written to the media, and the `ufsdump` command prompts you to insert the second volume of media.

Reason Error Occurred	How to Fix the Problem
If you used an invalid destination device name with the –`f` option, the `ufsdump` command wrote to a file in the `/dev` directory of the root (/) file system, filling it up. For example, if you typed `/dev/rmt/st0` instead of `/dev/rmt/0`, the backup file `/dev/rmt/st0` was created on the disk rather than being sent to the tape drive.	Use the `ls -tl` command in the `/dev` directory to identify which file is newly created and abnormally large, and remove it.

Make Sure the Backup and Restore Commands Match

You can only use `ufsrestore` to restore files backed up with `ufsdump`. If you back up with `tar`, restore with `tar`. If you use the `ufsrestore` command to restore a tape that was written with another command, an error message tells you that the tape is not in `ufsdump` format.

Check to Make Sure You Have the Right Current Directory

It is easy to restore files to the wrong location. Because the `ufsdump` command always copies files with full path names relative to the root of the file system, you should usually change to the root directory of the file system before running `ufsrestore`. If you change to a lower-level directory, after you restore the files you will see a complete file tree created under that directory.

Use the Old `restore` Command to Restore Multivolume Diskette Backups

You cannot use the `ufsrestore` command to restore files from a multivolume backup set of diskettes made with the `dump` command. You must restore the files on a SunOS 4.1 system.

Interactive Commands

When you use the interactive command, a `ufsrestore>` prompt is displayed, as shown in this example:

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump   date: Mon Jul 12 14:06:54 1999
Dumped from: the epoch
Level 0 dump of a partial file system on venus:/var/adm/acct
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore >
```

At the `ufsrestore>` prompt, you can use the commands listed on "The ufsdump and ufsrestore Commands (Reference)" in *System Administration Guide, Volume 1* to find files, create a list of files to be restored, and restore them.

CHAPTER 41

Troubleshooting File Access Problems

This is a list of troubleshooting topics in this chapter.

- "Solving Problems With Search Paths (`Command not found`)" on page 649
- "Solving File Access Problems" on page 652
- "Recognizing Problems With Network Access" on page 653

Users frequently experience problems—and call on a system administrator for help—because they cannot access a program, a file, or a directory that they could previously use. Whenever you encounter such a problem, investigate one of three areas:

- The user's search path may have been changed, or the directories in the search path may not be in the proper order.
- The file or directory may not have the proper permissions or ownership.
- The configuration of a system accessed over the network may have changed.

This chapter briefly describes how to recognize problems in each of these three areas and suggests possible solutions.

Solving Problems With Search Paths (`Command not found`)

A message of `Command not found` indicates one of the following:

- The command is not available on the system.
- The command directory is not in the search path.

To fix a search path problem, you need to know the pathname of the directory where the command is stored.

If the wrong version of the command is found, a directory that has a command of the same name is in the search path. In this case, the proper directory may be later in the search path or may not be present at all.

You can display your current search path by using the `echo $PATH` command.

```
$ echo $PATH
/home/kryten/bin:/sbin:/usr/sbin:/usr/bin:/usr/dt:/usr/dist/exe
```

Use the `which` command to determine whether you are running the wrong version of the command.

```
$ which maker
/usr/doctools/frame5.1/bin/maker
```

Note - The `which` command looks in the `.cshrc` file for path information. The `which` command may give misleading results if you execute it from the Bourne or Korn shell and you have a `.cshrc` file than contains aliases for the `which` command. To ensure accurate results, use the `which` command in a C shell, or, in the Korn shell, use the `whence` command.

▼ How to Diagnose and Correct Search Path Problems

1. **Display the current search path to verify that the directory for the command is not in your path or that it isn't misspelled.**

   ```
   $ echo $PATH
   ```

2. **Check the following:**
   - Is the search path correct?
   - Is the search path listed before other search paths where another version of the command is found?
   - Is the command in one of the search paths?

   If the path needs correction, go to step 3. Otherwise, go to step 4.

3. **Add the path to the appropriate file, as shown in this table.**

Shell	File	Syntax	Notes
Bourne and Korn	`$HOME/.profile`	`$ PATH=`*`$HOME/bin:/sbin:/usr/local /bin ...`* `$ export PATH`	A colon separates path names.

C	`$HOME/.cshrc`
	or
	`$HOME/.login`
		hostname% `set path=(`*`~bin / sbin /usr/local/bin ...)`*	A blank space separates path names.

4. Activate the new path as follows:

Shell	File Where Path Is Located	Activate The Path With ...
Bourne and Korn	`.profile`	`$ . ./.profile`
C	`.cshrc`	*hostname%* **`source .cshrc`**
	`.login`	*hostname%* **`source .login`**

5. Verify the path using the command shown below.

```
$ which command
```

Example—Diagnosing and Correcting Search Path Problems

This example shows that the `mytool` executable is not in any of the directories in the search path using the `which` command.

```
venus% mytool
mytool: Command not found
venus% which mytool
no mytool in /sbin /usr/sbin /usr/bin /etc /home/ignatz/bin  .
venus% echo $PATH
```

(continued)

(Continuation)

```
/sbin /usr/sbin /usr/bin /etc /home/ignatz/bin
venus% vi ~.cshrc
(Add appropriate command directory to the search path)
venus% source .cshrc
venus% mytool
```

If you cannot find a command, look at the man page for its directory path. For example, if you cannot find the `lpsched` command (the `lp` printer daemon), `lpsched`(1M) tells you the path is `/usr/lib/lp/lpsched`.

Solving File Access Problems

When users cannot access files or directories that they previously could access, the permissions or ownership of the files or directories probably has changed.

Changing File and Group Ownerships

Frequently, file and directory ownerships change because someone edited the files as superuser. When you create home directories for new users, be sure to make the user the owner of the dot (`.`) file in the home directory. When users do not own "`.`" they cannot create files in their own home directory.

Access problems can also arise when the group ownership changes or when a group of which a user is a member is deleted from the `/etc/group` database.

See Table 41–1 for information about how to change the permissions or ownership of a file that you are having problems accessing.

TABLE 41–1 Solving File Access Problems

If You Need to Change the ...	Use the ...	For More Details, See ...
Permission on a file	`chmod`(1) command	"How to Change Permissions in Absolute Mode" on page 306
Ownership of a file	`chown`(1) command	"How to Change the Owner of a File" on page 301
Group ownership of a file	`chgrp`(1) command	"How to Change Group Ownership of a File" on page 302

Recognizing Problems With Network Access

If users have problems using the `rcp` remote copy command to copy files over the network, the directories and files on the remote system may have restricted access by setting permissions. Another possible source of trouble is that the remote system and the local system are not configured to allow access.

See *System Administration Guide, Volume 3* for information about problems with network access and problems with accessing systems through AutoFS.

CHAPTER 42

Troubleshooting Printing Problems

This chapter explains how to troubleshoot printing problems that may occur when you set up or maintain printing services.

This is a list of step-by-step instructions in this chapter.

- "How to Troubleshoot No Printer Output" on page 661
- "How to Troubleshoot Incorrect Output" on page 675
- "How to Unhang the LP Print Service" on page 681
- "How to Troubleshoot an Idle (Hung) Printer" on page 682
- "How to Resolve Conflicting Printer Status Messages" on page 684

See Chapter 2 for information about printing and the LP print service.

Tips on Troubleshooting Printing Problems

Sometimes after setting up a printer, you find that nothing prints. Or, you may get a little farther in the process: something prints, but it is not what you expect—the output is incorrect or illegible. Then, when you get past these problems, other problems may occur, such as:

- LP commands hanging
- Printers becoming idle
- Users getting conflicting messages

Note - Although many of the suggestions in this chapter are relevant to parallel printers, they are geared toward the more common serial printers.

Troubleshooting No Output (Nothing Prints)

When nothing prints, there are three general areas to check:

- The printer hardware
- The network
- The LP print service

If you get a banner page, but nothing else, this is a special case of incorrect output. See "Troubleshooting Incorrect Output" on page 658.

Check the Hardware

The hardware is the first area to check. As obvious as it sounds, you should make sure that the printer is plugged in and turned on. In addition, you should refer to the manufacturer's documentation for information about hardware settings. Some computers use hardware switches that change the characteristics of a printer port.

The printer hardware includes the printer, the cable that connects it to the computer, and the ports into which the cable plugs at each end. As a general approach, you should work your way from the printer to the computer. Check the printer. Check where the cable connects to the printer. Check the cable. Check where the cable connects to the computer.

Check the Network

Problems are more common with remote print requests—those going from a print client to a print server. You should make sure that network access between the print server and print clients is enabled.

If the network is running the Network Information Service Plus (NIS+), see the *Solaris Naming Administration Guide* for instructions to enable access between systems. If the network is not running the Network Information Service (NIS) or NIS+, before you set up print servers and print clients, include the Internet address and system name for each client system in the `/etc/hosts` file on the print server. Also, the Internet address and system name for the print server must be included in the `/etc/hosts` file of each print client system.

Check the LP Print Service

For printing to work, the LP scheduler must be running on both the print server and print client. If it is not running, you need to start it using the `/usr/lib/lp/lpsched` command. If you have trouble starting the scheduler, see “How to Restart the Print Scheduler” on page 94.

In addition to the scheduler running, a printer must be enabled and accepting requests before it will produce any output. If the LP print service is not accepting requests for a printer, the submitted print requests are rejected. Usually, in that instance, the user receives a warning message after submitting a print request. If the LP print service is not enabled for a printer, print requests remain queued on the system until the printer is enabled.

In general, you should analyze a printing problem as follows:

- Follow the path of the print request step-by-step.
- Examine the status of the LP print service at each step.
  - Is the configuration correct?
  - Is the printer accepting requests?
  - Is the printer enabled to process requests?
- If the request is hanging on transmission, set up `lpr.debug` in `syslog.conf` to display the flow.
- If the request is hanging locally, examine the `lpsched` log (`/var/lp/logs/lpsched`).
- If the request is hanging locally, have notification of the printer device errors (faults) mailed to you, and re-enable the printer.

The procedures found in “Troubleshooting Printing Problems” on page 661 use this strategy to help you troubleshoot various problems with the LP print service.

If basic troubleshooting of the LP print service does not solve the problem, you need to follow the troubleshooting steps for the specific client/server case that applies:

- SunOS 5.8 print client using a SunOS 5.8 print server (for instructions, see “To check printing from a SunOS 5.8 print client to a SunOS 5.8 print server:” on page 668)
- SunOS 5.8 print client using a SunOS 4.1 print server (for instructions, see “To check printing from a SunOS 5.8 print client to a SunOS 4.1 print server:” on page 669)
- SunOS 4.1 print client using a SunOS 5.8 print server (for instructions, see “To check printing from a SunOS 4.1 client to a SunOS 5.8 print server: ” on page 672)

Troubleshooting Incorrect Output

If the printer and the print service software are not configured correctly, the printer may print, but it may provide output that is not what you expect.

Check the Printer Type and File Content Type

If you used the wrong printer type when you set up the printer with the LP print service, inappropriate printer control characters can be sent to the printer. The results are unpredictable: nothing may print, the output may be illegible, or the output may be printed in the wrong character set or font.

If you specified an incorrect file content type, the banner page may print, but that is all. The file content types specified for a printer indicate the types of files the printer can print directly, without filtering. When a user sends a file to the printer, the file is sent directly to the printer without any attempt to filter it. The problem occurs if the printer cannot handle the file content type.

When setting up print clients, you increase the chance for a mistake because the file content types must be correct on both the print server and the print client. If you set up the print client as recommended with `any` as the file content type, files are sent directly to the print server and the print server determines the need for filtering. Therefore, the file content types have to be specified correctly only on the server.

You can specify a file content on the print client to off-load filtering from the server to the client, but the content type must be supported on the print server.

Check the `stty` Settings

Many formatting problems can result when the default `stty` (standard terminal) settings do not match the settings required by the printer. The following sections describe what happens when some of the settings are incorrect.

Wrong Baud Settings

When the baud setting of the computer does not match the baud setting of the printer, usually you get some output, but it does not look like the file you submitted for printing. Random characters are displayed, with an unusual mixture of special characters and undesirable spacing. The default for the LP print service is 9600 baud.

Note - If a printer is connected by a parallel port, the baud setting is irrelevant.

Wrong Parity Setting

Some printers use a parity bit to ensure that data received for printing has not been garbled during transmission. The parity bit setting for the computer and the printer must match. If they do not match, some characters either will not be printed at all, or will be replaced by other characters. In this case, the output looks approximately correct; the word spacing is all right and many letters are in their correct place. The LP print service does not set the parity bit by default.

Wrong Tab Settings

If the file contains tabs, but the printer expects no tabs, the printed output may contain the complete contents of the file, but the text may be jammed against the right margin. Also, if the tab settings for the printer are incorrect, the text may not have a left margin, it may run together, it may be concentrated to a portion of the page, or it may be incorrectly double-spaced. The default is for tabs to be set every eight spaces.

Wrong Return Setting

If the output is double-spaced, but it should be single-spaced, either the tab settings for the printer are incorrect or the printer is adding a line feed after each return. The LP print service adds a return before each line feed, so the combination causes two line feeds.

If the print zigzags down the page, the `stty` option `onlcr` that sends a return before every line feed is not set. The `stty=onlcr` option is set by default, but you may have cleared it while trying to solve other printing problems.

Troubleshooting Hung LP Commands

If you type any of the LP commands (such as `lpsystem`, `lpadmin`, or `lpstat`) and nothing happens (no error message, status information, or prompt is displayed), chances are something is wrong with the LP scheduler. Such a problem can usually be resolved by stopping and restarting the LP scheduler. See “How to Stop the Print Scheduler” on page 93 for instructions.

Troubleshooting Idle (Hung) Printers

You may find a printer that is idle, even though it has print requests queued to it. A printer may seem idle when it should not be for one of the following reasons:

- The current print request is being filtered.
- The printer has a fault.

- Networking problems may be interrupting the printing process.

Check the Print Filters

Slow print filters run in the background to avoid tying up the printer. A print request that requires filtering will not print until it has been filtered.

Check Printer Faults

When the LP print service detects a fault, printing resumes automatically, but not immediately. The LP print service waits about five minutes before trying again, and continues trying until a request is printed successfully. You can force a retry immediately by enabling the printer.

Check Network Problems

When printing files over a network, you may encounter the following types of problems:

- Requests sent to print servers may back up in the client system (local) queue.
- Requests sent to print servers may back up in the print server (remote) queue.

Print Requests Backed Up in the Local Queue

Print requests submitted to a print server may back up in the client system queue for the following reasons:

- The print server is down.
- The printer is disabled on the print server.
- The network between the print client and print server is down.
- Underlying network software was not set up properly.

While you are tracking the source of the problem, you should stop new requests from being added to the queue. See “How to Accept or Reject Print Requests for a Printer” on page 110 for more information.

Print Requests Backed Up in the Remote Queue

If print requests back up in the print server queue, the printer has probably been disabled. When a printer is accepting requests, but not processing them, the requests are queued to print. Unless there is a further problem, once the printer is enabled, the print requests in the queue should print.

Troubleshooting Conflicting Status Messages

A user may enter a print request and be notified that the client system has accepted it, then receive mail from the print server that the print request has been rejected. These conflicting messages may occur for the following reasons:

- The print client may be accepting requests, while the print server is rejecting requests.
- The definition of the printer on the print client might not match the definition of that printer on the print server. More specifically, the definitions of the print job components, like filters, character sets, print wheels, or forms are not the same on the client and server systems.

You should check that identical definitions of these job components are registered on both the print clients and print servers so that local users can access printers on the print servers.

Troubleshooting Printing Problems

This section contains step-by-step instructions that explain:

- How to troubleshoot no output
- How to troubleshoot incorrect output
- How to unhang the LP commands
- How to troubleshoot an idle (hung) printer
- How to resolve conflicting status messages

▼ How to Troubleshoot No Printer Output

This task includes the following troubleshooting procedures to try when you submit a print request to a printer and nothing prints:

- Check the hardware (“To check the hardware:” on page 662).
- Check the network (“To check the network:” on page 663).
- Check the LP print service basic functions (“To check the basic functions of the LP print service: ” on page 664).
- Check printing from a SunOS 5.8 print client to a SunOS 5.8 print server (“To check printing from a SunOS 5.8 print client to a SunOS 5.8 print server:” on page 668).
- Check printing from a SunOS 5.8 print client to a SunOS 4.1 print server (“To check printing from a SunOS 5.8 print client to a SunOS 4.1 print server:” on page 669).

- Check printing from a SunOS 4.1 print client to a SunOS 5.8 print server (“To check printing from a SunOS 4.1 client to a SunOS 5.8 print server: ” on page 672).

Try the first three procedures in the order in which they are listed, before going to the specific print client/server case that applies. However, if the banner page prints, but nothing else does, turn to the instructions under “How to Troubleshoot Incorrect Output” on page 675.

To check the hardware:

1. **Check that the printer is plugged in and turned on.**

2. **Check that the cable is connected to the port on the printer and to the port on the system or server.**

3. **Make sure that the cable is the correct cable and that it is not defective.**

   Refer to the manufacturer‘s documentation. If the printer is connected to a serial port, verify that the cable supports hardware flow control; a NULL modem adapter supports this. The table below shows the pin configuration for NULL modem cables.

TABLE 42–1 Pin Configuration for NULL Modem Cables

	Host	**Printer**
Mini-Din-8	**25-Pin D-sub**	**25-Pin D-sub**
-	1 (FG)	1(FG)
3(TD)	2(TD)	3(RD)
5(RD)	3(RD)	2(TD)
6(RTS)	4(RTS)	5(CTS)
2(CTS)	5(CTS)	4(RTS)
4(SG)	7(SG)	7(SG)
7(DCD)	6(DSR), 8(DCD)	20(DTR)
1(DTR)	20(DTR)	6(DSR), 8(DCD)

4. **Check that any hardware switches for the ports are set properly.**

   See the printer documentation for the correct settings.

5. **Check that the printer is operational.**

   Use the printer's self-test feature, if the printer has one. Check the printer documentation for information about printer self-testing.

6. **Check that the baud settings for the computer and the printer are correct.**

   If the baud settings are not the same for both the computer and the printer, sometimes nothing will print, but more often you get incorrect output. For instructions, see "How to Troubleshoot Incorrect Output" on page 675.

To check the network:

1. **Check that the network link between the print server and the print client is set up correctly.**

```
print_client# ping print_server
print_server is alive
print_server# ping print_client
print_client not available
```

   If the message says the system is alive, you know you can reach the system, so the network is all right. The message also tells you that either a name service or the local `/etc/hosts` file has translated the host (system) name you entered into an IP address; otherwise, you would need to enter the IP address.

   If you get a `not available` message, try to answer the following questions: How is NIS or NIS+ set up at your site? Do you need to take additional steps so that print servers and print clients can communicate with one another? If your site is not running NIS or NIS+, have you entered the IP address for the print server in each print client's `/etc/hosts` file, and entered all print client IP addresses in the `/etc/hosts` file of the print server?

2. **(On a SunOS 5.0–5.1 print server only) Check that the `listen` port monitor is configured correctly.**

3. **(On a SunOS 5.0–5.1 print server only) Check that the network listen services are registered with the port monitor on the print server.**

To check the basic functions of the LP print service:

This procedure uses the printer `luna` as an example of checking basic LP print service functions.

1. **On both the print server and print client, make sure that the LP print service is running.**
   a. **Check whether the LP scheduler is running.**

```
# lpstat -r
scheduler is running
```

   b. **If the scheduler is not running, become superuser or `lp`, and start the scheduler.**

```
# /usr/lib/lp/lpsched
```

   If you have trouble starting the scheduler, see "How to Unhang the LP Print Service" on page 681.

2. **On both the print server and print client, make sure that the printer is accepting requests.**
   a. **Check that the printer is accepting requests.**

```
# lpstat -a
mars accepting requests since Jul 12 14:23 1999
luna not accepting requests since Jul 12 14:23 1999
unknown reason
```

   This command verifies that the LP system is accepting requests for each printer configured for the system.

   b. **If the printer is not accepting requests, become superuser or `lp`, and allow the printer to accept print requests.**

```
# accept luna
```

   The specified printer now accepts requests.

3. **On both the print server and print client, make sure that the printer is enabled to print submitted print requests.**

   a. **Check that the printer is enabled.**

```
# lpstat -p luna
printer luna disabled since Jul 12 14:25 1999.
available.
unknown reason
```

This command displays information about printer status. You can omit the printer name to obtain information about all printers set up for the system. The following example shows a printer that is disabled.

   b. **If the printer is disabled, become superuser or** `lp`**, and enable the printer.**

```
# enable luna
printer "luna" now enabled.
```

The specified printer is enabled to process print requests.

4. **On the print server, make sure that the printer is connected to the correct serial port.**

   a. **Check that the printer is connected to the correct serial port.**

```
# lpstat -t
scheduler is running
system default destination: luna
device for luna: /dev/term/a
```

The message `device for` *printer-name* shows the port address. Is the cable connected to the port to which the LP print service says is connected? If the port is correct, skip to Step 5 on page 666.

   b. **Become superuser or** `lp`**.**

   c. **Change the file ownership of the device file that represents the port.**

```
# chown lp device-filename
```

This command assigns the special user `lp` as the owner of the device file. In this command, *device-filename* is the name of the device file.

d. Change the permissions on the printer port device file.

```
# chmod 600 device-filename
```

This command allows only superuser or `lp` to access the printer port device file.

5. On both the print server and print client, make sure that the printer is configured properly.

a. Check that the printer is configured properly.

```
# lpstat -p luna -l
printer luna is idle. enabled since Jul 12 14:24 1999. available
        Content types: postscript
        Printer types: PS
```

The above example shows a PostScript printer that is configured properly, and that is available to process print requests. If the printer type and file content type are correct, skip to Step 6 on page 666.

b. If the printer type or file content type is incorrect, try setting the print type to `unknown` and the content type to `any` on the print client.

```
# lpadmin -p printer-name -T printer-type -I file-content-type
```

6. On the print server, make sure that the printer is not faulted.

a. Check that the printer is not waiting because of a printer fault.

```
# lpadmin -p printer-name -F continue
```

This command instructs the LP print service to continue if it is waiting because of a fault.

b. Force an immediate retry by re-enabling the printer.

```
# enable printer-name
```

c. (Optional) Instruct the LP print service to enable quick notification of printer faults.

```
# lpadmin -p printer-name -A 'write root'
```

This command instructs the LP print service to set a default policy of writing root—sending the printer fault message to the terminal on which root is logged in—if the printer fails. This may help you get quick notification of faults as you try to fix the problem.

7. Make sure that the printer is not set up incorrectly as a login terminal.

Note - It is easy to mistakenly set up a printer as a login terminal, so be sure to check this possibility even if you think it does not apply.

---

a. Look for the printer port entry in the `ps -ef` command output.

```
# ps -ef
    root   169   167  0   Apr 04 ?        0:08 /usr/lib/saf/listen tcp
    root   939     1  0 19:30:47 ?        0:02 /usr/lib/lpsched
    root   859   858  0 19:18:54 term/a   0:01 /bin/sh -c \ /etc/lp
/interfaces/luna
luna-294 rocket!smith ‘‘passwd\n##
#
```

In the output from this command, look for the printer port entry. In the above example, port `/dev/term/a` is set up incorrectly as a login terminal. You can tell by the `"passwd\n##` information at the end of the line. If the port is set correctly, skip the last steps in this procedure.

b. Cancel the print request(s).

```
# cancel request-id
```

In this command, *request-id* is the request ID number for a print request to be canceled.

c. Set the printer port to be a nonlogin device.

```
# lpadmin -p printer-name -h
```

d. Check the `ps -ef` command output to verify that the printer port is no longer a login device.

If you do not find the source of the printing problem in the basic LP print service functions, continue to one of the following procedures for the specific client/server case that applies.

To check printing from a SunOS 5.8 print client to a SunOS 5.8 print server:

1. Check the basic functions of the LP print service on the print server, if you have not done so already.

For instructions on checking basic functions, see "To check the basic functions of the LP print service: " on page 664. Make sure that the printer works locally before trying to figure out why nothing prints when a request is made from a print client.

2. Check the basic functions of the LP print service on the print client, if you have not done so already.

For instructions on checking basic functions, see "To check the basic functions of the LP print service: " on page 664. On the print client, the LP scheduler has to be running, and the printer has to be enabled and accepting requests before any request from the client will print.

Note - For most of the following steps, you must be logged in as root or `lp`.

---

3. Make sure that the print server is accessible.

a. On the print client, send an "are you there?" request to the print server.

```
print_client# ping print_server
```

If you receive the message *print_server* `not available`, you may have a network problem.

4. On SunOS 5.1 print client only, make sure that the print server is identified as type `s5` by viewing the Modify Printer window in Admintool.

5. Verify that the print server is operating properly.

```
# lpstat -t luna
scheduler is running
system default destination: luna
device for luna: /dev/term/a
luna accepting requests since Jul 12 14:26 1999
printer luna now printing luna-314. enabled since Jul 12 14:26 1999.
available.
luna-129            root               488   Jul 12 14:32
#
```

The above example shows a print server up and running.

6. **If the print server is not operating properly, go back to step 1.**

To check printing from a SunOS 5.8 print client to a SunOS 4.1 print server:

1. **Check the basic functions of the LP print service on the print client, if you have not done so already.**

   For instructions, see “To check the basic functions of the LP print service: ” on page 664.

2. **Make sure that the print server is accessible.**

   a. **On the print client, send an “are you there?” request to the print server.**

   print_client# **ping** *print_server*

   If you receive the message *print_server* `not available`, you may have a network problem.

3. **Make sure that the** `lpd` **daemon on the print server is running.**

   a. **On the print server, verify the** `lpd` **daemon is running.**

```
$ ps -ax | grep lpd
  126 ?  IW    0:00 /usr/lib/lpd
  200 p1 S     0:00 grep lpd
$
```

If the `lpd` daemon is running, a line is displayed, as shown in the above example. If it is not running, no process information is shown.

b. If `lpd` is not running on the print server, become superuser on the print server, and restart it.

```
# /usr/lib/lpd &
```

4. Make sure that the remote `lpd` daemon is configured properly.

a. On the print server, become superuser, and invoke the `lpc` command.

```
# /usr/ucb/lpc
lpc>
```

b. Get LP status information.

```
lpc> status
luna:
queuing is enabled
printing is enabled
no entries
no daemon present
lpc>
```

Status information is displayed. In the above example, the daemon is not running and needs to be restarted.

c. If no daemon is present, restart the daemon.

```
lpc> restart luna
```

The daemon is restarted.

d. Verify that the `lpd` daemon has started.

```
lpc> status
```

e. Quit the `lpc` command.

```
lpc> quit
```

The shell prompt is redisplayed.

5. **Make sure that the print client has access to the print server.**

   a. **Check if there is an `/etc/hosts.lpd` file on the SunOS 4.1 print server.**

      On a SunOS 4.1 print server, if this file exists, it is used to determine whether an incoming print request can be accepted. If the file does not exist, all print client systems have access, so skip steps b and c.

   b. **If the file exists, see if the print client is listed in the file.**

      Requests from client systems not listed in the file are not transferred to the print server.

   c. **If the client is not listed, add the print client to the file.**

Note - If you get this far without pinpointing the problem, the SunOS 4.1 system is probably set up and working properly.

---

6. **Make sure that the connection to the remote `lpd` print daemon from the print client is made correctly.**

   a. **On the print client, become superuser, and verify the `lpsched` daemon is running.**

```
# ps -ef | grep lp
   root   154     1 80   Jan 07 ?        0:02 /usr/lib/lpsched
```

      The `lpsched` daemon should be running, as shown in the above example.

   b. **Stop the LP print service.**

```
# lpshut
```

      The LP print service is stopped.

   c. **Restart the LP print service.**

```
# /usr/lib/lp/lpsched
```

      The LP print service is restarted.

7. **Make sure that the remote print server is identified correctly as a SunOS 4.1 system.**

To check printing from a SunOS 4.1 client to a SunOS 5.8 print server:

1. **Check the basic functions of the LP print service on the print server, if you have not done so already.**

   For instructions, see “To check the basic functions of the LP print service: ” on page 664. Make sure that the printer works locally before trying to figure out why nothing prints when a request is made from a print client.

   Note - You should be logged in as superuser or `lp` on the system specified in the following steps.

   ---

2. **Make sure that the print client is accessible.**

   a. **On the SunOS 5.8 print server, send an “are you there?” request to the print client.**

```
print_server# ping print_client
print_client is alive
```

   If you receive the message *print_client* `not available`, you may have a network problem.

3. **On the print client, verify the printer is set up correctly.**

```
# lpr -P luna /etc/fstab
lpr: cannot access luna
#
```

   This command shows whether the print client is working. The above example shows that the print client is not working correctly.

4. **Make sure that the `lpd` daemon is running on the print client.**

   a. **Verify the `lpd` daemon is running.**

```
# ps -ax | grep lpd
  118 ?  IW    0:02 /usr/lib/lpd
#
```

This command shows if the `lpd` daemon is running on the print client. The above example shows that the daemon is running.

b. On the print client, start the `lpd` daemon.

```
# /usr/lib/lpd &
```

5. On the print client, make sure that there is a `printcap` entry identifying the printer.

a. Verify the printer is known.

```
# lpr -P mercury /etc/fstab
lpr: mercury: unknown printer
#
```

The above example shows that the `/etc/printcap` file does not have an entry for the specified printer.

b. If there is no entry, edit the `/etc/printcap` file and add the following information:

```
printer-name|print-server:\
:lp=:rm=print-server:rp=printer-name:br#9600:rw:\
:lf=/var/spool/lpd/printer-name/log:\
:sd=/var/spool/lpd/printer-name:
```

The following example shows an entry for printer `luna` connected to print server `neptune`.

```
luna|neptune:\
        :lp=:rm=neptune:rp=luna:br#9600:rw:\
        :lf=/var/spool/lpd/luna/log:\
        :sd=/var/spool/lpd/luna:
```

c. Create a spooling directory (`/var/spool/lpd/`*printer-name*) for the printer.

6. Make sure that the print client `lpd` is not in a wait state by forcing a retry.

If the print server is up and responding, the print client `lpd` may be in a wait state before attempting a retry.

a. As superuser on the print client, invoke the `lpc` command.

The `lpc>` prompt is displayed.

b. Restart the printer.

c. Quit the `lpc` command.

The shell prompt is redisplayed.

```
# lpc
lpc> restart luna
luna:
       no daemon to abort
luna:
      daemon started
# quit
$
```

7. Check the connection to the print server.

a. On the print client, become superuser, and check the printer log file.

```
# more /var/spool/lpd/luna/log
```

Frequently, no information is displayed.

b. Also check the printer status log.

```
# more /var/spool/lpd/luna/status
waiting for luna to come up
#
```

c. If the connection is all right, on the print server, verify the print server is set up correctly.

```
# lpstat -t
scheduler is running
system default destination: luna
device for luna: /dev/term/a
luna accepting requests since Jul 12 14:29 1999
luna accepting requests since Jul 12 14:29 1999
printer luna is idle. enabled since Jul 12 14:29 1999. available.
#
```

The above example shows a print server that is up and running.

If the print server is not running, go back to Step 1 on page 672 before continuing.

▼ How to Troubleshoot Incorrect Output

1. Log in as superuser or `lp`.

2. Make sure that the printer type is correct.

An incorrect printer type may cause incorrect output. For example, if you specify printer type PS and the pages print in reverse order, try printer type PSR. (These type names must be in uppercase.) Also, an incorrect printer type may cause missing text, illegible text, or text with the wrong font. To determine the printer type, examine the entries in the `terminfo` database. For information on the structure of the `terminfo` database, see “Printer Type” on page 57.

a. On the print server, display the printer’s characteristics.

```
$ lpstat -p luna -l
printer luna is idle. enabled since Mon Jul 12 15:02:32 MDT 1999. available.
        Form mounted:
        Content types: postscript
        Printer types: PS
        Description:
        Connection: direct
        Interface: /usr/lib/lp/model/standard
        After fault: continue
        Users allowed:
                (all)
        Forms allowed:
                (none)
        Banner not required
        Character sets:

        Default pitch:
        Default page size: 80 wide 66 long
        Default port settings:
$
```

b. Consult the printer manufacturer's documentation to determine the printer model.

c. If the printer type is not correct, change it with Admintool's Modify Printer option, or use the following `lpadmin` command.

```
# lpstat -p printer-name -T printer-type
```

On the print client, the printer type should be `unknown`. On the print server, the printer type must match a `terminfo` entry that is defined to support the model of printer you have. If there is no `terminfo` entry for the type of printer you have, see "How to Add a `terminfo` Entry for an Unsupported Printer" on page 157.

3. If the banner page prints, but there is no output for the body of the document, check the file content types.

File content types specified for a printer indicate the types of files the printer can print directly without filtering. An incorrect file content type causes filtering to be bypassed when it may be needed.

a. Note the information on file content type that was supplied in the previous step by the `lpstat` command.

On the print client, the file content type should be `any`, unless you have good reason to specify one or more explicit content types. If a content is specified on the client, filtering is done on the print client, rather than the print server.

In addition, content types on the client must match the content types specified on the print server, which in turn must reflect the capabilities of the printer.

b. Consult your printer manufacturer's documentation to determine which types of files the printer can print directly.

The names you use to refer to these types of files do not have to match the names used by the manufacturer. However, the names you use must agree with the names used by the filters known to the LP print service.

c. If the file content type is not correct, change it with Admintool's Modify Printer option, or the following `lpadmin` command.

```
# lpadmin -p printer-name -I file-content-type(s)
```

Run this command on either the print client, or print server, or both, as needed. Try `–I any` on the print client, and `–I ""` on the print server. The latter specifies a null file content type list, which means an attempt should be made to filter all files, because the printer can directly print only files that exactly match its printer type.

This combination is a good first choice when files are not printing. If it works, you may want to try specifying explicit content types on the print server to reduce unnecessary filtering. For a local PostScript printer, you should use `postscript`, or `postscript,simple`— if the printer supports these types. Be aware that `PS` and `PSR` are not file content types; they are printer types.

If you omit `–I`, the file content list defaults to `simple`. If you use the `–I` option and want to specify file content types in addition to `simple`, `simple` must be included in the list.

When specifying multiple file content types, separate the names with commas. Or you can separate names with spaces and enclose the list in quotation marks. If you specify `any` as the file content type, no filtering will be done and only file types that can be printed directly by the printer should be sent to it.

4. Check that the print request does not bypass filtering needed to download fonts.

If a user submits a print request to a PostScript printer with the command `lp -T PS`, no filtering is done. Try submitting the request with the command `lp -T postscript` to force filtering, which may result in the downloading of non-resident fonts needed by the document.

5. Make sure that the `stty` settings for the printer port are correct.

a. Read the printer documentation to determine the correct `stty` settings for the printer port.

Note - If a printer is connected by a parallel port, the baud setting is irrelevant.

b. Examine the current settings by using the `stty` command.

```
# stty -a < /dev/term/a
speed 9600 baud;
rows = 0; columns = 0; ypixels = 0; xpixels = 0;
eucw 1:0:0:0, scrw 1:0:0:0
intr = ^c; quit = ^|; erase = ^?; kill = ^u;
eof = ^d; eol = <undef>; eol2 = <undef>; swtch = <undef>;
start = ^q; stop = ^s; susp = ^z; dsusp = ^y;
rprnt = ^r; flush = ^o; werase = ^w; lnext = ^v;
parenb -parodd cs7 -cstopb -hupcl cread -clocal -loblk -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 -defecho -flusho -pendin iexten
opost -olcuc onlcr -ocrnl -onocr -onlret -ofill -ofdel tab3
#
```

This command shows the current `stty` settings for the printer port.

The table below shows the default `stty` options used by the LP print service's standard printer interface program.

TABLE 42–2 Default `stty` Settings Used by the Standard Interface Program

Option	Meaning
`–9600`	Set baud rate to 9600
`–cs8`	Set 8-bit bytes
`–cstopb`	Send one stop bit per byte
`–parity`	Do not generate parity
`–ixon`	Enable XON/XOFF (also known as START/STOP or DC1/DC3)
`–opost`	Do "output post-processing" using all the settings that follow in this table

TABLE 42–2 Default `stty` Settings Used by the Standard Interface Program *(continued)*

Option	Meaning
`–olcuc`	Do not map lowercase to uppercase
`–onlcr`	Change line feed to carriage return/line feed
`–ocrnl`	Do not change carriage returns into line feeds
`–onocr`	Output carriage returns even at column 0
`–n10`	No delay after line feeds
`–cr0`	No delay after carriage returns
`–tab0`	No delay after tabs
`–bs0`	No delay after backspaces
`–vt0`	No delay after vertical tabs
`–ff0`	No delay after form feeds

c. Change the `stty` settings.

```
# lpadmin -p printer-name -o "stty= options"
```

Use the table below to choose `stty` options to correct various problems affecting print output.

TABLE 42–3 `stty` Options to Correct Print Output Problems

`stty` **Values**	**Result**	**Possible Problem From Incorrect Setting**
`110`, `300`, `600`, `1200`, `1800`, `2400`, `4800`, `9600`, `19200`, `38400`	Sets baud rate to the specified value (enter only one baud rate)	Random characters and special characters may be printed and spacing may be inconsistent
`oddp`	Sets odd parity	Missing or incorrect characters appear randomly
`evenp`	Sets even parity	
`-parity`	Sets no parity	
`-tabs`	Sets no tabs	Text is jammed against right margin
`tabs`	Sets tabs every eight spaces	Text has no left margin, is run together, or is jammed together
`-onlcr`	Sets no carriage return at the beginning of line(s)	Incorrect double spacing
`onlcr`	Sets carriage return at beginning of line(s)	The print zigzags down the page

You can change more than one option setting by enclosing the list of options in single quotation marks and separating each option with spaces. For example, suppose the printer requires you to enable odd parity and set a 7-bit character size. You would type a command similar to that shown in the following example:

```
# lpadmin -p neptune -o "stty='parenb parodd cs7'"
```

The `stty` option `parenb` enables parity checking/generation, `parodd` sets odd parity generation, and `cs7` sets the character size to 7 bits.

6. Verify that the document prints correctly.

```
# lp -d printer-name filename
```

▼ How to Unhang the LP Print Service

1. **Log in as superuser or lp.**

2. **Stop the LP print service.**

```
# lpshut
```

   If this command hangs, press Control-c and proceed to the next step. If this command succeeds, skip to step 4.

3. **Identify the LP process IDs.**

```
# ps -el | grep lp
   134 term/a   0:01 lpsched
#
```

   Use the process ID numbers (PIDs) from the first column in place of the *pid* variables in the next step.

4. **Stop the LP processes by using the `kill -15` command.**

```
# kill -15 103 134
```

   This should stop the LP print service processes. If the processes do not stop, as a last resort go to step 5.

5. **As a last resort, terminate the processes abruptly.**

```
# kill -9 103 134
```

   All the `lp` processes are terminated.

6. **Remove the `SCHEDLOCK` file so you can restart the LP print service.**

```
# rm /usr/spool/lp/SCHEDLOCK
```

7. **Restart the LP print service.**

```
# /usr/lib/lp/lpsched
```

The LP print service should restart. If you are having trouble restarting the scheduler, see “How to Restart the Print Scheduler” on page 94.

▼ How to Troubleshoot an Idle (Hung) Printer

This task includes a number of procedures to use when a printer appears idle but it should not be. It makes sense to try the procedures in order, but the order is not mandatory.

To check that the printer is ready to print:

1. **Display printer status information.**

```
# lpstat -p printer-name
```

The information displayed shows you whether the printer is idle or active, enabled or disabled, or available or not accepting print requests. If everything looks all right, continue with other procedures in this section. If you cannot run the `lpstat` command, see “How to Unhang the LP Print Service” on page 681.

2. **If the printer is not available (not accepting requests), allow the printer to accept requests.**

```
# accept printer-name
```

The printer begins to accept requests into its print queue.

3. **If the printer is disabled, re-enable it.**

```
# enable printer-name
```

This command re-enables the printer so that it will act on the requests in its queue.

To check for print filtering:

Check for print filtering by using the `lpstat -o` command.

```
$ lpstat -o luna
luna-10           fred         1261   Mar 12 17:34 being filtered
luna-11           iggy         1261   Mar 12 17:36 on terra
luna-12           jack         1261   Mar 12 17:39 on terra
```

(continued)

```
$
```

See if the first waiting request is being filtered. If the output looks like the above example, the file is being filtered; the printer is not hung, it just is taking a while to process the request.

To resume printing after a printer fault:

1. **Look for a message about a printer fault and try to correct the fault if there is one.**

   Depending on how printer fault alerts have been specified, messages may be sent to root by email or written to a terminal on which root is logged in.

2. **Re-enable the printer.**

   ```
   # enable printer-name
   ```

   If a request was blocked by a printer fault, this command will force a retry. If this command does not work, continue with other procedures in this section.

To send print requests to a remote printer when they back up in the local queue:

1. **On the print client, stop further queuing of print requests to the print server.**

   ```
   # reject printer-name
   ```

2. **On the print client, send an “are you there?” request to the print server.**

   ```
   print_client# ping print_server
   print_server is alive
   ```

   If you receive the message *print_server* `not available`, you may have a network problem.

3. **After you fix the above problem, allow new print requests to be queued.**

```
# accept printer-name
```

4. **If necessary, re-enable the printer.**

```
# enable printer-name
```

To free print requests from a print client that back up in the print server queue:

1. **On the print server, stop further queuing of print requests from any print client to the print server.**

```
# reject printer-name
```

2. **Display the** `lpsched` **log file.**

```
# more /var/lp/logs/lpsched
```

The information displayed may help you pinpoint what is preventing the print requests from the print client to the print server from being printed.

3. **After you fix the problem, allow new print requests to be queued.**

```
# accept printer-name
```

4. **If necessary, re-enable the printer on the print server.**

```
# enable printer-name
```

▼ How to Resolve Conflicting Printer Status Messages

1. **On the print server, verify the printer is enabled and is accepting requests.**

```
# lpstat -p printer-name
```

Users will see conflicting status messages when the print client is accepting requests, but the print server is rejecting requests.

2. **On the print server, check that the definition of the printer on the print client matches the definition of the printer on the print server.**

```
# lpstat -p -l printer-name
```

Look at the definitions of the print job components, like print filters, character sets, print wheels, and forms, to be sure they are the same on both the client and server systems so that local users can access printers on print server systems.

CHAPTER 43

Troubleshooting File System Problems

This is a list of the information in this chapter.

- "General `fsck` Error Messages " on page 689
- "Initialization Phase `fsck` Messages" on page 690
- "Phase 1: Check Blocks and Sizes Messages" on page 694
- "Phase 1B: Rescan for More DUPS Messages" on page 698
- "Phase 2: Check Path Names Messages" on page 698
- "Phase 3: Check Connectivity Messages" on page 707
- "Phase 4: Check Reference Counts Messages" on page 709
- "Phase 5: Check Cylinder Groups Messages" on page 713
- "Cleanup Phase Messages" on page 714

See "Checking File System Integrity" in *System Administration Guide, Volume 1* for information about the `fsck` program and how to use it to check file system integrity.

`fsck` Error Messages

Normally, `fsck` is run non-interactively to *preen* the file systems after an abrupt system halt in which the latest file system changes were not written to disk. Preening automatically fixes any basic file system inconsistencies and does not try to repair more serious errors. While preening a file system, `fsck` fixes the inconsistencies it expects from such an abrupt halt. For more serious conditions, the command reports the error and terminates.

When you run `fsck` interactively, `fsck` reports each inconsistency found and fixes innocuous errors. However, for more serious errors, the command reports the

inconsistency and prompts you to choose a response. When you run `fsck` using the `-y` or `-n` options, your response is predefined as yes or no to the default response suggested by `fsck` for each error condition.

Some corrective actions will result in some loss of data. The amount and severity of data loss may be determined from the `fsck` diagnostic output.

`fsck` is a multipass file system check program. Each pass invokes a different phase of the `fsck` program with different sets of messages. After initialization, `fsck` performs successive passes over each file system, checking blocks and sizes, path names, connectivity, reference counts, and the map of free blocks (possibly rebuilding it). It also performs some cleanup.

The phases (passes) performed by the UFS version of `fsck` are:

- Initialization
- Phase 1 – Check blocks and sizes
- Phase 2 – Check path names
- Phase 3 – Check connectivity
- Phase 4 – Check reference counts
- Phase 5 – Check cylinder groups

The next sections describe the error conditions that may be detected in each phase, the messages and prompts that result, and possible responses you can make.

Messages that may appear in more than one phase are described in “General `fsck` Error Messages ” on page 689. Otherwise, messages are organized alphabetically by the phases in which they occur.

Many of the messages include the abbreviations shown in the table below:

TABLE 43–1 Error Message Abbreviations

Abbreviation	Meaning
`BLK`	Block number
`DUP`	Duplicate block number
`DIR`	Directory name
`CG`	Cylinder group

TABLE 43–1 Error Message Abbreviations *(continued)*

Abbreviation	Meaning
MTIME	Time file was last modified
UNREF	Unreferenced

Many of the messages also include variable fields, such as inode numbers, which are represented in this book by an italicized term, such as *inode-number*. For example, this screen message:

```
INCORRECT BLOCK COUNT I=2529
```

is shown as:

```
INCORRECT BLOCK COUNT I=inode-number
```

General `fsck` Error Messages

The error messages in this section may be displayed in any phase after initialization. Although they offer the option to continue, it is generally best to regard them as fatal. They reflect a serious system failure and should be handled immediately. When confronted with such a message, terminate the program by entering n(o). If you cannot determine what caused the problem, contact your local service provider or another qualified person.

```
CANNOT SEEK: BLK block-number (CONTINUE)
```

Cause

A request to move to a specified block number, *block-number*, in the file system failed. This message indicates a serious problem, probably a hardware failure.

If you want to continue the file system check, `fsck` will retry the move and display a list of sector numbers that could not be moved. If the block was part of the virtual memory buffer cache, `fsck` will terminate with a fatal I/O error message.

Action

If the disk is experiencing hardware problems, the problem will persist. Run `fsck` again to recheck the file system.

If the recheck fails, contact your local service provider or another qualified person.

```
CANNOT READ: BLK block-number (CONTINUE)
```

Cause

A request to read a specified block number, *block-number*, in the file system failed. The message indicates a serious problem, probably a hardware failure.

If you want to continue the file system check, `fsck` will retry the read and display a list of sector numbers that could not be read. If the block was part of the virtual memory buffer cache, `fsck` will terminate with a fatal I/O error message. If `fsck` tries to write back one of the blocks on which the read failed, it will display the following message:

`WRITING ZERO'ED BLOCK` *sector-numbers* `TO DISK`

Action

If the disk is experiencing hardware problems, the problem will persist. Run `fsck` again to recheck the file system. If the recheck fails, contact your local service provider or another qualified person.

`CANNOT WRITE: BLK` *block-number* `(CONTINUE)`

Cause

A request to write a specified block number, *block-number*, in the file system failed.

If you continue the file system check, `fsck` will retry the write and display a list of sector numbers that could not be written. If the block was part of the virtual memory buffer cache, `fsck` will terminate with a fatal I/O error message.

Action

The disk may be write-protected. Check the write-protect lock on the drive. If the disk has hardware problems, the problem will persist. Run `fsck` again to recheck the file system. If the write-protect is not the problem or the recheck fails, contact your local service provider or another qualified person.

Initialization Phase `fsck` Messages

In the initialization phase, command-line syntax is checked. Before the file system check can be performed, `fsck` sets up tables and opens files.

The messages in this section relate to error conditions resulting from command-line options, memory requests, the opening of files, the status of files, file system size checks, and the creation of the scratch file. All such initialization errors terminate `fsck` when it is preening the file system.

`bad inode number` *inode-number* `to ginode`

Cause

An internal error occurred because of a nonexistent inode *inode-number*. `fsck` exits.

Action

Contact your local service provider or another qualified person.

```
cannot alloc size-of-block map bytes for blockmap
cannot alloc size-of-free map bytes for freemap
cannot alloc size-of-state map bytes for statemap
cannot alloc size-of-lncntp bytes for lncntp
```

Cause

Request for memory for its internal tables failed. `fsck` terminates. This message indicates a serious system failure that should be handled immediately. This condition may occur if other processes are using a very large amount of system resources.

Action

Killing other processes may solve the problem. If not, contact your local service provider or another qualified person.

```
Can't open checklist file: filename
```

Cause

The file system checklist file *filename* (usually `/etc/vfstab`) cannot be opened for reading. `fsck` terminates.

Action

Check if the file exists and if its access modes permit read access.

```
Can't open filename
```

Cause

`fsck` cannot open file system *filename*. When running interactively, `fsck` ignores this file system and continues checking the next file system given.

Action

Check to see if read and write access to the raw device file for the file system is permitted.

```
Can't stat root
```

Cause

`fsck` request for statistics about the root directory failed. `fsck` terminates.

Action

This message indicates a serious system failure. Contact your local service provider or another qualified person.

```
Can't stat filename
Can't make sense out of name filename
```

Cause

`fsck` request for statistics about the file system *filename* failed. When running interactively, `fsck` ignores this file system and continues checking the next file system given.

Action

Check if the file system exists and check its access modes.

```
filename: (NO WRITE)
```

Cause

Either the `-n` option was specified or `fsck` could not open the file system *filename* for writing. When `fsck` is running in no-write mode, all diagnostic messages are displayed, but `fsck` does not attempt to fix anything.

Action

If `-n` was not specified, check the type of the file specified. It may be the name of a regular file.

```
IMPOSSIBLE MINFREE=percent IN SUPERBLOCK (SET TO DEFAULT)
```

Cause

The superblock minimum space percentage is greater than 99 percent or less than 0 percent.

Action

To set the `minfree` parameter to the default 10 percent, type `y` at the default prompt. To ignore the error condition, type `n` at the default prompt.

```
filename: BAD SUPER BLOCK: message
USE AN ALTERNATE SUPER-BLOCK TO SUPPLY NEEDED INFORMATION;
e.g., fsck[-f ufs] -o b=# [special ...]
where # is the alternate superblock.  See fsck_ufs(1M)
```

Cause

The superblock has been corrupted.

Action

One of the following messages may be displayed:

```
CPG OUT OF RANGE
FRAGS PER BLOCK OR FRAGSIZE WRONG
INODES PER GROUP OUT OF RANGE
INOPB NONSENSICAL RELATIVE TO BSIZE
MAGIC NUMBER WRONG
NCG OUT OF RANGE
NCYL IS INCONSISTENT WITH NCG*CPG
NUMBER OF DATA BLOCKS OUT OF RANGE
NUMBER OF DIRECTORIES OUT OF RANGE
ROTATIONAL POSITION TABLE SIZE OUT OF RANGE
SIZE OF CYLINDER GROUP SUMMARY AREA WRONG
SIZE TOO LARGE
BAD VALUES IN SUPERBLOCK
```

Try to rerun `fsck` with an alternative superblock. Specifying block 32 is a good first choice. You can locate an alternative copy of the superblock by running the `newfs -N` command on the slice. Be sure to specify the `-N` option; otherwise, `newfs` overwrites the existing file system.

```
UNDEFINED OPTIMIZATION IN SUPERBLOCK (SET TO DEFAULT)
```

Cause

The superblock optimization parameter is neither `OPT_TIME` nor `OPT_SPACE`.

Action

To minimize the time to perform operations on the file system, type `y` at the `SET TO DEFAULT` prompt. To ignore this error condition, type `n`.

Phase 1: Check Blocks and Sizes Messages

This phase checks the inode list. It reports error conditions encountered while:

- Checking inode types
- Setting up the zero-link-count table
- Examining inode block numbers for bad or duplicate blocks
- Checking inode size
- Checking inode format

All errors in this phase except `INCORRECT BLOCK COUNT`, `PARTIALLY TRUNCATED INODE`, `PARTIALLY ALLOCATED INODE`, and `UNKNOWN FILE TYPE` terminate `fsck` when it is preening a file system.

These messages (in alphabetical order) may occur in phase 1:

block-number BAD I=*inode-number*

Cause

Inode *inode-number* contains a block number *block-number* with a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system. This error condition may generate the `EXCESSIVE BAD BLKS` error message in phase 1 if inode *inode-number* has too many block numbers outside the file system range. This error condition generates the `BAD/DUP` error message in phases 2 and 4.

Action

N / A

BAD MODE: MAKE IT A FILE?

Cause

The status of a given inode is set to all 1s, indicating file system damage. This message does not indicate physical disk damage, unless it is displayed repeatedly after `fsck -y` has been run.

Action

Type `y` to reinitialize the inode to a reasonable value.

BAD STATE *state-number* TO BLKERR

Cause

An internal error has scrambled the `fsck` state map so that it shows the impossible value *state-number*. `fsck` exits immediately.

Action

Contact your local service provider or another qualified person.

block-number `DUP I=`*inode-number*

Cause

Inode *inode-number* contains a block number *block-number*, which is already claimed by the same or another inode. This error condition may generate the `EXCESSIVE DUP BLKS` error message in phase 1 if inode *inode-number* has too many block numbers claimed by the same or another inode. This error condition invokes phase 1B and generates the `BAD/DUP` error messages in phases 2 and 4.

Action

N / A

```
DUP TABLE OVERFLOW (CONTINUE)
```

Cause

There is no more room in an internal table in `fsck` containing duplicate block numbers. If the `-o p` option is specified, the program terminates.

Action

To continue the program, type `y` at the `CONTINUE` prompt. When this error occurs, a complete check of the file system is not possible. If another duplicate block is found, this error condition repeats. Increase the amount of virtual memory available (by killing some processes, increasing swap space) and run `fsck` again to recheck the file system. To terminate the program, type `n`.

```
EXCESSIVE BAD BLOCKS I=inode-number (CONTINUE)
```

Cause

Too many (usually more than 10) blocks have a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system associated with inode *inode-number*. If the `-o p` (preen) option is specified, the program terminates.

Action

To continue the program, type `y` at the `CONTINUE` prompt. When this error occurs, a complete check of the file system is not possible. You should run `fsck` again to recheck the file system. To terminate the program, type `n`.

```
EXCESSIVE DUP BLKS I=inode-number (CONTINUE)
```

Cause

Too many (usually more than 10) blocks are claimed by the same or another inode or by a free-list. If the `-o p` option is specified, the program terminates.

Action

To continue the program, type `y` at the `CONTINUE` prompt. When this error occurs, a complete check of the file system is not possible. You should run `fsck` again to recheck the file system. To terminate the program, type `n`.

```
INCORRECT BLOCK COUNT I=inode-number (number-of-BAD-DUP-or-missing-blocks
should be
 number-of-blocks-in-filesystem)
(CORRECT)
```

Cause

The block count for inode *inode-number* is *number-of-BAD-DUP-or-missing-blocks*, but should be *number-of-blocks-in-filesystem*. When preening, `fsck` corrects the count.

Action

To replace the block count of inode *inode-number* by *number-of-blocks-in-filesystem*, type `y` at the `CORRECT` prompt. To terminate the program, type `n`.

```
LINK COUNT TABLE OVERFLOW (CONTINUE)
```

Cause

There is no more room in an internal table for `fsck` containing allocated inodes with a link count of zero. If the `-o p` (preen) option is specified, the program exits and `fsck` has to be completed manually.

Action

To continue the program, type `y` at the `CONTINUE` prompt. If another allocated inode with a zero-link count is found, this error condition repeats. When this error occurs, a complete check of the file system is not possible. You should run `fsck` again to

recheck the file system. Increase the virtual memory available by killing some processes or increasing swap space, then run `fsck` again. To terminate the program, type `n`.

```
PARTIALLY ALLOCATED INODE I=inode-number (CLEAR)
```

Cause

Inode *inode-number* is neither allocated nor unallocated. If the `-o p` (preen) option is specified, the inode is cleared.

Action

To deallocate the inode *inode-number* by zeroing out its contents, type `y`. This may generate the `UNALLOCATED` error condition in phase 2 for each directory entry pointing to this inode. To ignore the error condition, type `n`. A no response is appropriate only if you intend to take other measures to fix the problem.

```
PARTIALLY TRUNCATED INODE I=inode-number (SALVAGE)
```

Cause

`fsck` has found inode *inode-number* whose size is shorter than the number of blocks allocated to it. This condition occurs only if the system crashes while truncating a file. When preening the file system, `fsck` completes the truncation to the specified size.

Action

To complete the truncation to the size specified in the inode, type `y` at the `SALVAGE` prompt. To ignore this error condition, type `n`.

```
UNKNOWN FILE TYPE I=inode-number (CLEAR)
```

Cause

The mode word of the inode *inode-number* shows that the inode is not a pipe, special character inode, special block inode, regular inode, symbolic link, FIFO file, or directory inode. If the `-o p` option is specified, the inode is cleared.

Action

To deallocate the inode *inode-number* by zeroing its contents, which results in the `UNALLOCATED` error condition in phase 2 for each directory entry pointing to this inode, type `y` at the `CLEAR` prompt. To ignore this error condition, type `n`.

Phase 1B: Rescan for More DUPS Messages

When a duplicate block is found in the file system, this message is displayed:

```
block-number DUP I=inode-number
```

Cause

Inode *inode-number* contains a block number *block-number* that is already claimed by the same or another inode. This error condition generates the BAD/DUP error message in phase 2. Inodes that have overlapping blocks may be determined by examining this error condition and the DUP error condition in phase 1.

Action

When a duplicate block is found, the file system is rescanned to find the inode that previously claimed that block.

Phase 2: Check Path Names Messages

This phase removes directory entries pointing to bad inodes found in phases 1 and 1B. It reports error conditions resulting from:

- Incorrect root inode mode and status
- Directory inode pointers out of range
- Directory entries pointing to bad inodes
- Directory integrity checks

When the file system is being preened (–o p option), all errors in this phase terminate fsck, except those related to directories not being a multiple of the block size, duplicate and bad blocks, inodes out of range, and extraneous hard links.

These messages (in alphabetical order) may occur in phase 2:

```
BAD INODE state-number TO DESCEND
```

Cause

An fsck internal error has passed an invalid state *state-number* to the routine that descends the file system directory structure. fsck exits.

Action

If this error message is displayed, contact your local service provider or another qualified person.

```
BAD INODE NUMBER FOR '.' I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename (FIX)
```

Cause

A directory *inode-number* has been found whose inode number for "." does not equal *inode-number*.

Action

To change the inode number for "." to be equal to *inode-number*, type y at the FIX prompt To leave the inode numbers for "." unchanged, type n.

```
BAD INODE NUMBER FOR '..' I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename (FIX)
```

Cause

A directory *inode-number* has been found whose inode number for ".." does not equal the parent of *inode-number*.

Action

To change the inode number for ".." to be equal to the parent of *inode-number*, type y at the FIX prompt. (Note that ".." in the root inode points to itself.)To leave the inode number for ".." unchanged, type n.

```
BAD RETURN STATE state-number FROM DESCEND
```

Cause

An fsck internal error has returned an impossible state *state-number* from the routine that descends the file system directory structure. fsck exits.

Action

If this message is displayed, contact your local service provider or another qualified person.

```
BAD STATE state-number FOR ROOT INODE
```

Cause

An internal error has assigned an impossible state *state-number* to the root inode. `fsck` exits.

Action

If this error message is displayed, contact your local service provider or another qualified person.

```
BAD STATE state-number FOR INODE=inode-number
```

Cause

An internal error has assigned an impossible state *state-number* to inode *inode-number*. `fsck` exits.

Action

If this error message is displayed, contact your local service provider or another qualified person.

```
DIRECTORY TOO SHORT I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename (FIX)
```

Cause

A directory *filename* has been found whose size *file-size* is less than the minimum directory size. The owner *UID*, mode *file-mode*, size *file-size*, modify time *modification-time*, and directory name *filename* are displayed.

Action

To increase the size of the directory to the minimum directory size, type `y` at the `FIX` prompt. To ignore this directory, type `n`.

```
DIRECTORY filename: LENGTH file-size NOT MULTIPLE OF block-number (ADJUST)
```

Cause

A directory *filename* has been found with size *file-size* that is not a multiple of the directory block size *block-number*.

Action

To round up the length to the appropriate block size, type y. When preening the file system (-o p option), fsck only displays a warning and adjusts the directory. To ignore this condition, type n.

```
DIRECTORY CORRUPTED I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename (SALVAGE)
```

Cause

A directory with an inconsistent internal state has been found.

Action

To throw away all entries up to the next directory boundary (usually a 512-byte boundary), type y at the SALVAGE prompt. This drastic action can throw away up to 42 entries. Take this action only after other recovery efforts have failed. To skip to the next directory boundary and resume reading, but not modify the directory, type n.

```
DUP/BAD I=inode-number OWNER=O MODE=M SIZE=file-size
MTIME=modification-time TYPE=filename
(REMOVE)
```

Cause

Phase 1 or phase 1B found duplicate blocks or bad blocks associated with directory or file entry *filename*, inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, modification time *modification-time*, and directory or file name *filename* are displayed. If the -p (preen) option is specified, the duplicate/bad blocks are removed.

Action

To remove the directory or file entry *filename*, type y at the REMOVE prompt. To ignore this error condition, type n.

```
DUPS/BAD IN ROOT INODE (REALLOCATE)
```

Cause

Phase 1 or phase 1B has found duplicate blocks or bad blocks in the root inode (usually inode number 2) of the file system.

Action

To clear the existing contents of the root inode and reallocate it, type `y` at the `REALLOCATE` prompt. The files and directories usually found in the root inode will be recovered in phase 3 and put into the `lost+found` directory. If the attempt to allocate the root fails, `fsck` will exit with: `CANNOT ALLOCATE ROOT INODE`. Type `n` to get the `CONTINUE` prompt. Type: `y` to respond to the `CONTINUE` prompt, and ignore the `DUPS/BAD` error condition in the root inode and continue running the file system check. If the root inode is not correct, this may generate many other error messages. Type `n` to terminate the program.

```
EXTRA '.' ENTRY I=inode-number OWNER=UID
MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename (FIX)
```

Cause

A directory *inode-number* has been found that has more than one entry for "`.`".

Action

To remove the extra entry for "`.`" type `y` at the `FIX` prompt. To leave the directory unchanged, type `n`.

```
EXTRA '..' ENTRY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename(FIX)
```

Cause

A directory *inode-number* has been found that has more than one entry for "`..`" (the parent directory).

Action

To remove the extra entry for '`..`' (the parent directory), type `y` at the `FIX` prompt. To leave the directory unchanged, type `n`.

```
hard-link-number IS AN EXTRANEOUS HARD LINK TO A DIRECTORY filename (REMOVE)
```

Cause

`fsck` has found an extraneous hard link *hard-link-number* to a directory *filename*. When preening (`-o p` option), `fsck` ignores the extraneous hard links.

Action

To delete the extraneous entry *hard-link-number* type `y` at the `REMOVE` prompt. To ignore the error condition, type `n`.

```
inode-number OUT OF RANGE I=inode-number NAME=filename (REMOVE)
```

Cause

A directory entry *filename* has an inode number *inode-number* that is greater than the end of the inode list. If the `-p` (preen) option is specified, the inode will be removed automatically.

Action

To delete the directory entry *filename* type `y` at the `REMOVE` prompt. To ignore the error condition, type `n`.

```
MISSING '.' I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename
(FIX)
```

Cause

A directory *inode-number* has been found whose first entry (the entry for ".") is unallocated.

Action

To build an entry for "." with inode number equal to *inode-number*, type `y` at the `FIX` prompt. To leave the directory unchanged, type `n`.

```
MISSING '.' I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename
CANNOT FIX, FIRST ENTRY IN
DIRECTORY CONTAINS filename
```

Cause

A directory *inode-number* has been found whose first entry is *filename*. `fsck` cannot resolve this problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

```
MISSING '.' I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename
CANNOT FIX, INSUFFICIENT
SPACE TO ADD '.'
```

Cause

A directory *inode-number* has been found whose first entry is not ".". `fsck` cannot resolve the problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

```
MISSING '..' I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename
(FIX)
```

Cause

A directory *inode-number* has been found whose second entry is unallocated.

Action

To build an entry for ".." with inode number equal to the parent of *inode-number*, type `y` at the `FIX` prompt. (Note that ".." in the root inode points to itself.) To leave the directory unchanged, type `n`.

```
MISSING '..' I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename
CANNOT FIX, SECOND ENTRY IN
DIRECTORY CONTAINS filename
```

Cause

A directory *inode-number* has been found whose second entry is *filename*. `fsck` cannot resolve this problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

```
MISSING '..' I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time DIR=filename
CANNOT FIX, INSUFFICIENT SPACE
TO ADD '..'
```

Cause

A directory *inode-number* has been found whose second entry is not "`..`" (the parent directory). `fsck` cannot resolve this problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

```
NAME TOO LONG filename
```

Cause

An excessively long path name has been found, which usually indicates loops in the file system name space. This error can occur if a privileged user has made circular links to directories.

Action

Remove the circular links.

```
ROOT INODE UNALLOCATED (ALLOCATE)
```

Cause

The root inode (usually inode number 2) has no allocate-mode bits.

Action

To allocate inode 2 as the root inode, type `y` at the `ALLOCATE` prompt. The files and directories usually found in the root inode will be recovered in phase 3 and put into the `lost+found` directory. If the attempt to allocate the root inode fails, `fsck` displays this message and exits: `CANNOT ALLOCATE ROOT INODE`. To terminate the program, type `n`.

```
ROOT INODE NOT DIRECTORY (REALLOCATE)
```

Cause

The root inode (usually inode number 2) of the file system is not a directory inode.

Action

To clear the existing contents of the root inode and reallocate it, type `y` at the `REALLOCATE` prompt. The files and directories usually found in the root inode will be recovered in phase 3 and put into the `lost+found` directory. If the attempt to allocate the root inode fails, `fsck` displays this message and exits :`CANNOT ALLOCATE ROOT INODE`. To have `fsck` prompt with `FIX`, type `n`.

```
UNALLOCATED I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time type=filename(REMOVE)
```

Cause

A directory or file entry *filename* points to an unallocated inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, modify time *modification-time*, and file name *filename* are displayed.

Action

To delete the directory entry *filename*, type `y` at the `REMOVE` prompt. To ignore the error condition, type `n`.

```
ZERO LENGTH DIRECTORY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time
DIR=filename (REMOVE)
```

Cause

A directory entry *filename* has a size *file-size* that is zero. The owner UID, mode *file-mode*, size *file-size*, modify time *modification-time*, and directory name *filename* are displayed.

Action

To remove the directory entry *filename*, type `y` at the `REMOVE` prompt. This results in the `BAD/DUP` error message in phase 4. To ignore the error condition, type `n`.

Phase 3: Check Connectivity Messages

This phase checks the directories examined in phase 2 and reports error conditions resulting from:

- Unreferenced directories
- Missing or full `lost+found` directories

These messages (in alphabetical order) may occur in phase 3:

`BAD INODE` *state-number* `TO DESCEND`

Cause

An internal error has caused an impossible state *state-number* to be passed to the routine that descends the file system directory structure. `fsck` exits.

Action

If this occurs, contact your local service provider or another qualified person.

`DIR I=`*inode-number1* `CONNECTED. PARENT WAS I=`*inode-number2*

Cause

This is an advisory message indicating a directory inode *inode-number1* was successfully connected to the `lost+found` directory. The parent inode *inode-number2* of the directory inode *inode-number1* is replaced by the inode number of the `lost+found` directory.

Action

N / A

`DIRECTORY` *filename* `LENGTH` *file-size* `NOT MULTIPLE OF` *block-number* `(ADJUST)`

Cause

A directory *filename* has been found with size *file-size* that is not a multiple of the directory block size B. (This condition can recur in phase 3 if it is not adjusted in phase 2.)

Action

To round up the length to the appropriate block size, type `y` at the `ADJUST` prompt. When preening, `fsck` displays a warning and adjusts the directory. To ignore this error condition, type `n`.

```
lost+found IS NOT A DIRECTORY (REALLOCATE)
```

Cause

The entry for `lost+found` is not a directory.

Action

To allocate a directory inode and change the `lost+found` directory to reference it, type `y` at the `REALLOCATE` prompt. The previous inode reference by the `lost+found` directory is not cleared and it will either be reclaimed as an unreferenced inode or have its link count adjusted later in this phase. Inability to create a `lost+found` directory displays the message: `SORRY. CANNOT CREATE lost+found DIRECTORY` and aborts the attempt to link up the lost inode, which generates the `UNREF` error message in phase 4. To abort the attempt to link up the lost inode, which generates the `UNREF` error message in phase 4, type `n`.

```
NO lost+found DIRECTORY (CREATE)
```

Cause

There is no `lost+found` directory in the root directory of the file system. When preening, `fsck` tries to create a `lost+found` directory.

Action

To create a `lost+found` directory in the root of the file system, type `y` at the `CREATE` prompt. This may lead to the message `NO SPACE LEFT IN / (EXPAND)`. If the `lost+found` directory cannot be created, `fsck` displays the message: `SORRY. CANNOT CREATE lost+found DIRECTORY` and aborts the attempt to link up the lost inode. This in turn generates the `UNREF` error message later in phase 4. To abort the attempt to link up the lost inode, type `n`.

```
NO SPACE LEFT IN /lost+found (EXPAND)
```

Cause

Another entry cannot be added to the `lost+found` directory in the root directory of the file system because no space is available. When preening, `fsck` expands the `lost+found` directory.

Action

To expand the `lost+found` directory to make room for the new entry, type `y` at the `EXPAND` prompt. If the attempted expansion fails, `fsck` displays: `SORRY. NO SPACE IN lost+found DIRECTORY` and aborts the request to link a file to the `lost+found` directory. This error generates the `UNREF` error message later in phase 4. Delete any unnecessary entries in the `lost+found` directory. This error terminates `fsck` when preening is in effect. To abort the attempt to link up the lost inode, type `n`.

```
UNREF DIR I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time (RECONNECT)
```

Cause

The directory inode *inode-number* was not connected to a directory entry when the file system was traversed. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of directory inode *inode-number* are displayed. When preening, `fsck` reconnects the non-empty directory inode if the directory size is non-zero. Otherwise, `fsck` clears the directory inode.

Action

To reconnect the directory inode *inode-number* into the `lost+found` directory, type `y` at the `RECONNECT` prompt. If the directory is successfully reconnected, a `CONNECTED` message is displayed. Otherwise, one of the `lost+found` error messages is displayed. To ignore this error condition, type `n`. This error causes the `UNREF` error condition in phase 4.

Phase 4: Check Reference Counts Messages

This phase checks the link count information obtained in phases 2 and 3. It reports error conditions resulting from:

- Unreferenced files
- A missing or full `lost+found` directory
- Incorrect link counts for files, directories, symbolic links, or special files
- Unreferenced files, symbolic links, and directories
- Bad or duplicate blocks in files and directories
- Incorrect total free-inode counts

All errors in this phase (except running out of space in the `lost+found` directory) are correctable when the file system is being preened.

These messages (in alphabetical order) may occur in phase 4:

```
BAD/DUP type I=inode-number
OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (CLEAR)
```

Cause

Phase 1 or phase 1B found duplicate blocks or bad blocks associated with file or directory inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed.

Action

To deallocate inode *inode-number* by zeroing its contents, type `y` at the `CLEAR` prompt. To ignore this error condition, type `n`.

```
(CLEAR)
```

Cause

The inode mentioned in the `UNREF` error message immediately preceding cannot be reconnected. This message does not display if the file system is being preened because lack of space to reconnect files terminates `fsck`.

Action

To deallocate the inode by zeroing out its contents, type `y` at the `CLEAR` prompt. To ignore the preceding error condition, type `n`.

```
LINK COUNT type I=inode-number
OWNER=UID MODE=file-mode
SIZE=file-size
MTIME=modification-time COUNT link-count SHOULD BE
corrected-link-count (ADJUST)
```

Cause

The link count for directory or file inode *inode-number* is *link-count* but should be *corrected-link-count*. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed. If the `-o p` option is specified, the link count is adjusted unless the number of references is increasing. This condition does not occur unless there is a hardware failure. When the number of references is increasing during preening, `fsck` displays this message and exits: `LINK COUNT INCREASING`

Action

To replace the link count of directory or file inode *inode-number* with *corrected-link-count*, type `y` at the `ADJUST` prompt. To ignore this error condition, type `n`.

```
lost+found IS NOT A DIRECTORY (REALLOCATE)
```

Cause

The entry for `lost+found` is not a directory.

Action

To allocate a directory inode and change the `lost+found` directory to reference it, type `y` at the `REALLOCATE` prompt. The previous inode reference by the `lost+found` directory is not cleared. It will either be reclaimed as an unreferenced inode or have its link count adjusted later in this phase. Inability to create a `lost+found` directory displays this message: `SORRY. CANNOT CREATE lost+found DIRECTORY` and aborts the attempt to link up the lost inode. This error generates the `UNREF` error message later in phase 4. To abort the attempt to link up the lost inode, type `n`.

```
NO lost+found DIRECTORY (CREATE)
```

Cause

There is no `lost+found` directory in the root directory of the file system. When preening, `fsck` tries to create a `lost+found` directory.

Action

To create a `lost+found` directory in the root of the file system, type `y` at the `CREATE` prompt. If the `lost+found` directory cannot be created, `fsck` displays the message: `SORRY. CANNOT CREATE lost+found DIRECTORY` and aborts the attempt to link up the lost inode. This error in turn generates the `UNREF` error message later in phase 4. To abort the attempt to link up the lost inode, type `n`.

```
NO SPACE LEFT IN / lost+found (EXPAND)
```

Cause

There is no space to add another entry to the `lost+found` directory in the root directory of the file system. When preening, `fsck` expands the `lost+found` directory.

Action

To expand the `lost+found` directory to make room for the new entry, type `y` at the `EXPAND` prompt. If the attempted expansion fails, `fsck` displays the message:

SORRY. NO SPACE IN lost+found DIRECTORY and aborts the request to link a file to the lost+found directory. This error generates the UNREF error message later in phase 4. Delete any unnecessary entries in the lost+found directory. This error terminates fsck when preening (-o p option) is in effect. To abort the attempt to link up the lost inode, type n.

```
UNREF FILE I=inode-number OWNER=UID
MODE=file-mode SIZE=file-size
MTIME=modification-time (RECONNECT)
```

Cause

File inode *inode-number* was not connected to a directory entry when the file system was traversed. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed. When fsck is preening, the file is cleared if either its size or its link count is zero; otherwise, it is reconnected.

Action

To reconnect inode *inode-number* to the file system in the lost+found directory, type y. This error may generate the lost+found error message in phase 4 if there are problems connecting inode *inode-number* to the lost+found directory. To ignore this error condition, type n. This error always invokes the CLEAR error condition in phase 4.

```
UNREF type I=inode-number OWNER=UID MODE=file-mode SIZE=file-size
MTIME=modification-time (CLEAR)
```

Cause

Inode *inode-number* (whose *type* is directory or file) was not connected to a directory entry when the file system was traversed. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed. When fsck is preening, the file is cleared if either its size or its link count is zero; otherwise, it is reconnected.

Action

To deallocate inode *inode-number* by zeroing its contents, type y at the CLEAR prompt. To ignore this error condition, type n.

```
ZERO LENGTH DIRECTORY I=inode-number OWNER=UID MODE=file-mode
SIZE=file-size MTIME=modification-time(CLEAR)
```

Cause

A directory entry *filename* has a size *file-size* that is zero. The owner *UID*, mode *file-mode*, size *file-size*, modification time *modification-time*, and directory name *filename* are displayed.

Action

To deallocate the directory inode *inode-number* by zeroing out its contents, type `y`. To ignore the error condition, type `n`.

Phase 5: Check Cylinder Groups Messages

This phase checks the free-block and used-inode maps. It reports error conditions resulting from:

- Allocated inodes missing from used-inode maps
- Free blocks missing from free-block maps
- Free inodes in the used-inode maps
- Incorrect total free-block count
- Incorrect total used inode count

These messages (in alphabetical order) may occur in phase 5:

```
BLK(S) MISSING IN BIT MAPS (SALVAGE)
```

Error Message

```
BLK(S) MISSING IN BIT MAPS (SALVAGE)
```

Cause

A cylinder group block map is missing some free blocks. During preening, `fsck` reconstructs the maps.

Action

To reconstruct the free-block map, type `y` at the `SALVAGE` prompt. To ignore this error condition, type `n`.

```
CG character-for-command-option: BAD MAGIC NUMBER
```

Cause

The magic number of cylinder group *character-for-command-option* is wrong. This error usually indicates that the cylinder group maps have been destroyed. When running interactively, the cylinder group is marked as needing reconstruction. `fsck` terminates if the file system is being preened.

Action

If this occurs, contact your local service provider or another qualified person.

```
FREE BLK COUNT(S) WRONG IN SUPERBLK (SALVAGE)
```

Cause

The actual count of free blocks does not match the count of free blocks in the superblock of the file system. If the `-o p` option was specified, the free-block count in the superblock is fixed automatically.

Action

To reconstruct the superblock free-block information, type `y` at the SALVAGE prompt. To ignore this error condition, type `n`.

```
SUMMARY INFORMATION BAD (SALVAGE)
```

Cause

The summary information is incorrect. When preening, `fsck` recomputes the summary information.

Action

To reconstruct the summary information, type `y` at the SALVAGE prompt. To ignore this error condition, type `n`.

Cleanup Phase Messages

Once a file system has been checked, a few cleanup functions are performed. The cleanup phase displays the following status messages.

number-of `files,` *number-of-files*
`used,` *number-of-files* `free (`*number-of*
`frags,` *number-of* `blocks,`

(continued)

```
 percent fragmentation)
```

This message indicates that the file system checked contains *number-of* files using *number-of* fragment-sized blocks, and that there are *number-of* fragment-sized blocks free in the file system. The numbers in parentheses break the free count down into *number-of* free fragments, *number-of* free full-sized blocks, and the *percent* fragmentation.

```
***** FILE SYSTEM WAS MODIFIED *****
```

This message indicates that the file system was modified by `fsck`. If this file system is mounted or is the current root (`/`) file system, reboot. If the file system is mounted, you may need to unmount it and run `fsck` again; otherwise, the work done by `fsck` may be undone by the in-core copies of tables.

```
filename FILE SYSTEM STATE SET TO OKAY
```

This message indicates that file system *filename* was marked as stable. Use the `fsck -m` command to determine if the file system needs checking.

```
filename FILE SYSTEM STATE NOT SET TO OKAY
```

This message indicates that file system *filename* was *not* marked as stable. Use the `fsck -m` command to determine if the file system needs checking.

CHAPTER 44

Troubleshooting Software Administration Problems

This chapter describes problems you may encounter when installing or removing software packages. There are two sections: Specific Software Administration Errors, which describes package installation and administration errors you might encounter, and General Software Administration Problems, which describes behavioral problems that might not result in a particular error message.

This is a list of information in this chapter.

- "Specific Software Administration Errors " on page 718
- "General Software Administration Problems" on page 719

See "Software Administration (Overview)" in *System Administration Guide, Volume 1* for information about managing software packages.

What's New in Troubleshooting Software Administration Problems?

In previous Solaris releases, there was no way to specify a symbolic link target in the `pkgmap` file when creating a software package. This meant a package or patch-related symbolic link was always followed to the source of the symbolic link rather than to the target of the symbolic link when a package was added with the `pkgadd` command. This created problems when upgrading a package or a patch package that needed to change a symbolic link target destination to something else.

In this Solaris release, the default behavior is that if a package needs to change the target of a symbolic link to something else, the target of the symbolic link and not the source of the symbolic link is inspected by the pkgadd command.

Unfortunately, this means that some packages may or may not conform to the new pkgadd behavior.

The PKG_NONABI_SYMLINKS environment variable might help you transition between the old and new pkgadd symbolic link behaviors. If this environment variable is set to true, pkgadd follows the source of the symbolic link.

Setting this variable enables a non-conforming package to revert to the old behavior if set by the administrator before adding a package with the pkgadd command

The new pkgadd symbolic link behavior might cause an existing package to fail when added with the pkgadd command. You might see the following error message in this situation:

```
unable to create symbolic link to <path>
```

If a package doesn't install due to this problem, do the following:

1. If this is a Sun-supplied package, call the Resolution Center and report the non-conforming package name.
2. Set the PKG_NONABI_SYMLINKS environment variable and try adding the package with the pkgadd command again:

```
# PKG_NONABI_SYMLINKS=true
# export PKG_NONABI_SYMLINKS
# pkgadd pkg-name
```

Specific Software Administration Errors

```
WARNING: filename <not present on Read Only file system>
```

Reason Error Occurred	How to Fix the Problem
This error message indicates that not all of a package's files could be installed. This usually occurs when you are using pkgadd to install a package on a client. In this case, pkgadd attempts to install a package on a file system that is mounted from a server, but pkgadd doesn't have permission to do so.	If you see this warning message during a package installation, you must also install the package on the server. See "Software Administration (Overview)" in *System Administration Guide, Volume 1* for details.

General Software Administration Problems

Reason Error Occurred	How to Fix the Problem
There is a known problem with adding or removing some packages developed prior to the Solaris 2.5 release and compatible versions. Sometimes, when adding or removing these packages, the installation fails during user interaction or you are prompted for user interaction and your responses are ignored.	Set the following environment variable and try to add the package again. `NONABI_SCRIPTS=TRUE`

Index

Special Characters

* (asterisk)
    wildcard character 433
+ (plus sign)
    /etc/hosts.equiv file syntax 206
. (dot)
    path variable entry 283
    rcp command syntax 225, 229
? (question mark) in ASET tune files 433
~ (tilde)
    abbreviated pathnames 223
    rcp command syntax 225, 229

Numbers

4.1 systems (running with 5.8 systems) 52

A

absolute mode
    changing file permissions 303, 306
    described 303
    setting special permissions 304
accept command 111
accepting print requests 110, 111, 139
access
    getting to server, with SEAM 409
    obtaining for a specific service 411
    root access
        displaying attempts on console 333, 334
        monitoring su command use 289, 333
        restricting 293, 332
    security 281, 284
        ACLs 285, 312
        file access restriction 282
        firewall setup 284
        login access restrictions 286
        login control 282
        monitoring system usage 283
        network control 282
        path variable setting 283
        physical site security 282
        reporting problems 284
        root access restrictions 332
        root login tracking 289
        setuid programs 283
    sharing files 293
    system logins 287
    to forms
        limiting for printers 144
        limiting for users 143
    to printers
        deleting 88
accounting 515, 522, 525, 543
    automatic 516
    billing users 518, 519, 528, 532
    connect 525 to 527, 531, 532, 538, 541
    daily 527, 543
        reports 529, 536
        step-by-step summary of 527, 529
    disk 526 to 528, 532
    files for 540, 543
    fixing corrupted files
        tacct file 520
        wtmpx file 519, 520, 538

maintaining 520, 522
process 526, 528, 531, 532
raw data 527
reports 529, 536
daily command summary 533, 538, 541 to 543
daily report (tty line utilization) 530, 531
daily usage report 531, 532
last login report 535
overview 529
total command summary (monthly) 534, 542, 543
setting up 519
types of 518, 525
user fee calculation 518, 519, 528, 532
acct.h format files 536
acctcms command 538, 543
acctcom command 536
acctcon command 519, 538, 541
acctdusg command 526, 532, 540
acctprc command 538
acctwtmp command 526, 527, 530
ACLs (access control lists) 312
adding entries 318
changing entries 318
checking entries 317
commands 286
default entries for directories 314
deleting entries 286, 319
described 285, 312
directory entries 314
displaying entries 286, 320
format of entries 312
setting entries 315, 317
valid file entries 313
active file 521, 538, 541
active.MMDD file 521, 541
adapter board (serial port) 235
adding
access to remote printers 72
forms 137
local or attached printers 70
printer description 94
terminfo entry 154
address space map 558, 559
adjusting printer port characteristics 151
administering
character sets 44, 122, 129
fonts 44, 145, 149
forms 44, 135, 145
print filters 44, 130, 134
printers 87, 119
Admintool
terminals and modems 236
Admintool: Serial Ports
Modify window field descriptions 240
prerequisites for running 245
starting 245
alert message priority 618
alerts
for mounting character sets 126
for mounting font cartridges 124
for mounting forms 136, 140
for mounting print wheels 124
for printer faults 101
alias for selectable character set 123, 128, 129
aliases file (ASET)
described 424
example 434
format 434
specification 427
alignment pattern
defining 176
printing 139
protection of 137
allow list
for printer access to forms 144
for user access to forms 144
for user access to printers 105
alphanumeric terminal, *see* terminal
anonymous ftp accounts 215
application threads 551, 552
apptrace 610
ASCII file
file content type of 59
ASET CKLISTPATH_level variable 432
aset command
d option 435
n option 420
p option 436
initiating ASET sessions 416
running ASET interactively 435
running ASET periodically 436
stop running periodically 437

ASET error messages 439
aset.restore utility 428
ASETDIR variable 429, 430
asetenv file
described 425
modifying 425
running ASET periodically 436
ASETSECLEVEL variable
described 429
setting security levels 430
assuming a role 346
asterisk (*)
wildcard character 433
at command 450, 508, 509, 514
l option (list) 511, 512
m option (mail) 509, 510
automatic scheduling of 500
controlling access to 497, 509, 513, 514
error messages 514
overview 450, 497, 508
quitting 450
at job files 508, 512
creating 509, 510
deleting 512
described 450
displaying 511, 512
displaying queue of 511
location of 450
submitting 509
verifying 511
at.deny file 497, 509, 513, 514
atjobs directory 497, 500
atq command 511
audio devices 282
authentication 373
and share command 403
defined 291
DH 351, 359
network security 291, 293
overview of Kerberos 408
remote logins using ftp 217
remote logins using ftp command 215, 216
remote logins using rlogin 204, 213
/etc/hosts.equiv file 206
direct vs. indirect logins 209
network vs. remote system authentication 204, 206, 209
remote logins using rlogin command 207
/etc/hosts.equiv file 206
.rhosts files 207
terminology 375
types 291
authenticator 410
definition 376
authorization 373
defined 291
network security 291, 293
types 291
authorizations database (auth_attr) 339
auth_attr 339
AUTH_DH authentication 359
AUTH_DH client-server session 351, 354
additional transaction 354
client authenticates server 354
contacting the server 352
decrypting the conversation key 353
generating public and secret keys 351
generating the conversation key 352
running keylogin 351
storing information on the server 353
verifier returned to client 353
Automated Security Enhancement Tool (ASET) 415
automatic accounting 516
automatic quota turn on 448, 483
automatic system activity data collection 602, 605
automatic system activity reporting 602, 603
automatic system event execution
repetitive events 449, 497, 506, 507
single events 450, 497, 508, 509, 514
auxiliary (remote) console 619

B

back-end mechanism 387
backup files 450, 510
banner option 98
banner pages
making optional 97
nothing else prints 676
reasons to turn off 97

setting
with Admintool 96
with lpadmin command 96, 97
with nobanner variable 98
with Solaris Print Manager 55
troubleshooting incorrect output 658
turning off 98
baud settings 658
bidirectional modem service 234, 260
billing users 518, 519, 528, 532
booting
displaying messages generated during 615, 616
running sadc command when 602
Bourne shell
ASET working directory specification 430
busstat 547
bye command 217

C

C shell
ASET working directory specification 430
cable pin configuration 662
cache, credential 408
cancel command 114
canceling
print requests 114
by disabling printer 113
for specific user 115
remote logins 204
cartridges, *see* font cartridges
centralized
print configuration 51
changes to share command 403
changing
/etc/system file 459, 463, 465
crontab files 501
date 459, 461, 462
forms paper 139
message of the day 462, 463
number of lock requests 459
number of processes per user 459, 463, 464
number of shared memory segments 459, 464
priority 567, 568, 571
timesharing processes 568, 570, 571
priority of print requests 107, 118
quotas for individual users 493, 494
scheduling classes 569, 570
soft limit time 492, 493
time 459, 461
changing your password 398
with kpasswd command 399
with passwd command 398
character sets
hardware 122, 123
managing 122, 129
number 122
selectable 122, 123
software 122
chargefee command 518, 519, 528, 532
chgrp command
described 285
syntax 302
chkey command 351, 358
chmod command
changing special permissions 307, 308
described 285
syntax 307
choosing your password 397
chown command
described 285
syntax 301
cklist.rpt file
described 418, 422
format 423
ckpacct command 515, 517, 528
class (printer) 99
checking status for 109
defining with lpadmin command 100
not valid for enabling/disabling printer 113
client 375
configuring 382
clients
AUTH_DH client-server session 351, 354
clock
skew 391
synchronizing 391
clock skew 391
closewtmp command 538
closing remote system connections 217
cmsprev file 542

command
table of SEAM 403
Command not found error message 649
commands
monitoring usage of 541, 543
common key 351
calculation 353
Computer Emergency Response Team/Coordination Center (CERT/CC) 284
configuring
ASET 425, 427
printer ports 57
for IA systems 57
configuring NFS servers 386
configuring SEAM clients 382
see also configuration decisions
connect accounting 525 to 527, 531, 532, 541
consadm command 621
disabling an auxiliary console 623
displaying list of auxiliary consoles 622
enabling an auxiliary console 621
consistency checking 487
console
displaying su command use on 333, 334
root access restriction to 332
controlling
access to at command 497, 509, 513, 514
access to crontab command 497, 505 to 507
printer access to forms 144
processes 561
user access to forms 143
conversation key
decrypting 353
generating 352
copying (remote)
using rcp 222, 229
copying files (remote)
using ftp command 216
core file name pattern 628
core files
automatically deleting 509
finding and deleting 475, 478, 479
CPU (central processing unit)
displaying information on
time usage 532, 556, 572
high-usage processes 572
crash utility 615, 637
crashes 616, 644
customer service and 612, 633
deleting crash dump files 479
displaying system information generated by 615, 637
examining crash dumps 637
procedure following 612, 644
rebooting fails after 641
saving crash dump information 626
saving other system information 615
creating
at jobs 509, 510
crontab files 501, 502
form definitions 177
forms 174
print filters 163, 172
creating a credential table 387
creating tickets 393
with kinit 394
cred database 351, 355, 356
cred table
information stored by server 353
credential 377
cache 408
definition 375
obtaining for a server 410
obtaining for a TGS 409
vs. ticket 377
credential cache 408
credential table
adding single entry to 388
changing the back-end mechanism 387
creating 387
credentials 559
described 352
cron daemon 450, 500
cron.allow file 505 to 507
cron.deny file 505, 506
defaults 506
crontab command 449, 506
accounting commands run by 515, 518
controlling access to 505 to 507
denying access 505, 506
limiting access to specific users 505 to 507
overview 497, 505, 506

cron daemon and 500
e option (edit) 501
l option (list) 503
r option (remove) 504, 505
/var/adm maintenance and 615
daily tasks 449
error messages 508
files used by 499, 500
overview 449, 450, 497
quitting without saving changes 501
scheduling of 500
crontab files
creating 501, 502
defaults 499
deleting 504, 505
described 499, 500
displaying 503
editing 501, 502
location of 499
running ASET periodically 416
stop running ASET periodically 437
syntax 500
verifying existence of 502
crypt command 285
csh program 329, 330
.cshrc file 283
ctacct.MMDD file 538, 541
ctmp file 541
current user 223
customer service 612, 633
customizing
exit codes, printer 160
LP print service 44, 151, 179
printer interface program 158, 161
stty modes 159
system logging 616

D

daemon
table of 404
daemons
keyserv 354
lpd 669
lpsched 191, 194, 198
print 183
daily accounting 527, 543
reports 529, 536
daily command summary 533, 538, 541 to 543
daily report (tty line utilization) 530, 531
daily usage report 531, 532
last login report 535
overview 529
step-by-step summary of 527, 529
daily tasks (scheduling with crontab) 449
Data Encryption Standard, *see* DES
date
changing 459, 461
displaying 454, 458
synchronizing with another system 459, 461
date command
accounting data and 526, 527
described 454, 458, 459
daytacct file 532, 538, 541, 543
decrypting
conversation key 353
secret key 351
default printer
ability to set with Solaris Print Manager 55
setting with Admintool 95
setting with lpadmin command 95, 96
defaults
ACL entries for directories 314
at.deny file 513
/etc/syslog.conf file 371
message of the day 463
nice number 571
scheduling classes 567
soft limit time 492, 493
defining
font cartridges 124
print wheels 124
printer characteristics 94
deleting
access to printers 88
ACL entries 286, 319
at jobs 512
backup files 450, 510
core files 478, 479
crash dump files 479

crontab files 504, 505
.rhosts files 207
forms 138
log files 502
old/inactive files 450, 475, 480, 502
print filters 133
temporary files 478
deny list
for printer access to forms 144
for user access to forms 144
for user access to printers 105
DES encryption 350
destination printer 95
destroying tickets 396
/dev/term/a 57
/dev/term/b 57
devices
system device access control 282, 332
df command 467, 582, 583
F option (unmounted file systems) 467
k option (kilobytes) 467, 468, 582, 583
t option (total blocks) 467, 469
examples 468, 582, 583
overview 467, 582
dfstab file 293, 389
kerberos option 389
DH authentication 351
AUTH_DH client-server session 351, 354
mounting files 359
sharing files 358
DH security
for an NIS client 357
for an NIS+ client 355
dial-in modem service 234
dial-out modem service 234
dial-up passwords 327, 330
basic sequence 328
/etc/dialups file 328
/etc/d_passwd file 328 to 330
disabling dial-up logins temporarily 332
dialups file
creating 330
described 328
direct printing 59, 61
direct remote logins
indirect logins vs.
rlogin command 208, 209
using rlogin command 213, 214
directories
abbreviated pathnames 223
ACL entries 314
ASET files 416
checklist task (CKLISTPATH)
setting 426, 432
master files 424
reports 422
working directory 430, 435
current working directory for
processes 559
displaying files and related
information 285, 299, 301
displaying information about 470, 471,
473, 475
permissions
defaults 299
described 296
public directories 298
remote copying 225
setgid permissions 298
size of 473, 475
sticky bit permissions 298
temporary, clearing out 475, 478
working directory 223
disable command 110, 113
disabling
dial-up logins temporarily 332
printers 89, 113
quotas for individual users 494, 495
user logins 325
disabling an auxiliary console
consadm command 623
disk accounting 526 to 528, 532
disk drives
displaying information about
free disk space 582
UFS file system user allocation 474
finding and deleting old/inactive
files 502
disk space
amount of free 467, 582

displaying information about 467
df command 467, 582
directory sizes 473, 475
file sizes 470, 471, 473
mount point 582
UFS file system user allocation 475
file system usage 467, 582
finding and deleting old/inactive files 475, 480
finding files exceeding a size limit 472
finding large files 471
optimizing 467, 480
space
optimizing usage 467
disk space for print queue 53
disktacct file 527, 528, 538, 541
disktacct.MMDD file 538
diskusg command 526, 527
dispadmin command
overview 566
displaying
acct.h format files 536
ACL entries 286, 320
ASET task status 417, 421
at jobs 511, 512
at queue 511
booting messages 615, 616
crash information 615, 637
crontab files 503
date 454, 458
directory information 470, 471, 473, 475
file information 470, 471, 473, 475, 476
file system information 467, 582
files and related information 285, 299, 301
host ID 454, 457
linked libraries 558, 559
LWP information 558
operating system information 456, 457
pacctn file 536
priority information 556, 567
quota information 483, 489, 490
root access attempts on console 333, 334
scheduling class information 556, 566, 567
status of forms 143
su command use on console 333, 334
system activity information 583, 603
system information
commands for 454, 458
time 454, 458
user's login status 323, 324
dmesg command 615, 616
dodisk command 526, 527, 538
caution 527
crontab entry that runs 518
files created by 527, 528, 538, 541
overview 526 to 528
domains (remote logins) 204
dot (.)
path variable entry 283
rcp command syntax 225, 229
download filter 146, 147
downloaded PostScript fonts 148
downloading
fonts 146, 677
host-resident fonts 147
dtmp file 540
du command 473, 475
dump command 646
dumpadm 631
d_passwd file
creating 330
described 328 to 330
disabling dial-up logins temporarily 332
/etc/passwd file and 329

E

editing
crontab files 501, 502
edquota command
changing quotas for individual users 493
disabling quotas for individual users 494, 495
p option (prototype) 487
t option (time limit) 493
overview 483, 484, 492
setting up user quotas 486, 487
eeprom.rpt file 419, 422, 423
enable command 110
enabling
printers 113
enabling an auxiliary console
consadm command 621

encrypting
capturing encrypted passwords 331
files 285
encryption 350
privacy service 373
env.rpt file 419, 422, 423
environment file (ASET)
described 425
modifying 425
running ASET periodically 436
environment variables
ASET 429, 432
ASETDIR 430
ASETSECLEVEL 430
CKLISTPATH_level 426, 432
PERIODIC_SCHEDULE 427, 431, 432, 436
summary table 429
TASKS 426, 432
UID_ALIASES 424, 427, 432
YPCHECK 427, 432
LPDEST 96
PRINTER 96
equals sign (=)
file permissions symbol 304
error message
with kpasswd 399
error messages
at command 514
crash messages 615, 616
crontab command 508
customizing logging of 616
log file for 612, 615
priorities for 618
runacct command 521
sources of 616, 617
specifying storage location for 615 to 617
error protection 538
/etc/acct/holidays file 517, 518
/etc/cron.d/at.deny file 509, 513, 514
/etc/cron.d/cron.allow file 505 to 507
/etc/cron.d/cron.deny file 505, 506
/etc/default/login file
restricting root access to console 332
/etc/default/su file
displaying su command use on console 333, 334
/etc/dfs/dfstab file 293
/etc/dfs/dfstab file
kerberos option 389
/etc/dialups file
described 328, 330
/etc/d_passwd file
described 328 to 330, 332
/etc/passwd file and 329
/etc/group file
ASET check 418
/etc/hosts.equiv file 206
/etc/hosts.lpd file 671
/etc/init.d/acct file 516
/etc/init.d/perf file 602, 605
/etc/inittab file 258
/etc/logindevperm file 282
/etc/lp directory 181
/etc/lp/classes/printer-class file 100
/etc/lp/default file 96
/etc/lp/fd directory 131, 193
/etc/lp/filter.table file 131
/etc/lp/filter.table file
filter added in 132, 133
/etc/lp/forms directory 137
/etc/lp/forms/form-name file 138
/etc/lp/forms/form-name/alert.sh file 141
/etc/lp/forms/form-name/allow file 144
/etc/lp/forms/form-name/deny file 144
/etc/lp/forms/form-name/describe file 137
/etc/lp/printers directory 182
/etc/lp/printers directory
of print client 89
of print server 90
/etc/lp/printers/printer-name/alert.sh file 102
/etc/lp/printers/printer-name/comment file 95
/etc/lp/printers/printer-name/configuration file 98, 104
/etc/lp/printers/printer-name/configuration file 125, 139
banner page setting entered in 99, 124, 126, 139
/etc/lp/printers/printer-name/form.allow file 145
/etc/lp/printers/printer-name/form.deny file 145

/etc/lp/printers/printer-name/users.allow file 106
/etc/lp/printers/printer-name/users.deny file 107
/etc/lp/pwheels/charset-name/alert.sh file 127
/etc/lp/Systems file 89
/etc/motd file 459, 462, 463
/etc/nologin file 326
/etc/nsswitch.conf file 205, 286
/etc/passwd file
ASET checks 418
/etc/d_passwd file and 329
/etc/password file 215
/etc/printcap file 181, 673
/etc/saf/_sactab file 258
/etc/syslog.conf file 616
/etc/system file
changing 459, 463, 465
number of lock requests 459
number of processes per user 463, 464
number of shared memory segments 459, 464
number of processes per user 459
/etc/utmpx file 260
/etc/vfstab file 485, 488
/etc/publickey file 351
execute permissions
symbolic mode 304
execution attributes 343
execution attributes database (exec_attr) 344
execution log (ASET) 420, 421
execution profiles database (prof_attr) 341
exec_attr 344
exit codes (printer interface) 160
standard 159
table of 160
exit command 214
export command 430
export restrictions 380

F

failed login attempts 326, 327
fast print filters 147
fault notification (printer)
ability to set with Solaris Print Manager 55
setting with Admintool 100
setting with lpadmin command 100, 102
values for alerts 101
fault recovery (printer) 56, 103
fcntl information 558 to 560
fd2log file 521, 538, 541
fee file 519, 528, 538, 540
fees (user) 518, 519, 528, 532
file
gsscred 412
kdc.conf 406
table of SEAM 401
file content type 59
ability to set with Solaris Print Manager 55
converted by print filters 130, 164
for common printers 60
menu in Solaris Print Manager 60
non-PostScript printers 60
PostScript 59
simple 59
troubleshooting incorrect output 658
file or group ownership
solving file access problems 652
file systems
disk space usage 467, 582
displaying information about 467, 582
mount point 582
restoring 518, 519, 532
File Transfer Protocol, *see* ftp command
files
accounting 540, 543
backup 450
checking access operations 584
deleting old/inactive 450, 475, 480, 502
displaying information about
listing 470, 471
listing newest 476
size 470, 471, 473, 475
finding and deleting old/inactive 475, 480, 502, 510
finding files exceeding a size limit 472
fixing corrupted
tacct file 520
wtmpx file 519, 520, 538
for setting search path 650

fstat and fcntl information display 558 to 560
lock requests 459
size of 470, 471, 473, 475
usage monitoring 526, 527, 532
used by LP print service 183
files and file systems
abbreviated pathnames 223
ACL entries
adding or modifying 318
checking 317
deleting 286, 319
displaying 286, 320
setting 315, 317
valid entries 313
ASET checks 418
ownership
changing 285
setgid permission and 298
setuid permission and 297
permissions
absolute mode 303, 306
changing 285, 303, 309
defaults 299
described 296
setgid 298
setuid 297
sticky bit 298
symbolic mode 303, 304, 308, 309
umask setting 299
security 284, 295, 312
access restriction 282
ACLs (access control lists) 285, 312
changing ownership 301, 303
changing permissions 303, 309
directory permissions 296
displaying file information 285, 299, 301
encryption 285
file permissions 296
file types 300
overview 281
special file permissions 298, 304, 310
umask default 299
user classes 296
sharing files 293
filtering 59
printing without 59
filters 130
download 146, 147
find command 472, 475, 477 to 479
finding files with setuid permissions 309, 310
searching .rhosts files 211
finding
files exceeding a size limit 472
large files 471
old/inactive files 475, 480, 502, 510
finding and deleting, old/inactive files 511
firewall systems
ASET setup 291, 420
described 284, 290
packet smashing 291
trusted hosts 291
firewall.rpt file 420, 422, 423
fiscrptn file 543
font cartridges 123
alerts for mounting 124, 126
defining 124
mounting 125
naming 123
unmounting 125
fonts
downloaded PostScript 148
downloading 146, 147, 677
host-resident 146, 147
installing 148
managing 145
permanently downloaded 146
PostScript 145
printer-resident 146
styles 122
forcing programs to quit 643
forms
adding 135, 137
alerts for mounting 136
allowing user access 143
changing 135
controlling access to 137
creating 174
default values for 175, 176
definition, creating 177
deleting 135, 138
denying user access 143
displaying attributes of 136

limiting printer access to 144
limiting user access to 143
managing 135
mounting 136, 138
paper (loading and removing) 139
printer access required for 138
setting alerts for mounting 140
tracking forms mounted 136
unmounting 138
viewing status of 143
forwardable ticket 377, 395, 405
frame buffers 282
fsck command 450
fstat information 558 to 560
ftp command 403
authenticating remote logins 215
authentication 291
interrupting logins 204
opening remote system connections 216, 217
remote logins compared to rlogin and rcp 215
ftp sessions 201
anonymous ftp accounts 215
closing remote system connections 217
copying files
from remote system 218
to remote system 220
opening remote system connections 217
ftp sub commands
described 216
ftpd daemon 404

G

get command
copying from remote systems 218
example 219
getfacl command
described 286
displaying ACL entries 320
examples 320
verifying ACLs set on files 316
getting a credential for a server 410
getting a credential for a TGS 409
getting access to a specific service 411
getty 235
.gkadmin file 401
gkadmin command 403
global core file path 627
global priorities
defined 566
displaying 567
group ACL entries
default entries for directories 314
described 313
setting 315, 317
group identifier numbers (GIDs) 287
groups
changing file ownership 302
GSS-API 374
gsscred command 403
gsscred file
changing backend mechanism 387
changing background mechanisms 412
using 412
gsscred.conf file 387, 401
gssd daemon 404

H

hard disk
recommended for print server 54
held signals 558, 559
hex+symbolic stack trace 558
hierarchical realms 378
high ASET security level 417
history log (print requests) 184
holidays file 518
host-resident fonts
downloading 147
PostScript 148
hostid command 454, 457
hosts
in /etc/hosts.equiv file 206
trusted hosts 291
hosts.equiv file 206

I

ID
mapping UNIX to Kerberos principals 412
principals vs. UNIX IDs 387
UNIX 387

indirect remote logins 208, 209
init program 258
initial ticket 405
initializing quotas 484, 487, 488
installation
    post-installation 381
installing
    local or attached printers 70
    PostScript fonts 148
instance 377
integrity 373, 380
    and share command 403
interactive
    commands for restore 646
interactively running ASET 435
interface program (printer)
    customizing 158, 161
    standard 162
Internet firewall setup 284
interprocess communication
    increasing shared memory 464
interrupting programs 643
interrupting remote logins 204
invalid ticket 405
iostat command
    basic information display 580
    xtc option (extended) 581
    overview 579

K

.k5.REALM file 401
.k5login file 401
kadm5.acl file 401
kadm5.keytab file 401
kadmin command 403
kadmin.local command 403
kadmin.log file 401
kadmind daemon 375, 404
kdb5_util command 403
KDC 375
    master 375
    slave 375
    slave vs. master 379
kdc file 401
kdc.conf file 401, 406
kdc.log file 401
kdc.master file 401
kdestroy command 396, 403
KERB authentication
    dfstab file option 389
Kerberos
    and Kerberos V5 374
    and SEAM 374
    terminology 374
Kerberos (KERB) authentication 389
kerberos, dfstab file option 389
kernel thread
    scheduling and 556
    structures 551, 556
key
    definition 375
    private 375
    service 375
    session 375, 408
Key Distribution Center, *see* KDC
key, how to create for an NIS user 357
keyboards 282
keylogin command 355, 356
    running 351
keyserv daemon, verifying 354
keyserver, starting 354
killing processes 552, 563, 564
kilobytes
    file system disk usage in 582
kinds of tickets 404
kinit command 394, 403
    F 395
    ticket lifetime 406
klist command 395, 403
    –f option 395
klwp structure 551
Korn shell
    ASET working directory specification 430
kpasswd command 399, 403
    error message 399
    vs. passwd command 399
kprop command 403
kpropd daemon 404
kpropd.acl file 401
krb5.conf file 401
krb5.keytab file 401
krb5cc_uid file 401
krb5kdc daemon 375, 404
ksh program 329, 330

kthread structure 551
ktkt_warnd daemon 404
ktutil command 403

L

large files 471
last login report 535
lastdate file 538, 541
lastlogin command 538
libraries (linked) 558, 559
lifetime of ticket 406
limiting
  printer access to forms 144
  user access to forms 143
line discipline 259
line usage monitoring 525, 526, 530, 531, 543
lineuse file 538, 543
linked libraries 558, 559
linking remote logins 207
listing
  files and directories 470, 471, 476
  processes being executed 557
loading form paper 139
local or attached printer
  adding by using Solaris Print Manager 70
local printer
  defined 52
  task map for setting up 63
local printing 195
lock file 521, 538, 541
lock1 file 538
locks
  increasing number of requests for 459
log file 541
log files 194
  ASET execution log 420, 421
  cleaning out 194
  codes in request log 185
  deleting automatically 502
  for LP print service 183, 657
  monitoring su command 289, 333
  print queue 184
  print request history log 184
  requests 194
log.MMDD file 541
logging in
  displaying user's login status 323, 324
  remote logins 204, 218
    authentication (rlogin) 204, 207
    direct vs. indirect (rlogin) 208, 209
    finding who is logged in 212
    ftp command 216
    interrupting 204
    linking logins 207
    opening ftp connection 216, 217
    using rlogin 213, 214
  root login
    access restrictions 332
    account 287, 332
    restricting to console 332
    tracking 289
  security
    access restrictions 286
    saving failed attempts 326, 327
    system access control 282
    system device access control 282
    tracking root login 289
  system logins 287
logging out (remote systems) 214
.login file 283
  restricting root access to console 332
  restricting root access to devices 332
login monitoring
  last login 535, 538, 543
  number of logins 532
  time usage 525, 526, 528, 532
logindevperm file 282
loginlog file 538, 542, 543
  overview 326
  saving failed login attempts 326, 327
logins command
  displaying user's login status 323, 324
  displaying users with no passwords 325
  syntax 324, 325
low ASET security level 417
LP commands 681
LP print service
  checking basic functions of 664, 668
  configuration files in 181
  customizing 44, 151, 177
  daemons 183
  defined 179
  defining printer characteristics to 52

directories in 180
files used by 183
hung LP commands 681
interface program 193
log files 184, 657
overview of 45, 179
structure of 180
tracking forms 136
tracking print wheels 124
troubleshooting 657, 659
LP print service scheduler, *see* print scheduler
LP print spooler 79
lpadmin command
adding printer description with 94
adjusting printer port characteristics with 153
defining font cartridges with 124
defining print wheels with 124
defining printer class with 100
limiting access to printers with 106
limiting printer access to forms with 144
making banner pages optional with 97
mounting font cartridge with 126
mounting forms with 139
mounting print wheel with 126
setting alerts to mount forms with 140
setting alerts to mount print wheels with 127
setting default printer with 96
setting printer fault alerts with 101
setting printer fault recovery with 104
unmounting forms with 139
lpd daemon 669
LPDEST environment variable 96
lpfilter command 131
lpsched daemon 191, 194, 198
lpsched log file 194
ls command 470, 471, 476
checking directory sizes 470
l option (size in bytes) 471
s option (size in blocks) 471
t option (newest files) 476
LWPs (lightweight processes)
defined 551
displaying information on 558
processes and 551
structures for 551

M

managing passwords 397
mapping UNIX IDs to Kerberos principals 412
mask ACL entries
default entries for directories 314
described 313
setting 315, 317
master and slave KDCs 379
master files (ASET) 418, 424
master KDC 375
maximums
finding files exceeding maximum size 472
nice number 571
priority 567
maxuprc parameter 463, 464
max_life 406
max_renewable_life 407
mech file 401
medium ASET security level 417
memory
displaying information on
amount installed 454, 458
virtual memory statistics 454, 458
process structures and 551
shared
increasing number of segments 459, 464
process virtual memory 552
virtual
displaying information on 454, 458
process 552
message of the day (MOTD) facility 459, 462, 463
messages file 612, 616
messages.n file 615
mget command
copying from remote systems 218
example 219
minimums
nice number 571
priority 567
minus sign (-)
file permissions symbol 304
/etc/hosts.equiv file syntax 206
modem

defined 234
modems
bidirectional service 234, 260
dial-in service 234
dial-out service 234
different ways to use 234
menu items in Admintool: Serial Ports 243
overview of Admintool: Serial Ports 239
setting up 247
setting up for use with UUCP 249
tools for managing 236
monacct command
crontab entry that runs 517
files used/produced by 543
monthly command summary and 533, 534
runacct command and 529, 538
scheduling running of 515
monitoring
su command use 289, 333
system usage 283
monthly command summary 534
monthly tasks 450
MOTD (message of the day) facility 459, 462, 463
motd file 459, 462, 463
mount
and Kerberos 404
and security mode 404
mount point 582
mounting
font cartridges 125
forms 136, 138
print wheels 125
mouse (system device access control) 282
moving print requests 116, 117
mput command
copying to remote systems 220
example 222
multiple files (ftp) 218

N

names 450
network authentication for remote logins 204, 206, 209
network printer
adding 74
defined 74
task map for setting up 63
network security 289, 293
authentication 291, 293
authorization 291, 293
firewall systems 290, 291
described 284, 290
need for 284
packet smashing 291
trusted hosts 291
issues 282
overview 281, 289
Network Time Protocol, *see* NTP
networks
recognizing access problems 653
newkey command 351, 357
NFS server 386
configuring 386
NFS system 350
NFS systems (ASET) 428, 429
nice command 570 to 572
nice number 556, 571
NIS+
ASET checks 427
authentication 291
authorization 291
cred database 350, 356
publickey database 351
nisaddcred command 351, 355
nlsadmin command 262
nobanner option 97, 98
nobody user 293
non-hierarchical realms 378
non-PostScript printers 60, 122, 123
nsswitch.conf file 205

O

obtaining a credential for a server 410
obtaining a credential for a TGS 409
obtaining access to a specific service 411
obtaining forwardable tickets 395
obtaining tickets 393
with kinit 394
opening remote system connections 216, 217
operating system 456, 457

optimizing disks 467, 480
other ACL entries
default entries for directories 314
described 313
setting 315, 317
ovsec_adm.xxxxx file 401
ownership of files
ACLs and 285, 312
changing 285, 301
changing group ownership 302
owtmp file 542

P

pacctn file
displaying 536
monitoring size of 528, 538
overview 528, 538, 540
packet transfers
firewall security 284
packet smashing 291
PAM 376, 402
configuration file 402
try_first_pass 400
pam.conf file 401, 402
panic: messages 615
parallel printer 57
parity bit 659
passwd command 398
try_first_pass 400
vs. kpasswd command 398
passwd file
ASET checks 418
/etc/d_passwd file and 329
password 397
and policies 399
changing 398
changing with kpasswd command 399
changing with passwd command 398
management 397
suggestions on choosing 397
UNIX vs. Kerberos 397
password management 397
passwords
authentication for remote logins
ftp command 215, 217
rlogin 213
rlogin command 204, 209
capturing encrypted passwords 331
dial-up passwords 327, 330
basic sequence 328
disabling dial-up logins temporarily 332
/etc/dialups file 328
/etc/d_passwd file 328 to 330
displaying users with no passwords 325
eeprom security 287
login security 282, 286, 288
secret-key decryption 351
system logins 287, 288
path variable 283
pathnames
rcp command
absolute vs. abbreviated 223
syntax options 223
tilde (~) in 223
pcred command 559
pending signals 558, 559
per-process core file path 627
perf file 602
performance
activities that are tracked 553
automatic collection of activity data 602, 604
books on 550
file access 584
manual collection of activity data 584, 603
process management 551, 555, 563, 571
reports on 583
system activity monitoring 553, 584, 602
tools for monitoring 553
PERIODIC_SCHEDULE variable
described 429
scheduling ASET 427, 431, 432, 436
permissions
ACLs and 285, 312
ASET handling of 416, 418
changing file permissions 285
absolute mode 303, 306
symbolic mode 303, 304, 308, 309
copying requirements 225
defaults 299
directory permissions 296

file permissions
absolute mode 303, 306
changing 303, 309
described 296
special permissions 298, 304, 310
symbolic mode 303, 304, 308, 309
setgid permissions
absolute mode 304, 308
described 298
symbolic mode 304
setuid permissions
absolute mode 304, 308
described 297
finding files with permissions set 309, 310
security risks 298
symbolic mode 304
special file permissions 298, 304, 310
sticky bit 298
tune files (ASET) 424, 427
umask settings 299
user classes and 296
pfiles command 558 to 560
pflags command 558, 559
physical security 282
pin configuration in cables 662, 679
ping command 212
pkill command 563, 564
pldd command 558, 559
Pluggable Authentication Module, *see* PAM
plus sign (+)
file permissions symbol 304
/etc/hosts.equiv file syntax 206
pmadm command
adding a ttymon service with 267
disabling a ttymon service with 271
enabling a ttymon service with 271
explained 258
listing a ttymon service with 268
pmap command 558, 559
policy
and passwords 399
port 57
defined 234
disabling 251
initialization process of 259
initializing 250
removing service 252
states of (table) 276
port monitor
defined 235
states of (table) 275
ttymon and listen (defined) 235, 260
ports in /etc/dialups file 328
post-installation 381
postdatable ticket 405
postdated ticket 377
PostScript fonts 145, 148
installing 148
PostScript printers 122
character sets for 123
default print filters 131, 132
file content type for 59
printer type for 58
PostScript Reverse printer, *see* Reverse PostScript printer
power cycling 644
power failure recoveries 530
prdaily command
files used by 541, 542
line usage reporting and 543
overview 538
runacct command and 538, 543
primary 377
principal 377
instance 377
name 377
primary 377
principal name 377
realm 377
service 378
user 378
vs. UNIX ID 387
principal name 377
principal.db file 401
principal.kadm5 file 401
principal.kadm5.lock file 401
principal.ok file 401
print client
checking configuration of 666
defined 52
deleting access to printers 88
freeing jobs in 684
print configuration
centralized 51

using SunOS 5.8 and 4.1 systems 52
print daemons 183
print filters
adding 131
bypassing 676, 677
changing 130, 131
characteristics of 170
converting file content type 130
creating 130, 163, 172
defined 130, 193
definitions 166
deleting 133
displaying definition of 133
download 147
fast 147
handling special modes 164
managing 130, 131
options keywords 169
PostScript 131
removing 130, 131
required for printer fault recovery 103
requirements for 165
restoring 130, 131
slow 147
templates to define options 169
to convert from troff to PostScript 173
TranScript 131
types of 163
used to convert files 163
used to put request on hold 165
viewing information about 133
print forms, *see* forms
print jobs, *see* print requests
print management 39, 45
print queue
log of 184
print requests
IDs
canceling print requests by 114, 115
changing priority of print requests using 119
accepting 110, 139
canceling 114
by disabling printer 113
for specific user 115
changing priority of 107, 118
checking status of 108
cleaning out from log file 194
IDs 109
canceling print requests by 114
components of 114
in banner page 96
in status of print request 109
moving print requests by 117
log 185
managing 107
moving to another printer 116, 117
moving to head of queue 118
putting on hold 118
rejecting 89, 110
scheduling 192
setting printer to accept or reject 111
print scheduler
if not running 664
managing 88
restarting 93
starting 94, 192
stopping 93, 192
updating LP system files 191
print server
checking access to 668
checking configuration of 666
checking connections of 665
defined 52, 69
deleting printer from 90
hard disk requirements for 54
setting up 69, 70
spooling space required for 53
system resource requirements for 53
print service, *see* LP print service
print spooler (SVR4) 79
print wheels 123
alerts for mounting 124, 126
defining 124
mounting 125
naming 123
tracking 124
unmounting 125
printcap entry 673
printer class 99
checking status for 109
defining with lpadmin command 100
not valid for enabling/disabling printer 113
setting 56

printer description
ability to set with Solaris Print Manager 55
setting with lpadmin command 94
printer destination
ability to set with Solaris Print Manager 55
setting with Admintool 95
setting with lpadmin command 95
PRINTER environment variable 96
printer interface program 193
printer name 55
printer port
ability to set with Solaris Print Manager 55
adjusting characteristics of 153
characteristics, adjusting 151
configuring 57
enabling multiple ports 57
parallel 57
serial 57
printer status 91
printer type
ability to set with Solaris Print Manager 55
defined in terminfo database 182
not in terminfo database 59
troubleshooting setting of 675
printer(s)
accepting print requests 111
access to
deleting 88
access to forms 138
administering 87, 119
allow list 105
allowing user access 106
baud settings 658
controlling availability of 111
default 96
deleting 88, 91
deleting client access to 88
deleting from print server 88, 90
deny list 105
denying user access 106
description, adding 94
disabling 89, 110, 113
enabling 110, 113
fault alerts 101
fault notification 100
faults
detecting 164
messages 161
restarting printing 683
file content types
effect on filtering 676
incorrect output 676
font cartridges 122
interface program
customizing 158, 161
exit codes 160
standard 159, 162
local 52, 63, 195
local or attached
adding by using Solaris Print Manager 70
network 63, 74
non-PostScript 122, 163
parity bit 659
PostScript
character sets for 123
treatment of text 122
print wheels 122
rejecting print requests 111
remote 196
setting definitions for 94
setting up 63
with Solaris Print Manager 66
settings
baud 658
parity 659
return 659
stty 152, 155, 678 to 680, 683
tab 659
status
checking 91, 108
conflicting messages 661
stty settings 159, 678 to 680, 683

troubleshooting
backup in printer queue 660
conflicting status messages 661, 684
hung LP commands 681
hung printers 659, 682
incorrect output 658, 675
incorrect printer type 675
not accepting requests 664
only banner page prints 676
print service commands 659
unsupported 154
printer(s), alerts, *see* alerts
printer(s), class, *see* printer class
printer(s), fonts, *see* fonts
printer(s), type, *see* printer type
printer-resident fonts 146
printers database 40
printing
banner pages 96
local (diagram) 195
processing or stopping 110
remote 196
special modes 164
spooling directory 188
status messages 684
user fee calculation for 518, 519
priocntl command
overview 566
c option (scheduling class designation) 567, 569
e option (execute) 567
i option (ID type) 568, 569
l option (scheduling class display) 566
m option (max/min priority) 567, 568
p option (priority designation) 567, 568
s option (priority upper limit/change priority) 568, 569
priority (process)
changing 567, 568, 571
timesharing processes 568, 570, 571
designating 567, 568
displaying information on 556, 567
global
defined 566
displaying 567
maximums 567
overview 566, 571
scheduling classes and 567, 568
user-mode priority 566
priority of print requests 107, 118
privacy 373, 380
and share command 403
availability 380
private interfaces 181
private key 351, 375
/proc directory 558
proc structure 551, 556
process accounting 526, 528, 531, 532
process file system (PROCFS) 558
processes 562
accounting utilities for 526, 528, 531, 532
address space map 558, 559
application threads and 551, 552
controlling 561
credentials 559
current working directory for 558 to 560
defined 551
displaying information on 556, 561
acctcom command 536
daily usage report 531, 532
dead processes 536
listing processes being executed 557
LWPs 558
priocntl command 566
ps command 556, 557, 567
tool commands 558, 559, 561
fstat and fcntl information for open files 558 to 560
killing 563, 564
libraries linked into 558, 559
nice number of 556, 570 to 572
number per user 459, 463, 464
priority 571
changing 567, 568, 571
changing timesharing process priority 568, 570, 571
designating 567, 568
displaying information on 556, 567
global priorities 566, 567
maximums 567
overview 565, 571
scheduling classes and 565, 567, 568
user-mode priority 566
restarting 558, 561
runaway 572

scheduling classes 565, 571
changing 569, 570
changing priority of 567, 568, 571
defaults 567
designating 567, 568
displaying information on 556, 566, 567
priority levels and 565, 567, 568
signal actions 558, 559
stack trace 558, 559
stopping temporarily 558, 561
structures for 551, 556
terminology 551, 552
timing 559
tool commands 558, 559
tracing flags 558, 559
trees 558 to 560
troubleshooting 572
waiting for 561
processing printing 110
processors, allocating groups for applications 553
PROCFS (process file system) 558
.profile file 283
prof_attr 341
programs
disk-dependency of 584
forcing to quit running 643
interrupting 643
proxiable ticket 406
proxy ticket 406
prtconf command 454, 458
prun command 561
ps command 556, 558
fields in reports from 556, 571, 572
overview 556
c option (scheduling class) 556, 572
ecl option (global priority) 567
ef option (full information) 557, 562
PS printer type 58
psig command 558, 559
PSR printer type 58
psrset command 553
pstack command 558, 559
pstop command 561
ptacctn.MMDD file 540
ptime command 559
ptree command 558 to 560
public directories 298
public key 351
public-key cryptography
AUTH_DH client-server session 351, 354
changing public and secret keys 351
common key
calculation 353
database of public keys 351
generating keys
conversation key 352
public and secret keys 351
secret key
changing 351
database 351
decrypting 351
generating 351
publickey map 351
put command
copying to remote systems 220
example 221
pwait command 559, 561
pwdx command 558 to 560

Q

qop file 401
question mark (?) wildcard character 433
queue
displaying at queue 511
quitting
at command 450
forcing programs to quit 643
quot command 474, 475
quota command 483, 489, 490, 494
quotacheck command 483, 484, 487, 488
quotaon command 483, 488, 489
quotas 481, 491
administering 491
automatic turning on of 448, 483
changing 492
changing for individual users 493, 494
checking 489
checking for exceeded user quotas 489, 490
checking on file systems 490, 491
configuring file systems for 485
consistency checking 487, 488

disabling for individual users 494, 495
displaying information on 483, 489, 490
initializing 484, 487, 488
overview 483
prototype for multiple users 487
removing 492
requirements 484
setting up 483, 489
soft limit time
changing 492, 493
exceeding 492
turning off 484, 495
turning on 484, 488, 489
user
changing for individual users 493, 494
checking for exceeded 489, 490
disabling for individual users 494, 495
setting up 486, 487
soft limit time 492, 493
verifying 483, 489, 493
quotas file 448, 483, 485

R

rcp command 222, 229, 403
authentication 291
copying between local and remote systems 225, 229
copying directories 225
described 222
examples 225, 229
pathnames
absolute vs. abbreviated 223
syntax options 223, 224
security issues 223
specifying source and target 223
rdate command 459, 461
read permissions
symbolic mode 304
real-time processes
changing class of 569, 570
priority of
specifying relative 567
realm 377, 378
and servers 379
contents of 379
hierarchical vs. non-hierarchical 378
in principal names 377
realms and servers 379
rebooting
fails after crash 641
monitoring 525 to 527, 530, 538
reboots file 538, 541
recognizing network access problems 653
reject command 111
rejecting 110
rejecting print requests 89, 111
remote console messaging 611
remote copying
using rcp 222, 229
remote file copying
using ftp command 216
remote login and security 353
remote logins
authentication 291
authentication (ftp) 215
authentication (rlogin) 204, 207
network authentication vs. remote system authentication 204, 206
/etc/hosts.equiv file 206
.rhosts files 207
authorization 291
closing ftp connection 218
direct vs. indirect (rlogin) 208, 209
domains 204
finding who is logged in 212
ftp commands 216
interrupting 204
linking logins 207
opening ftp connection 216, 217
removing .rhosts files 211
using rlogin 214
using rlogin command 213
verifying remote system operation 212
remote printing
diagram of process 196
user fee calculation for 518, 519
remote systems 204
defined 204
logging in 204, 218
authentication 291
authorization 291

logging out (exit) 214, 215
remote copying
using rcp 222, 229
remote file copying
using ftp command 216
verifying operation 212
removing form paper 139
renewable ticket 406
repetitive system events 449, 497, 506, 507
replayed transactions 353
reporting
automatic 602, 603
reports
ASET 422 to 424
reports (ASET) 422, 423, 429
reports directory 422
repquota command 489 to 491
requests log 194
residentfonts file 148
restarting
lpd daemon 670
print scheduler 93
processes 558, 561
runacct command 521, 538, 540
restore
interactive commands 646
using matching commands 646
restoring (ASET) 428
restoring file systems 518, 532
restricted shell (rsh) 289
return settings 659
Reverse PostScript printer
file content type for 60
method of printing 59
printer type for 58 to 60
.rhosts files
remote system authentication process 205 to 207, 211
rlogin command 403
authentication 204, 207, 291
network vs. remote system authentication 204, 205
/etc/hosts.equiv file 206
.rhosts files 207
described 204
direct vs. indirect logins 208, 209
interrupting logins 204
process after logging in 210, 211
using 213, 214
rlogind daemon 404
rm command 207, 477, 478
role 337
role-based access control
assuming a role 346
authorizations 339
execution attributes 343
execution profiles 341
overview 335
role 337
tools for managing 347
role-based access control (rbac) 335
root access
displaying attempts on console 333, 334
monitoring su command use 289, 333
restricting 293, 332
root file 526
root login
account
access restrictions 332
described 287
restricting to console 332
tracking 289
rprt.MMDD file 528, 542, 543
rpt.MMDD file 538
RS-232-C, *see* serial port
rsh command 403
rsh program 289
rshd daemon 404
runacct command 538, 543
caution 540
crontab entry that runs 538
diagnostics file 538
error messages 521
error protection 538
failure of 521
files used/produced by 540, 541, 543
fixing corrupted files 519, 520, 538
last time executed 541
monacct command and 538
overview 528, 538
prdaily command and 538, 543
progress file 538
restarting 521, 538, 540
running after midnight 538
scheduling running of 515

serial use of 541
states of 538
user fee calculation and 518, 532
runaway processes 572
rusers command 212

S

sa1 command 602
sa2 command 602, 603
SAC, *see* Service Access Controller
sacadm command
adding a ttymon port monitor with 263
disabling a ttymon port monitor with 265
enabling a ttymon port monitor with 266
explained 257
killing a ttymon port monitor with 265
listing a ttymon port monitor with 263
removing a ttymon port monitor with 266
starting a ttymon port monitor with 265
sadc command 602 to 604
sadd file 602
SAF, *see* Service Access Facility
sar command 584, 603
description of all options 603
options listed 603
overview 583, 603
a option (file access) 584
A option (overall performance) 602, 603
b option (buffers) 584, 586
c option (system calls) 586
d option (block devices) 587
e option (ending time) 603
f option (file to extract data from) 603
g option (page-out/memory freeing) 588
i option (interval) 603
k option (kernel memory) 590
m option (interprocess communication) 592
p option (page-in/page faults) 593
q option (queue) 594, 595
r option (unused memory) 595
s option (starting time) 603
u option (CPU usage) 596
v option (system tables) 598
w option (swapping/switching volume) 599
y option (terminal devices) 600
saving
crash dump information 626
failed login attempts 326, 327
scheduler, *see* print scheduler
scheduling ASET execution (PERIODIC_SCHEDULE) 416, 427, 431, 432, 436
scheduling classes 565, 571
changing 569, 570
changing priority of 567, 568, 571
defaults 567
designating 567, 568
displaying information on 556, 566, 567
priority levels and 565, 567, 568
scheduling one-time system events 450, 497, 508, 509, 514
scheduling repetitive system events 449, 497, 506, 507
SEAM
acronym 373
and Kerberos V5 374
commands 403
components of 376
daemons 404
files 401
gaining access to server 409
overview 377
overview of authentication 408
password management 397
post-installation 381
table of commands 403
table of daemons 404
table of files 401
terminology 374
SEAM commands 403
SEAM files 401
search path
files for setting 650
searching
files with setuid permissions 309, 310
.rhosts files 211
users logged in to remote system 212
secret key
changing 351

database 351
decrypting 351
generating 351
secure access 359
secure NIS+, adding a user 356
Secure RPC 349
implementation of 351
Secure RPC authentication 291
security
at command 509
copy operation issues 223
crontab command 506
DH authentication
AUTH_DH client-server session 351, 354
KERB authentication 389
/etc/hosts.equiv file issues 206
.rhosts file issues 207, 211
security mode
see also security, security service
and share command 403
setting up environment with multiple 389
security service 380
see also security, security mode
export restrictions on 380
integrity 380
privacy 380
selectable character sets 122, 123, 128, 129
seminfo_xxxxxx parameters 464
serial port
adapter board 235
defined 234
Serial Port Manager 236
when to use 236
serial printer, adding to system 57
server
and realms 379
definition 375
gaining access with SEAM 409
obtaining credential for 410
servers
AUTH_DH client-server session 351, 354
servers and realms 379
service
definition 375
obtaining access for specific service 411
Service Access Controller 257
Service Access Facility
description 236
files associated with (table) 272
overview of 238, 255
programs associated with (table) 256
services controlled by
states of (table) 275
uses for 238, 255
when to use 236
service key 375
service principal 378
service, security, *see* security service
session key 375, 408
setenv command
ASET security level specification 430
ASET working directory specification 430
setfacl command
adding ACL entries 318
deleting ACL entries 319
described 286
examples 316, 318
modifying ACL entries 318
setting ACL entries 315, 317
syntax 315
setgid permissions
absolute mode 304, 308
described 298
symbolic mode 304
setting
alerts to mount font cartridges 126
alerts to mount print wheels 126
alias for selectable character set 128, 129
setuid permissions
absolute mode 304, 308
described 297
finding files with permissions set 309, 310
security risks 298
symbolic mode 304
setuid programs 283
sh program 329, 330
share command 293
changes to 403
security mode 403
shared memory
increasing number of segments 459, 464
process virtual memory 552

sharing files (network security) 293
shell programs
ASET security level specification 430
ASET working directory specification 430
/etc/d_passwd file entries 329, 330
shminfo_xxxxxx parameters 464
shutacct command 527, 529
shutdown command 529
shutdowns 527, 529, 530
signal actions 558, 559
simple file content type 59
single-sign-on system 374
size
directory 473, 475
file 470, 471, 473, 475
slave and master KDCs 379
slave KDC 375
slave_datatrans file 401
slow print filters 147
soft limit time
changing 492, 493
exceeding 492
software administration
adding packages
troubleshooting 717
removing packages
troubleshooting 717
troubleshooting 717
Solaris environment 52
Solaris Print Manager
ability to define printer with 55
New Attached Printer 70
overview 39
prerequisites for using 66
starting 66
Spacctn.MMDD file 538, 540
space (disk)
amount of free 467, 582
optimizing usage 467, 480
spooling directory
creating for a printer 674
print service 188
spooling space required for print server 53
stack trace 558, 559
standard printer interface program 162
starting
ASET
initiating sessions from shell 416
running interactively 435
lpd daemon 670
print scheduler 94
Solaris Print Manager 66
startup command
acct 527
stash file 375
statefile file 521, 538, 541
states, (runacct command) 538
status
of print requests 108
of printer 91, 108
sticky bit permissions
absolute mode 304, 308
described 298
symbolic mode 304
stopping
dial-up logins temporarily 332
print scheduler 93
printer 110
processes temporarily 558, 561
structures
process 551, 556
stty option 153
stty settings
customizing 159
defaults 152, 155, 678, 679
suggestions for 680, 683
troubleshooting 658
su command
displaying use on console 333, 334
monitoring use 289, 333
su file
displaying su command use on console 333, 334
monitoring su command 333
sulog file 289, 333
Sun Enterprise Authentication Manager, *see* SEAM
SVR4 LP print spooler 79
symbolic links
file permissions 296
latest directory (ASET) 422
symbolic mode
changing file permissions 304, 308, 309

described 303
synchronizing clocks 391
synchronizing date and time with another system 461
sys file 602
sysconf.rpt file 419, 422, 423
syslog.conf file 616
syslogd daemon 615
system activities
automatic collection of data on 602, 604
list of activities tracked 553
manual collection of data on 584, 603
system authentication for remote logins 204, 205
system events
scheduling
one-time events 450, 497, 508, 509, 514
repetitive events 449, 497, 506, 507
system logging (customizing) 616
system messages
specifying storage location for 615
system resources
allocating for print server 53
monitoring 448, 509
accounting 515, 522, 525
accounting system for 543
automatic 449, 509
crashes 616, 644
quotas 481, 491
overview 549
system security 286, 323, 333
dial-up passwords 327, 330
basic sequence 328
disabling dial-up logins temporarily 332
/etc/dialups file 328
/etc/d_passwd file 328 to 330
displaying
user's login status 323, 324
users with no passwords 325
login access restrictions 286
overview 281, 286
passwords 288
restricted shell 289
restricting root login to console 332
root access restrictions 293, 332
saving failed login attempts 326, 327
special logins 287
su command monitoring 289, 333

T

tab settings 659
table of SEAM daemons 404
tacct file 520, 538, 542, 543
tacct.MMDD file 520, 538, 542
tacct.prev file 521
tacctn file 543
tacctprev file 542
TASKS variable
configuring ASET 426, 432
described 429
taskstat utility (ASET) 417, 421
technical support 612, 633
telnet command 403
telnetd daemon 404
templates (print filters) 169
temporary directories 475, 478
terminal
alphanumeric 234
defined 234
distinctions between types of 234
terminal characteristics 159
terminals
line usage monitoring 525, 526, 530, 531, 543
menu items in Admintool: Serial Ports 242
overview of Admintool: Serial Ports 239
process controlling 556
setting up 245
tools for managing 236
troubleshooting bad lines 531
terminfo database
character set names 122
terminfo entry
adding 154
for unsupported printer 157
list of required items 154
selectable character sets in 122
terminology
authentication-specific 375
Kerberos-specific 374
SEAM 374

TGS 409
getting credential for 409
ticket 377
creating 393
creating with kinit 394
definition 375
destroying 396
forwardable 377, 395, 405
initial 405
invalid 405
klist command 395
lifetime 406
maximum renewable lifetime 407
obtaining 393
postdatable 405
postdated 377
proxiable 406
proxy 406
renewable 406
types of 404
viewing 395
vs. credential 377
warning about expiration 384
ticket file, *see* credential cache
ticket, *see* credential cache
file
ticket-granting service, *see* TGS
tilde (~)
abbreviated pathnames 223
rcp command syntax 225, 229
time
changing 459, 461
CPU usage 532, 556, 572
displaying 454, 458
process timing 559
processes accumulating large amounts of CPU time 572
soft limit 492, 493
synchronizing with another system 459, 461
timesharing processes
changing scheduling parameters 568
priority of
changing 568, 570, 571
overview 565
range of 565
/tmp/disktacct.MMDD file 538
tmpfs file system 298
tmpwtmp file 538, 541, 543
tools
process 558, 559
system performance monitoring 553
tools for managing role-based access control 347
total command summary 535, 542
tracing flags 558, 559
TranScript filters 131
transparency 377
trees 558 to 560
troff filter to convert to PostScript 173
Trojan horse 283
troubleshooting 659
printing problems 655, 685
processes 572, 573
software package installation/removal 717
tty lines 531
trusted hosts 291
trusting network environment
remote login
authentication process 205
process after logging in 210, 211
try_first_pass 400
tty lines
troubleshooting bad lines 531
usage monitoring 525, 526, 530, 531, 543
ttyadm command 262
ttymon command 260
ttymon port monitor
adding 263
bidirectional modem service and 260
disabling 265
enabling 266
(figure) 259
killing 265
listing 263
overview of function of 258
removing 266
starting 265
ttymon service
adding 267
disabling 271
enabling 271
listing 268
tune files (ASET)

described 424, 427
example files 433
format 433
modifying 427
rules 433
tune.rpt file 418, 422, 423
tuning
daily command summary and 533
turnacct switch command 528, 538
turning off
quotas 484, 495
turning on
quotas 484, 488, 489
types of tickets 404

U

UFS file systems
displaying information about
user allocation 474, 475
uid_aliases file
described 424
specifying 427
UID_ALIASES variable
aliases file specification 427, 432
described 424, 429
umask 299
uname command 456, 457
UNIX
IDs, in NFS services 387
IDs, vs. principals 387
UNIX systems (crash information) 626
unmounting
font cartridges 125
forms 138
print wheels 125
unsupported printers 61, 154
user
access to printer
ability to limit with Solaris Print Manager 55
limiting with lpadmin command 104
limiting with Solaris Print Manager 104
canceling print requests for 116
user accounts
ASET check 418
displaying login status 323, 324
user ACL entries
default entries for directories 314
described 313
setting 315, 317
user allocation 474, 475
user attributes database (user_attr) 337
user classes of files 296
user fees 519, 528, 532
user logins
last login monitoring 535, 538, 543
number of logins 532
time monitoring 525, 526, 532, 538
user names
current user 223
direct vs. indirect logins (rlogin) 208
finding users logged in to remote system 212
user principal 378
user processes
changing priority 570, 571
CPU usage by 532
number per user 459, 463, 464
priority of 566
user quotas
changing for individual users 493, 494
checking for exceeded 489, 490
disabling for individual users 494, 495
setting up 486, 487
soft limit time 492
user structure 551
user-mode priority 566
user_attr 337
/usr/adm directory 612
/usr/adm/messages file 612
/usr/aset directory 416
/usr/aset/asetenv file
described 425, 436
/usr/aset/masters/tune files
described 424, 427, 433
/usr/aset/masters/uid_aliases file 424
/usr/aset/reports directory
structure 421, 422
/usr/aset/reports/latest directory 422
/usr/lib/lp directory 131
/usr/lib/lp/model directory 193
/usr/lib/lp/postscript directory 130, 193
/usr/lib/sa/sa1 command 602

/usr/lib/sa/sa2 command 602, 603
/usr/lib/sa/sadc command 602 to 604
/usr/proc/bin directory 558, 559
/usr/sbin/crash utility 615, 637
usrgrp.rpt file
  described 419, 422
  example 423
utmp2wtmp command 538
uucico program 329, 330

V

/var file system 53
/var/adm directory
  controlling size of 477, 540
  controlling size of 612, 615
  raw accounting data in 527
/var/adm/acct directory 541
/var/adm/acct/fiscal directory 543
/var/adm/acct/fiscal directory 541
/var/adm/acct/nite directory 541
/var/adm/acct/nite/active file 521, 538, 541
/var/adm/acct/nite/active.MMDD file 538, 541
/var/adm/acct/nite/cms file 538, 541
/var/adm/acct/nite/ctacct.MMDD file 538, 541
/var/adm/acct/nite/ctmp file 541
/var/adm/acct/nite/daycms file 538, 541, 543
/var/adm/acct/nite/daytacct file 532, 538, 541, 543
/var/adm/acct/nite/disktacct file 527, 528, 538, 541
/var/adm/acct/nite/disktacct.MMDD file 538
/var/adm/acct/nite/fd2log file 521, 538, 541
/var/adm/acct/nite/lastdate file 538, 541
/var/adm/acct/nite/lineuse file 538, 541, 543
/var/adm/acct/nite/lock file 521, 538, 541
/var/adm/acct/nite/lock1 file 538
/var/adm/acct/nite/log file 541
/var/adm/acct/nite/log.MMDD file 541
/var/adm/acct/nite/owtmp file 542
/var/adm/acct/nite/reboots file 538, 541
/var/adm/acct/nite/statefile file 521, 538, 541
/var/adm/acct/nite/tmpwtmp file 538, 541, 543
/var/adm/acct/nite/wtmp.MMDD file 520, 538, 542
/var/adm/acct/nite/wtmperror file 541
/var/adm/acct/nite/wtmperror.MMDD file 541
/var/adm/acct/sum directory 528, 541, 542
/var/adm/acct/sum/cms file 538, 542, 543
/var/adm/acct/sum/cmsprev file 542
/var/adm/acct/sum/daycmsfile 538, 542, 543
/var/adm/acct/sum/loginlog file 538, 542, 543
/var/adm/acct/sum/rprt.MMDD file 528, 543
/var/adm/acct/sum/rpt.MMDD file 538
/var/adm/acct/sum/tacct file 520, 538, 542, 543
/var/adm/acct/sum/tacct.MMDD file 520, 538, 542
/var/adm/acct/sum/tacct.prev file 521
/var/adm/acct/sum/tacctprev file 542
/var/adm/dtmp file 540
/var/adm/fee file 519, 528, 538, 540
/var/adm/loginlog file
  overview 326, 327
/var/adm/messages file 612, 616
/var/adm/messages.n file 615
/var/adm/sa/sadd file 603
/var/adm/sa/sadd file 602
/var/adm/Spacctn.MMDD file 538, 540
/var/adm/sulog file 289, 333
/var/lp/logs directory 194
/var/lp/logs/lpsched file 194, 657
/var/spool/cron/atjobs directory 450, 497, 500
/var/spool/cron/crontabs directory 499, 500
/var/spool/cron/crontabs/lp file 194
/var/spool/cron/crontabs/root file 498, 526
/var/spool/cron/crontabs/sys file 602, 605
/var/spool/lp directory 53, 188
/var/spool/lp/requests directory 184
/var/spool/lp/tmp directory 184
variables

ASET environment variables 429, 432
ASETDIR 430
ASETSECLEVEL 430
CKLISTPATH_level 424, 426, 432
PERIODIC_SCHEDULE 427, 431, 432, 436
summary table 429
TASKS 426, 432
UID_ALIASES 424, 427, 432
YPCHECK 427, 432
verifiers
described 352
returned to client 353
window 352
verifying
at jobs 511
crontab file existence 502
quotas 483, 489, 493
remote system operation 212
vfstab file, quotas and 485, 487
viewing tickets 395
virtual memory
displaying information on 454, 458
process 552
viruses
Trojan horse 283
vmstat command
fields in reports from 574
overview 574
S option (swapping statistics) 577
c option (cache flushing) 578
i option (interrupts) 578
s option (system events) 576

W

waiting
for processes 561
warn.conf file 401
warning about ticket expiration 384
Watchdog reset ! message 615
weekly tasks, scheduling with crontab 449
wildcard characters 433
window verifier 352
working directory
defined 223
write permissions
symbolic mode 304
wtmp.MMDD file 520, 538, 542
wtmperror file 541
wtmperror.MMDD file 541
wtmpfix command 519, 538, 541
wtmpx file
daily report and 530
fixing corrupted 519, 520, 538
overview 519, 527, 538
shutdowns and 529

X

xfn 387, 413
xfn_files 387, 413
xfn_nis 387, 413
xfn_nisplus 387, 413

Y

YPCHECK variable
described 429
specifying system configuration file tables 427, 432