System Administration Guide, Volume II

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

SunSoft
A Sun Microsystems, Inc. Business

2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

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

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

Please
Recycle

Contents

Finding System and Network Administration Information xxxv

About This Book xxxix

Part 10 —Managing Printing Services

47. Overview of Print Management 865

Planning Printers on Your Network 866

Distributing Printers on the Network 866

Assigning Print Servers and Print Clients 867

Allocating System Resources for a Print Server 868

Spooling Space 868

Memory 869

Swap Space 869

Hard Disk 869

Choosing a Method to Install Printers. 869

Setting Definitions for Printers. 870

Printer Name 871

Printer Description 871

Printer Port 872

Printer Type 873

File Content Type. 875

Fault Notification. 878

Default Printer Destination 878

Printing Banner Pages. 879

Limiting User Access to a Printer 880

Printer Class 880

Fault Recovery 882

Administering Printers 882

Deleting Printers and Printer Access. 883

Checking Printer Status 883

Restarting the Print Scheduler 883

Managing Print Requests 884

Canceling a Print Request. 884

Moving a Print Request. 885

Changing the Priority of Print Requests 885

Accepting or Rejecting Print Requests 886

Processing or Stopping Printing 887

Administering Character Sets, Filters, Forms, and Fonts 888

The LP Print Service 888

The Structure of the LP Print Service 889

LP Print Service Commands. 898

Functions of the LP Print Service. 899

How LP Administers Files and Schedules Local Print Requests 899

Scheduling Network Print Requests 901

Filtering Print Files 901

What the Printer Interface Program Does. 901

How the `lpsched` Daemon Tracks the Status of Print Requests 902

Tracking Forms. 902

Tracking Print Wheels 903

Receiving Printing Problem Alerts 903

Cleaning Out Log Files 903

Customizing the LP Print Service 912

48. Setting Up Printers 913

Setting Up Printing 914

▼ How to Start Admintool 915

Setting Up a Print Server. 917

▼ How to Add a Local Printer 917

Setting Up a Print Client 919

▼ How to Add Access to a Remote Printer 919

Adding a Network Printer 920

▼ How to Add a Network Printer. 921

Setting Up a Printer With the LP Print Service Commands . . . 923

Example—Initializing a Print Server 925

Example—Adding a Local Printer 927

Example—Adding Access to a Remote Printer 928

49. Administering Printers 929

Managing Printers and the Print Scheduler 930

▼ How to Delete a Printer and Remote Printer Access 930

▼ How to Check the Status of Printers 933

▼ How to Stop the Print Scheduler 936

▼ How to Restart the Print Scheduler 936

Setting Print Definitions 937

▼ How to Add a Printer Description 939

▼ How to Set a System’s Default Printer 940

▼ How to Make Banner Pages Optional 941

▼ How to Turn Off Banner Pages 942

▼ How to Define a Class of Printers 943

▼ How to Set Printer Fault Recovery 944

▼ How to Set Fault Alerts for a Printer 945

▼ How to Limit User Access to a Printer 947

Managing Print Requests 949

▼ How to Check the Status of Print Requests 949

▼ How to Accept or Reject Print Requests for a Printer 951

▼ How to Enable or Disable a Printer 953

▼ How to Cancel a Print Request 954

▼ How to Cancel a Print Request From a Specific User 956

▼ How to Move Print Requests to Another Printer 957

▼ How to Change the Priority of a Print Request 959

50. Managing Character Sets, Filters, Forms, and Fonts 961

Managing Character Sets 962

Selectable Character Sets. 962

Hardware-Mounted Character Sets. 963

Alerts for Mounting Print Wheels or Cartridges 964

▼ How to Define a Print Wheel or Font Cartridge 965

▼ How to Unmount and Mount a Print Wheel or Font Cartridge 966

▼ How to Set an Alert to Mount a Print Wheel or Font Cartridge 968

▼ How to Set Up an Alias for a Selectable Character Set . 970

Managing Print Filters. 973

Creating Print Filters 973

Adding, Changing, Removing, and Restoring Print Filters 973

▼ How to Add a Print Filter. 975

▼ How to Delete a Print Filter 976

▼ How to View Information About a Print Filter 977

Managing Forms 978

Adding, Changing, or Deleting Forms 978

Mounting Forms 979

Defining Alerts for Mounting Forms. 979

Checking Forms 980

Limiting Access to Forms 980

▼ How to Add a Form 981

▼ How to Delete a Form. 982

▼ How to Unmount and Mount a Form. 983

▼ How to Set an Alert to Mount a Form 986

▼ How to View Information About a Form 988

▼ How to View the Current Status of a Form 989

▼ How to Limit User Access to a Form 990

▼ How to Limit Printer Access to a Form 991

Managing Fonts 992

Managing Printer-Resident Fonts 993

Downloading Host-Resident Fonts 994

Installing and Maintaining Host-Resident Fonts. 995

▼ How to Install Downloaded PostScript Fonts 996

▼ How to Install Host-Resident PostScript Fonts 996

51. Customizing the LP Print Service 999

Adjusting Printer Port Characteristics 1000

▼ How to Adjust the Printer Port Characteristics. 1001

Adding a `terminfo` Entry for an Unsupported Printer 1002

▼ How to Add a `terminfo` Entry for an Unsupported Printer. 1005

Customizing the Printer Interface Program 1006

The Standard Printer Interface Program. 1007

Customizing `stty` Modes 1007

Exit Codes 1008

Fault Messages 1009

Using a Customized Printer Interface Program 1009

▼ How to Set Up a Custom Printer Interface Program . . . 1010

Creating a New Print Filter 1011

Writing a Print Filter Program 1011

Creating a Print Filter Definition 1015

▼ How to Create a New Print Filter 1022

Creating a New Printer Form 1025

▼ How to Create a New Form Definition 1027

52. Troubleshooting Printing Problems 1029

Tips on Troubleshooting 1030

Troubleshooting No Output (Nothing Prints) 1030

Troubleshooting Incorrect Output 1033

Troubleshooting Hung LP Print Service Commands 1035

Troubleshooting Idle (Hung) Printers 1035

Troubleshooting Conflicting Status Messages 1036

Troubleshooting Printing Problems 1037

▼ How to Troubleshoot No Output 1037

▼ How to Troubleshoot Incorrect Output 1060

▼ How to Unhang the LP Commands 1065

▼ How to Troubleshoot an Idle (Hung) Printer 1066

▼ How to Resolve Conflicting Status Messages 1068

Part 11 —Working With Remote Systems

53. Working With Remote Systems 1071

Logging In to a Remote System (rlogin) 1072

Authentication for Remote Logins (rlogin). 1073

Linking Remote Logins. 1076

Direct vs. Indirect Remote Logins 1076

What Happens After You Log In Remotely 1078

▼ How to Search for and Remove .rhosts Files. 1080

▼ How to Find Out If a Remote System Is Operating. . . . 1080

▼ How to Find Who Is Logged In to a Remote System . . 1081

▼ How to Log In to a Remote System (rlogin). 1082

▼ How to Log Out From a Remote System (exit). 1083

Logging In to a Remote System (ftp). 1084

Authentication for Remote Logins (ftp) 1084

Essential ftp Commands. 1085

▼ How to Open an ftp Connection to a Remote System 1086

▼ How to Close an ftp Connection to a Remote System 1087

▼ How to Copy Files From a Remote System (ftp). 1087

▼ How to Copy Files to a Remote System (ftp). 1090

Remote Copying With rcp. 1093

Security Considerations for Copy Operations 1093

Specifying Source and Target. 1094

▼ How to Copy Files Between a Local and a Remote System (rcp) 1096

Part 12 —Managing Terminals and Modems

54. Overview of Managing Terminals and Modems 1103

Terminals, Modems, Ports, and Services. 1103

Terminals. 1104

Modems 1104

Ports. 1104

Services 1105

Port Monitors 1105

Tools for Managing Terminals and Modems 1106

Serial Port Manager. 1107

Service Access Facility. 1108

55. Setting Up Terminals and Modems With Serial Port Manager 1109

Setting Up Terminals and Modems 1110

Setting Up Terminals. 1112

Setting Up Modems. 1113

▼ How to Start Serial Port Manager 1115

▼ How to Set Up a Terminal 1117

▼ How to Set Up a Modem 1119

▼ How to Set Up a Modem for Use With UUCP. 1121

▼ How to Initialize a Port. 1123

▼ How to Disable a Port. 1125

▼ How to Remove a Port Service 1126

▼ How To Troubleshoot Terminals and Modems 1127

56. Setting Up Terminals and Modems With the Service Access Facility 1129

Using the Service Access Facility 1130

Overall Administration: sacadm Command 1131

Service Access Controller: SAC Program 1131

SAC Initialization Process. 1131

Port Monitor Service Administrator: pmadm Command 1132

A Port Monitor at Work: ttymon. 1132

Port Initialization Process 1134

Bidirectional Service 1134

Port Monitors: TTY Monitor and Network Listener 1135

TTY Port Monitor: ttymon. 1135

Special ttymon-Specific Administrative Command: ttyadm. 1135

Network Listener Service: listen 1135

Special listen-Specific Administrative Command: nlsadmin. 1136

Administering ttymon Port Monitors 1137

▼ How to Add a ttymon Port Monitor 1137

▼ How to View ttymon Port Monitor Status 1138

▼ How to Stop a ttymon Port Monitor 1138

▼ How to Start a ttymon Port Monitor 1138

▼ How to Disable a ttymon Port Monitor. 1139

▼ How to Enable a ttymon Port Monitor 1139

▼ How to Remove a ttymon Port Monitor 1140

Administering ttymon Services 1141

▼ How to Add a Service 1141

▼ How to View the Status of a TTY Port Service 1142

▼ How to Enable a Port Monitor Service 1143

▼ How to Disable a Port Monitor Service 1144

Administering listen Port Monitors 1144

▼ How to Add a listen Port Monitor 1144

▼ How to View listen Port Monitor Status 1145

▼ How to Stop a listen Port Monitor 1145

▼ How to Start a listen Port Monitor 1146

▼ How to Enable a listen Port Monitor 1146

▼ How to Disable a listen Port Monitor. 1146

▼ How to Remove a listen Port Monitor 1147

Administering listen Port Monitor Services. 1147

▼ How to Add a listen Port Monitor Service. 1147

▼ How to List listen Port Monitor Services 1148

▼ How to Enable a listen Port Monitor Service. 1149

▼ How to Disable a listen Port Monitor Service 1149

Troubleshooting the Network Listener: listen Port Monitor 1150

Reference Material for Service Access Facility Administration 1150

Files Associated With SAF 1150

Service States 1151

Port Monitor States 1151

Port States 1152

Part 13 —Managing System Security

57. Overview of Managing System Security 1155

Granting Access to a Computer System 1156

Maintaining Physical Site Security 1156

Maintaining Login and Access Control. 1156

Restricting Access to Data in Files. 1157

Maintaining Network Control 1157

Monitoring System Usage. 1157

Setting the Correct Path 1157

setuid Programs 1158

Tracking Root Login 1158

Installing a Firewall. 1158

Reporting Security Problems 1159

File Security 1160

User Classes 1160

File Permissions 1160

Directory Permissions 1161

File Types 1161

Special File Permissions (setuid, setgid and Sticky Bit) 1162

File Administration Commands 1163

Default umask 1163

File Encryption. 1164

Access Control Lists (ACLs). 1164

System Security 1165

Login Access Restrictions 1165

Saving Failed Login Attempts 1166

Special Logins. 1166

Passwords 1167

Password Protection Using Dial-Up Passwords 1168

Restricted Shell. 1171

Root Access Restrictions 1172

Monitoring Who Is Using the `su` Command 1172

Network Security 1173

Firewall Systems 1174

Authentication and Authorization 1175

Sharing Files 1176

Restricting Root Access. 1176

Alternative to Secure RPC 1176

Automated Security Enhancement Tool (ASET) 1177

ASET Security Levels 1178

ASET Tasks 1179

ASET Execution Log 1182

ASET Reports 1183

ASET Master Files 1186

ASET Environment File (`asetenv`). 1187

Configuring ASET 1187

Restoring System Files Modified by ASET. 1191

Network Operation Using the NFS System 1191

Environment Variables 1193

ASET File Examples. 1196

58. Securing Files 1199

Displaying File Information 1200

▼ How to Display File Information. 1200

Changing File Ownership 1202

▼ How to Change the Owner of a File 1202

▼ How to Change a Group Ownership of a File 1203

Changing File Permissions 1204

▼ How to Change Permissions in Absolute Mode 1206

▼ How to Change Permissions in Symbolic Mode 1207

Setting and Searching for Special Permissions. 1208

▼ How to Set Special Permissions in Absolute Mode. . . . 1209

▼ How to Find Files With `setuid` Permissions Set 1210

Using ACLs. 1212

ACL Entries for Files 1213

ACL Entries for Directories 1213

▼ How to Set ACL Entries on a File 1215

▼ How to Check If a File Has an ACL 1217

▼ How to Add or Modify ACL Entries on a File. 1218

▼ How to Delete ACL Entries From a File 1219

▼ How to Display ACL Entries for a File 1220

59. Securing Systems 1223

▼ How to Display a User’s Login Status 1224

▼ How to Display Users With No Passwords 1225

▼ How to Save Failed Login Attempts 1226

▼ How to Create a Dial-up Password 1227

▼ How to Temporarily Disable Dial-up Logins 1229

▼ How to Restrict Root Login to the Console 1229

▼ How to Monitor Who Is Using the `su` Command 1229

▼ How to Display Root Access Attempts to the Console. 1230

60. Running ASET 1231

▼ How to Run ASET Interactively 1232

▼ How to Run ASET Periodically 1234

▼ How to Stop Running ASET Periodically 1235

▼ How to Collect Reports on a Server 1235

61. ASET Troubleshooting 1237

Part 14 —Managing System Resources

62. Overview of System Resource Management 1243

System Crashes 1244

Crash Dump Files 1244

Message Files 1244

What to Do After a Crash 1245

Quotas 1247

Accounting Utilities 1248

Types of Daily Accounting 1248

How Daily Accounting Works 1251

Daily Accounting Reports 1252

Accounting Files 1263

Fixing Corrupted Files and `wtmp` Errors. 1265

Executing Routine Tasks Automatically 1266

Scheduling Repetitive Jobs: `crontab`. 1266

How the `cron` Daemon Handles Scheduling 1266

Inside a `crontab` File 1267

`crontab` Command Security. 1268

Scheduling a Single Job: `at` 1269

63. Examining and Changing System Information 1271

Using the Workstation Information Window to Display System Information 1272

▼ How to Add "Workstation Info" to the Workspace Menu 1272

Using Commands to Display System Information 1274

▼ How to Display General System Information 1274

▼ How to Display a System's Host ID Number 1275

▼ How to Display a System's Installed Memory. 1275

▼ How to Display the Date and Time. 1276

Using Commands to Change System Information 1277

▼ How to Synchronize Date and Time From Another System 1278

▼ How to Set a System's Date and Time Manually. 1279

▼ How to Set Up a Message of the Day 1280

▼ How to Set the Number of Processes per User 1281

▼ How to Increase the Number of Pseudo-ttys to 256 . . . 1281

▼ How to Increase the Number of Lock Requests. 1282

▼ How to Increase Shared Memory Segments 1282

64. Saving Crash Dumps 1283

Enabling and Disabling Crash Dumps 1284

▼ How to Create a Directory to Save Crash Dump Files . 1284

▼ How to Reserve Space for Crash Dump Files 1285

▼ How to Enable Crash Dump Files. 1286

▼ How to Disable Crash Dump Files 1287

Viewing System Information Generated by a Crash 1288

▼ How to View Crash and Boot Messages. 1288

▼ How to Examine a Crash Dump 1290

Customizing System Logging 1291

▼ How to Customize System Logging 1293

65. Managing Disk Use 1295

Displaying Blocks and Files Used 1296

▼ How to Display Information About Blocks,
Files, and Disk Space. 1296

Checking the Size of Files 1299

▼ How to Display the Size of Files 1299

▼ How to Find Large Files 1301

▼ How to Find Files That Exceed a Given Size Limit. . . . 1302

Checking the Size of Directories 1303

▼ How to Display the Size of Directories,
Subdirectories, and Files. 1303

▼ How to Display the User Allocation of Local UFS File System 1304

Finding and Removing Old and Inactive Files. 1306

▼ How to List the Newest Files 1306

▼ How to Find and Remove Old or Inactive Files 1308

▼ How to Clear Out Temporary Directories 1309

▼ How to Find and Delete `core` Files 1310

▼ How to Delete Crash Dump Files 1311

66. Managing Quotas 1313

Setting Up Quotas 1314

Guidelines for Setting Up Quotas 1314

▼ How to Configure File Systems for Quotas 1316

▼ How to Set Up Quotas for a User 1317

▼ How to Use Prototype Quotas to Set Up Multiple Users 1318

▼ How to Check Quota Consistency. 1319

▼ How to Turn Quotas On 1320

Checking Quotas 1321

▼ How to Check for Exceeded Quotas 1321

▼ How to Check Quotas on a File System 1322

Changing and Removing Quotas. 1323

▼ How to Change the Soft Time Limit Default 1323

▼ How to Change Quotas for a User 1325

▼ How to Disable Quotas for a User. 1327

▼ How to Turn Quotas Off. 1328

67. Setting Up and Maintaining Accounting 1331

Setting Up Accounting 1332

The `runacct` Program 1332

▼ How to Set Up Accounting 1336

▼ How to Bill Users. 1338

Maintaining Accounting 1339

▼ How to Fix a `wtmp` File 1339

▼ How to Fix `tacct` Errors 1340

▼ How to Restart `runacct` 1341

68. Scheduling System Events 1343

Commands for Scheduling System Events. 1344

Scheduling a Repetitive System Event 1344

Syntax of `crontab` File Entries 1344

Creating and Editing `crontab` Files. 1346

▼ How to Create or Edit a `crontab` File 1346

Displaying `crontab` Files 1348

▼ How to Display a `crontab` File 1348

Removing `crontab` Files 1350

▼ How to Remove a `crontab` File 1350

Controlling Access to `crontab` 1352

▼ How to Deny `crontab` Access 1353

▼ How to Limit `crontab` Access to Specified Users 1354

Scheduling a Single System Event 1356

`at` Command Description 1356

▼ How to Create an at Job 1357

▼ How to Display the at Queue 1358

▼ How to Display at Jobs 1359

▼ How to Remove at Jobs 1360

Controlling Access to at 1361

▼ How to Deny at Access 1361

Part 15 —Managing System Performance

69. Overview of System Performance 1367

System Performance and System Resources. 1368

Other Sources of Information. 1368

Processes and System Performance 1369

Process Commands 1370

Process Priority Levels 1372

Process Troubleshooting 1374

Disk I/O and System Performance 1375

Memory and System Performance. 1375

Swap Space. 1376

Buffer Resources 1376

Kernel Parameters and System Performance 1377

About Monitoring Performance. 1378

The sar Command 1378

The vmstat Command. 1379

The iostat Command. 1381

The df Command 1383

The profil Command. 1383

Automatic Collection of System Activity Data 1384

Collecting System Activity Data With sar. 1385

Monitoring Tools 1387

70. Managing Processes 1389

Displaying Information About Processes (ps). 1390

▼ How to List Processes 1391

Displaying Information About Processes (/proc Tools) 1393

▼ How to Display Information About Processes 1394

Controlling Processes (/proc Tools). 1396

▼ How to Control Processes. 1397

Killing a Process (kill) 1398

▼ How to Kill a Process 1398

Managing Process Class Information 1399

▼ How to Display Basic Information About Process Classes. 1399

▼ How to Display the Global Priority of a Process 1400

▼ How to Designate Priority 1401

▼ How to Change Scheduling Parameters of a Timeshare Process 1402

▼ How to Change the Class of a Process 1403

▼ How to Change the Priority of a Process 1404

71. Monitoring Performance 1405

Displaying Virtual Memory Statistics 1406

▼ How to Display Virtual Memory Statistics. 1406

▼ How to Display System Event Information 1409

▼ How to Display Swapping Statistics................ 1410

▼ How to Display Cache Flushing Statistics 1410

▼ How to Display Interrupts Per Device 1411

Displaying Disk Utilization Information 1412

▼ How to Display Disk Utilization Information 1412

▼ How to Display Extended Disk Statistics............ 1414

Displaying Disk Usage Statistics 1415

▼ How to Display File System Information............ 1415

Monitoring System Activities........................... 1416

▼ How to Check File Access 1416

▼ How to Check Buffer Activity 1417

▼ How to Check System Call Statistics............... 1419

▼ How to Check Disk Activity 1421

▼ How to Check Page-Out and Memory 1423

▼ How to Check Kernel Memory Allocation........... 1424

▼ How to Check Interprocess Communication......... 1426

▼ How to Check Page-In Activity.................. 1427

▼ How to Check Queue Activity.................... 1429

▼ How to Check Unused Memory 1431

▼ How to Check CPU Utilization 1432

▼ How to Check System Table Status................ 1433

▼ How to Check Swap Activity..................... 1435

▼ How to Check Terminal Activity.................. 1436

▼ How to Check Overall System Performance 1437

▼ How to Set Up Automatic Data Collection 1438

72. Monitoring Network Performance. 1439

Monitoring Network Performance 1440

▼ How to Check the Response of Hosts on the Network. 1440

▼ How to Send Packets to Hosts on the Network. 1441

▼ How to Capture Packets From the Network 1442

▼ How to Check the Network Status 1443

▼ How to Display NFS Server and Client Statistics 1446

73. Tuning Kernel Parameters. 1449

▼ How to List the Kernel Parameters. 1450

▼ How to Change the Value of a Kernel Parameter 1451

▼ How to Set the Value of a Kernel Module Variable. . . . 1451

Buffer Cache Parameters. 1452

UFS File System Parameters. 1452

STREAMS Parameters. 1452

Interprocess Communication (IPC™) Parameters. 1453

▼ How to Tune the Interprocess Communication Parameters. 1454

Memory Management Parameters 1454

▼ How to Tune Memory Management Parameters. 1455

Miscellaneous Parameters 1455

▼ How to Tune Miscellaneous Parameters 1456

74. The Scheduler 1457

About the Scheduler 1457

Scheduler Class Policies 1459

Timesharing Class Policies 1459

System Class Policies. 1460

Real-Time Class Policies 1460

Scheduler Configuration 1461

Default Global Priorities 1462

Tunable Parameters 1463

Scheduler Parameter Tables 1466

Kernel-Mode Parameter Table 1470

Figures

Figure P-1 System and Network Administration Books. xxxvi

Figure 47-1 How to Distribute Printers on a Network 866

Figure 47-2 Example Print Configurations on SunOS 5.x and 4.x Systems 867

Figure 47-3 What Happens When a Print Queue Accepts
or Rejects Requests 887

Figure 47-4 What Happens When a Printer Is Enabled or Disabled. 888

Figure 47-5 The `lpsched` Daemon Schedules Local Print Requests 900

Figure 47-6 The Local Printing Process 905

Figure 47-7 Network Printing Between a SunOS 5.x Print Client and
a SunOS 4.1 Print Server. 907

Figure 47-8 Network Printing Between a SunOS 4.1 Print Client and a
SunOS 5.x Print Server 909

Figure 47-9 Network Printing Between a SunOS 5.x Print Client
and a SunOS 5.x Print Server. 911

Figure 48-1 Parts of the Universal Address 926

Figure 53-1 A Remote System. 1072

Figure 53-2 Authentication for Remote Logins (`rlogin`) 1073

Figure 53-3 Direct and Indirect Logins 1077

Figure 56-1 How ttymon Helps Process a Login Request. 1133
Figure 57-1 Basic Dial-up Password Sequence 1170
Figure 57-2 Security Restrictions for Remote Operations. 1173
Figure 57-3 reports Directory Structure 1184
Figure 69-1 Process Structures 1370
Figure 74-1 Sample ts_dptbl Table 1467
Figure 74-2 Sample rt_dptbl Table 1469

Tables

Table P-1 Where to Find System and Network Administration Information xxxvii

Table P-2 SPARC and x86 System Administration Differences xli

Table P-3 Typographic Conventions xlii

Table P-4 Shell Prompts xlii

Table 47-1 Print Definitions Set With Admintool 870

Table 47-2 Choosing File Content Type With Admintool 876

Table 47-3 Some Non-PostScript Printers for Which Sun Does Not Supply Filters 877

Table 47-4 Search Order for Default Printer Destinations 879

Table 47-5 Print Request Management With Admintool 884

Table 47-6 Functions of `accept/reject` and `enable/disable` Commands 886

Table 47-7 Directories for the LP Print Service 889

Table 47-8 Contents of the `/etc/lp` Directory 890

Table 47-9 Contents of the `/usr/lib/lp` Directory 892

Table 47-10 Letter Codes in the LP `requests` Log 895

Table 47-11 Outcome Codes in the LP `requests` Log. 896

Table 47-12 Contents of the `/var/spool/lp` Directory. 897

Table 47-13 Quick Reference to LP Print Service Commands 898

Table 48-1 Task Map: Setting Up Printing 914

Table 48-2 Steps Required to Set Up a Printer by Using Commands 924

Table 49-1 Values for Printer Fault Recovery 937

Table 49-2 Values for Alerts 937

Table 49-3 Values for Allow and Deny Lists 938

Table 49-4 Values for Changing the Priority of a Print Request 949

Table 50-1 Default PostScript Filters 974

Table 51-1 `stty` Default Settings Used by the LP Print Service 1000

Table 51-2 Required `terminfo` Items for a Printer 1002

Table 51-3 Default Printer Port Characteristics 1007

Table 51-4 Printer Interface Program Exit Codes. 1008

Table 51-5 Default Values for `lpfilter` Arguments 1016

Table 51-6 Print Filter Options Keywords 1019

Table 51-7 Default Form Values 1026

Table 52-1 Pin Configuration for NULL Modem Cables 1038

Table 52-2 Default `stty` Settings Used by the Standard Interface Program. 1063

Table 52-3 `stty` Options to Correct Print Output Problems. 1064

Authentication Method (`rlogin`)
and Authentication Method (`rlogin`) 1078

Table 53-2 Essential `ftp` Commands. 1085

Table 53-3 Allowed Syntaxes for Directory and File Names 1095

Table 54-1 Port Monitor Types 1106

Table 54-2 When to Use Solstice Serial Port Manager or Service Access Facility. 1107

Table 55-1 Modify Service Window Items 1111

Table 55-2 Terminal - Hardwired Default Values 1112

Table 55-3 Modem Templates. 1113

Table 55-4 Modem Template Default Values 1113

Table 55-5 Initialize Only - No Connection Default Values 1114

Table 55-6 Required Access Privileges for Setting Up Terminals and Modems 1115

Table 56-1 SAF Control Hierarchy 1130

Table 57-1 File Permissions 1160

Table 57-2 Directory Permissions. 1161

Table 57-3 File Types 1161

Table 57-4 File Administration Commands 1163

Table 57-5 `umask` Settings for Different Security Levels 1164

Table 57-6 ACL Commands 1165

Table 57-7 System Logins 1167

Table 57-8 Types of Authentication and Authorization 1175

Table 57-9 ASET Tasks and Resulting Reports. 1185

Table 57-10 Environment Variables and Their Meanings 1193

Table 58-1 Setting File Permissions in Absolute Mode 1204

Table 58-2 Setting File Permissions in Symbolic Mode. 1205

Table 58-3 Setting Special Permissions in Absolute Mode 1208

Table 58-4 ACL Entries for Files 1213

Table 58-5 Default ACL Entries for Directories 1214

Table 62-1 Raw Accounting Data 1251

Table 62-2 Daily Report Data 1254

Table 62-3 Daily Usage Report Data 1255

Table 62-4 Daily Command Summary 1258

Table 62-5 `acctcom` Options 1261

Table 62-6 Files in `/var/adm` Directory 1263

Table 62-7 Files in the `/var/adm/acct/nite` Directory. 1263

Table 62-8 Files in the `/var/adm/acct/sum` directory 1264

Table 62-9 Files in the `/var/adm/acct/fiscal` Directory 1265

Table 62-10 Default `crontab` Files 1268

Table 63-1 Commands for Displaying System Information. 1274

Table 63-2 Commands for Changing System Information. 1277

Table 64-1 Sources for `syslog.conf` Messages. 1292

Table 64-2 Priorities for `syslog.conf` Messages 1292

Table 66-1 Commands for Setting Up Quotas 1314

Table 66-2 Task Map: Setting Up Quotas 1315

Table 66-3 Commands for Checking Quotas 1321

Table 66-4 Commands for Changing and Removing Quotas 1323

Table 67-1 `runacct` States 1333

Table 67-2 Files Produced by `runacct`. 1335

Table 68-1 Command Summary: Scheduling System Events 1344

Table 68-2 Values for `crontab` Time Fields 1344

Table 69-1 Process Terminology. 1369

Table 69-2 Process Structures 1369

Table 69-3 Process Tools 1371

Table 69-4 Default Swap Sizes 1376

Table 69-5 Kernel Parameters 1377

Table 69-6 Default Settings for Kernel Parameters 1377

Table 69-7 sar Options 1386

Table 70-1 Summary of Fields in ps Reports 1390

Table 70-2 /usr/proc/bin Process Tools That Display Information . . 1393

Table 70-3 /usr/proc/bin Process Tools That Provide Control. 1396

Table 71-1 Output From the vmstat Command. 1406

Table 71-2 Output From the vmstat -S Command 1410

Table 71-3 Output From the vmstat -c Command 1411

Table 71-4 Output From the iostat -n Command 1412

Table 71-5 Output From the iostat -xtc Command 1414

Table 71-6 Output From the df -k Command. 1415

Table 71-7 Output From the sar -a Command. 1417

Table 71-8 Output From the sar -b Command. 1418

Table 71-9 Output From the sar -c Command. 1420

Table 71-10 Output From the sar -d Command. 1421

Table 71-11 Output From the sar -g Command. 1423

Table 71-12 Output From the sar -k Command. 1425

Table 71-13 Output From the sar -m Command. 1426

Table 71-14 Output From the sar -p Command. 1428

Table 71-15 Output From the sar -q Command. 1430

Table 71-16 Output From the sar -r Command. 1431

Table 71-17 Output From the sar -u Command. 1432

Table 71-18 Output From the sar -v Command. 1433

Table 71-19 Output From the sar -w Command. 1435

Table 71-20 Output From the `sar -y` Command. 1436

Table 72-1 Network Monitoring Commands 1440

Table 72-2 Output From the `netstat -r` Command. 1445

Table 72-3 Output of the `nfsstat -c` Command. 1447

Table 72-4 Output of the `nfsstat -m` Command. 1448

Table 73-1 UFS File System Parameters 1452

Table 73-2 STREAMS Parameters 1452

Table 73-3 Interprocess Communication Parameters 1453

Table 73-4 Tuning Interprocess Communication Parameters 1454

Table 73-5 Memory Management Parameters 1454

Table 73-6 Miscellaneous Parameter 1455

Table 74-1 Scheduling Order and Global Priorities. 1462

Table 74-2 Scheduler Parameters 1466

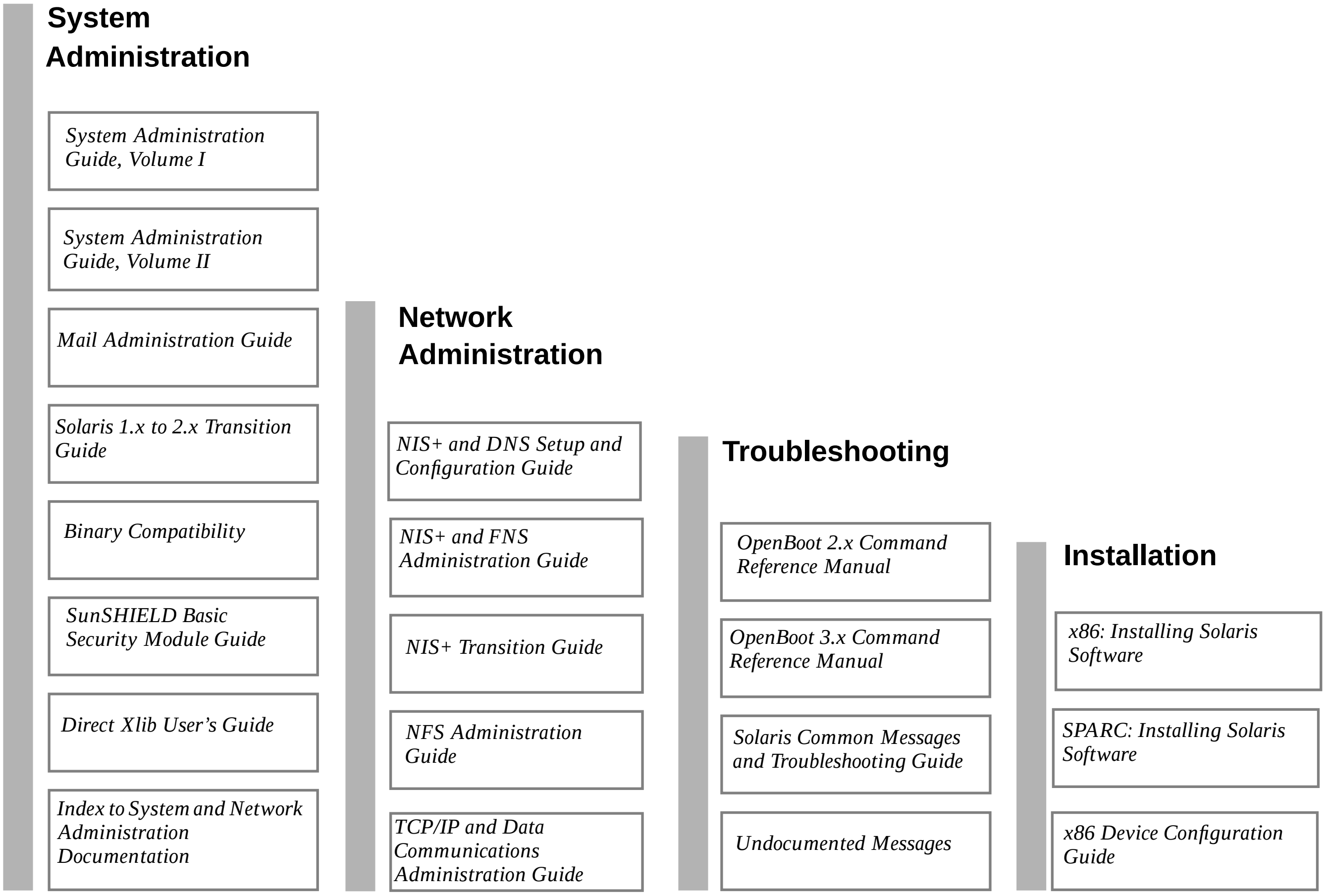
Table 74-3 Fields in the `ts_dptbl` Table 1467

Table 74-4 Fields in the `rt_dptbl` Table 1470

Finding System and Network Administration Information

System Administration Guide, Volume II is part of a multibook set describing Solaris™ 2.5 system and network administration, which is shown in Figure P-1. The multibook set is provided with the *Solaris 2.5 System Administrator AnswerBook.*

Table P-1 lists what information is covered in each of the Solaris system and network administration books. Use this table as a high-level guide to find the right book for the information you need.



System Administration

- *System Administration Guide, Volume I*
- *System Administration Guide, Volume II*
- *Mail Administration Guide*
- *Solaris 1.x to 2.x Transition Guide*
- *Binary Compatibility*
- *SunSHIELD Basic Security Module Guide*
- *Direct Xlib User's Guide*
- *Index to System and Network Administration Documentation*

Network Administration

- *NIS+ and DNS Setup and Configuration Guide*
- *NIS+ and FNS Administration Guide*
- *NIS+ Transition Guide*
- *NFS Administration Guide*
- *TCP/IP and Data Communications Administration Guide*

Troubleshooting

- *OpenBoot 2.x Command Reference Manual*
- *OpenBoot 3.x Command Reference Manual*
- *Solaris Common Messages and Troubleshooting Guide*
- *Undocumented Messages*

Installation

- *x86: Installing Solaris Software*
- *SPARC: Installing Solaris Software*
- *x86 Device Configuration Guide*

Figure P-1 System and Network Administration Books

Table P-1 Where to Find System and Network Administration Information

If You Need Information On ...	**Then Go To ...**	
• Backing up and restoring data • Shutting down and booting a system • Managing - Disks - File systems - Removable Media (CDs, diskettes, PCMCIA) - Software (packages, patches, AnswerBook) - Server and client support - User accounts and groups • Working with remote files	*System Administration Guide, Volume I*	**System Administration**
• Managing - Printing Services - System resources (accounting, crash dumps, disk use and quotas, crontabs, system information) - System performance - Terminals and modems • System security (ACLs, file permissions, ASET)	*System Administration Guide, Volume II*	
• Managing mail	*Mail Administration Guide*	
• Transitioning SunOS 4.x systems to Solaris 2.5	*Solaris 1.x to 2.x Transition Guide*	
• Setting up binary compatibility	*Binary Compatibility Guide*	
• Setting up auditing	*SunSHIELD Basic Security Module Guide*	
• Managing runtime libraries	*Direct Xlib User's Guide*	

Table P-1 Where to Find System and Network Administration Information *(Continued)*

If You Need Information On ...	Then Go To ...	
• Managing NIS+, DNS, or FNS	*NIS+ and DNS Setup and Configuration Guide* *NIS+ and FNS Administration Guide*	**Network Administration**
• Transitioning from NIS to NIS+	*NIS+ Transition Guide*	
• Managing NFS	*NFS Administration Guide*	
• Configuring TCP/IP, PPP, or UUCP	*TCP/IP and Data Communications Administration Guide*	
• Testing hardware and software from the PROM	*OpenBoot 2.x Command Reference Manual* *OpenBoot 3.x Command Reference Manual*	**Troubleshooting**
• Error messages and troubleshooting	*Solaris Common Messages and Troubleshooting Guide* *Undocumented Messages*	
• Installing Solaris	*SPARC: Installing Solaris Software* *x86: Installing Solaris Software* *x86 Device Configuration Guide*	**Installing Solaris Software**

About This Book

System Administration Guide, Volume II is part of a two-volume set that covers a significant part of the Solaris™ system administration information. It includes both SPARC™ and x86 information and describes how to use the Solstice™ AdminSuite tools to perform some of the system administration tasks.

This book assumes that you have already installed the SunOS 5.5™ operating system and Solstice AdminSuite, and you have set up any networking software that you plan to use. The SunOS 5.x operating system is part of the Solaris 2.x product family, which also includes many utilities and OpenWindows™ Version 3.x. The SunOS 5.x operating system is compliant with AT&T's System V, Release 4 operating system.

System Administration Guide, Volume II and *System Administration Guide, Volume I* have replaced the following books previously released with the Solaris operating environment:

- *Security, Performance, and Accounting Administration*
- *User Accounts, Printers, and Mail Administration*
- *Administration Supplement for Solaris Platforms*
- *Common Administration Tasks*
- *File System Administration*
- *Administration Application Reference Manual*
- *Peripherals Administration*

Mail Administration is now covered in the *Mail Administration Guide*.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Solaris 2.x release. To use this book, you should have one to two years of UNIX® system administration experience and preferably a Computer Science B.S. degree or equivalent knowledge.

How This Book Is Organized

This book is split into parts that each cover a major system administration topic. Each part contains chapters that provide both overview and task information.

Most of the overview information about a topic is usually described in the beginning chapters of each part, and the other chapters provide step-by-step instructions on system administration tasks that you need to perform. Each set of steps is usually followed by a way to verify that the task was successfully performed and an example of how to perform the task.

Using AnswerBook to Read This Book

If you are reading this book from within the AnswerBook™ online document viewer, you can double-click on any cross reference, represented by text in a box, to quickly access the referenced information. To return to the previous display, click on the Go Back button.

SPARC and x86 Information

This book provides system administration information for both SPARC and x86 systems. Unless otherwise noted, information throughout this book applies to both types of systems. Table P-2 summarizes the differences between the SPARC and x86 system administration tasks.

Table P-2 SPARC and x86 System Administration Differences

Category	**SPARC**	**x86**
System operation before kernel is loaded	A programmable read-only memory (PROM) chip with a monitor program runs diagnostics and displays device information. It is also used to program default boot parameters and test the devices connected to the system.	The basic input/output system (BIOS) runs diagnostics and displays device information. A Solaris boot diskette with a program called Multiple Device Boot (MDB) is used to boot from non-default boot partitions, the network, or CD-ROM.
Booting the system	Commands and options at the PROM level.	Commands and options at the MDB, primary, and secondary boot subsystems level.

Boot programs	`bootblk` – the primary boot program, loads `ufsboot`
	`ufsboot` – the secondary boot program loads the kernel
		`mboot` – the master boot record, loads `pboot`
		`pboot` – the Solaris partition boot program, loads `bootblk`
		`bootblk` – the primary boot program, load `ufsboot`
		`ufsboot` – the secondary boot program, executes the `/etc/bootrc` script and loads the kernel

Reboot commands	The `shutdown`, `init 6`, or `reboot` commands can be used without additional operation intervention.	The `shutdown`, `init 6`, or `reboot` commands are used but requires operator intervention at the `type any key to continue` prompt.
Disk Controllers	SCSI, IPI, and Xylogics	SCSI and IDE
Disk slices and partitions	Maximum of eight slices, numbered 0-7.	Maximum of four `fdisk` partitions. The Solaris `fdisk` partition may contain up to ten slices, numbered 0-9, but only 0-7 can be used to store user data. You can only have one Solaris `fdisk` partition per disk.
Diskette drives	Desktop systems usually contain one 3.5-inch diskette drive.	Systems may contain two diskette drives: a 3.5-inch and a 5.25 inch drive.

What Typographic Changes Mean

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

Table P-3 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. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser (root) prompt for the Bourne shell and Korn shell.

Table P-4 Shell Prompts

Shell	**Prompt**
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.

- Some code examples have a backslash (\) at the end of a line to specify line continuation, such as the following code example:

```
# pmadm -a -p tcp -s lpd -i root -m ‘nlsadmin -o \
/var/spool/lp/fifos/listenBSD -A \
'\x000202038194180e0000000000000000'‘ -v ‘nlsadmin -V‘
```

  If the line is an example of what to type, ignore the backslashes (don’t type them) and press Return at the end of the line that does not end with a backslash. In the example above, you would ignore the two backslashes when typing the `pmadm` command and press Return after the third line.

- 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.x 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. Commands like `ps` or `df` are duplicated in `/usr/ucb` with different formats and options from the SunOS 5.x commands.

Part 10 — Managing Printing Services

This part provides instructions for managing printing services in the Solaris environment.

47 **Overview of Print Management**
Provides overview information for planning and managing printing services on a network. This chapter provides information on print servers, print clients, and the LP print service.

48 **Setting Up Printers**
Provides step-by-step instructions for setting up a printer on a system and making it available to other systems on the network.

49 **Administering Printers**
Provides step-by-step instructions for administering printers, such as deleting printers, setting print policies, and managing print requests.

50 **Managing Character Sets, Filters, Forms, and Fonts**
Provides step-by-step instructions for setting up and maintaining character sets, print filters, forms, and fonts.

51 **Customizing the LP Print Service**
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.

52 **Troubleshooting Printing Problems**
Provides step-by-step instructions for troubleshooting problems with printing services.

Overview of Print Management 47

This chapter provides fundamental information about managing printers, print clients, and the LP print service. This is a list of the overview information in this chapter.

Planning Printers on Your Network	*page 866*
Allocating System Resources for a Print Server	*page 868*
Choosing a Method to Install Printers	*page 869*
Setting Definitions for Printers	*page 870*
Administering Printers	*page 882*
Managing Print Requests	*page 884*
Administering Character Sets, Filters, Forms, and Fonts	*page 888*
The LP Print Service	*page 888*

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

- Chapter 48, “Setting Up Printers”
- Chapter 49, “Administering Printers”
- Chapter 50, “Managing Character Sets, Filters, Forms, and Fonts”
- Chapter 51, “Customizing the LP Print Service”
- Chapter 52, “Troubleshooting Printing Problems”

Planning Printers on Your Network

The goal for setting up printers on a network is to give users access to one or more printers that are used by other systems. This section provides information about distributing printers across your network to gain the best efficiency and about assigning systems as print servers and print clients.

The SunSoft print client software offers a better solution than the LP print software in the Solaris environment for setting up and managing printers on a network. The SunSoft software supports a name service, which enables you to centralize print administration for a network.

Distributing Printers on the Network

As an administrator, you must decide how to allocate printers and determine whether each printer would be best used if it is dedicated to one system or available to many systems and users. 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.

Figure 47-1 shows an example of how you can have a centralized print configuration and still connect printers to users' systems.

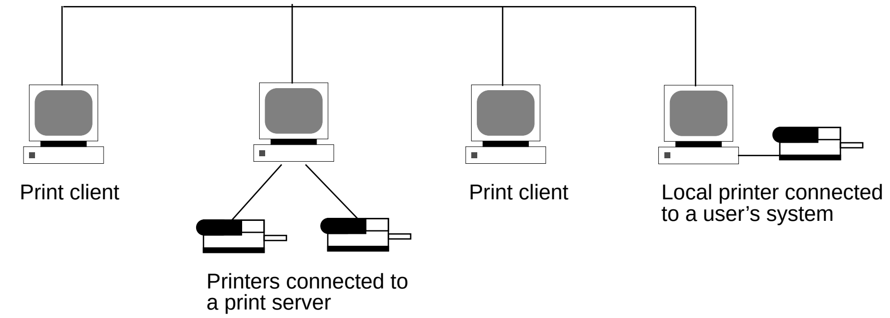

Print client
Print client
Local printer connected to a user's system
Printers connected to a print server

Figure 47-1 How to Distribute Printers on a Network

Assigning Print Servers and Print Clients

You need to decide which systems will have local printers physically attached to them, and which systems will 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 is the software that manages 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.x operating system can print to existing print servers running the SunOS 4.x operating system, and systems running the SunOS 4.x operating system can print to print servers running the SunOS 5.x operating system.

Note – SunOS 5.x is part of the Solaris 2.x operating environment.

Figure 47-2 shows example print configurations on a network with systems running the SunOS 5.x and 4.x operating systems.

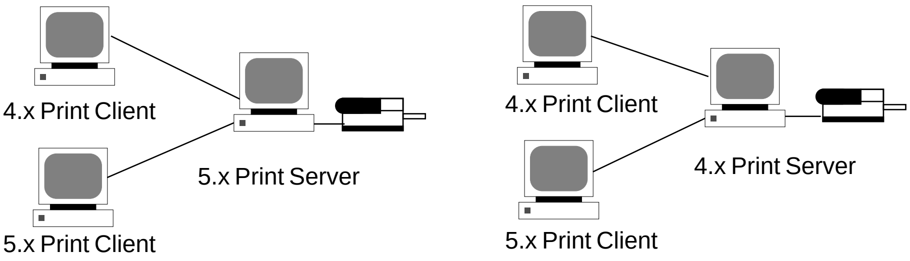


Figure 47-2 Example Print Configurations on SunOS 5.x and 4.x Systems

Allocating System Resources for a Print Server

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 divided into slices. Spooling space may 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 slice, the `/var` directory uses space in the root slice, which is likely to be insufficient.

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

Also carefully evaluate the printing needs and usage 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 may 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 slice, 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 "Managing File Systems" in the *System Administration Guide, Volume I*, for information about mounting file systems and editing the `vfstab` file.

Memory

The Solaris environment requires a minimum of 16 Mbytes of memory to run. A print server does not require additional memory. However, you may 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 "Managing File Systems" in the *System Administration Guide, Volume I*, 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.

Choosing a Method to Install Printers

The SunSoft print client software and the Printer Manager application in Solstice AdminSuite™ offer the best solution for setting up and managing printers on a network. The advantage of the SunSoft print client software is that it supports a name service (NIS or NIS+), which enables you to centralize print administration for a network.

Admintool™ provides an alternative method to install printers in the Solaris environment. Admintool is a graphical user interface that simplifies tasks for setting up and managing printers. See Chapter 48, “Setting Up Printers,” for step-by-step instructions on using Admintool.

You must run Admintool on the system to which you have attached the printer, because Admintool does not enable you to make changes to a remote system. When setting up a printer, Admintool makes the appropriate changes in the system’s `/etc/lp` directory. You can use Admintool to set up a system as a print server or print client only if it is running the SunOS 5.x operating system. Setting up SunOS 4.x print servers and clients is fully described in the SunOS 4.x documentation.

Most of your needs for setting up printing services should be met by Admintool. However, if you have special needs, such as writing scripts, you may want to use the LP print service commands (which underlie Admintool) directly. The setup process with commands is described in “Setting Up a Printer With the LP Print Service Commands” on page 923.

Setting Definitions for Printers

Establishing definitions for the printers on your network is an ongoing task that enables you to 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.

The `lpadmin` command enables you to set all of the print definitions, while Admintool enables you to set only some of them when you install or modify a printer. Table 47-1 lists the print definitions and shows whether you can assign the definition with Admintool.

Table 47-1 Print Definitions Set With Admintool

Print Definition	**Can You Set It With Admintool?**
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

Table 47-1 Print Definitions Set With Admintool *(Continued)*

Print Definition	**Can You Set It With Admintool?**
Limiting user access to a printer	Yes, but with less functionality than the `lpadmin` command
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 known to the system. The name can contain a maximum of 14 alphanumeric characters, which may include dashes and underscores. When administering printers in a complex network, keep printer names unique within the bounds of the administrative domain.

You should also establish conventions when naming printers. Choose printer names that are meaningful and easy to remember. A printer name 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 Admintool. 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 Admintool 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 connect more than two serial printers and a parallel printer to one system.

With Admintool, 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 973 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 1000 for information about customizing the port settings.

Note – If you are using multiple ports on an x86 system or a PowerPC™ microprocessor-based system, keep in mind that the first port is the only port that 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 port configuration file for each additional `asy` (serial) port or `lp` (parallel) port.

For the x86 platform, the pathnames for the port configuration files are:

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

For the PowerPC platform, the pathnames for the port configuration files are:

```
/platform/prep/kernel/drv/asy.conf
/platform/prep/kernel/drv/lp.conf
```

See the *x86 Device Configuration Guide* (available in hardcopy only) for information about configuring serial and parallel ports on x86 systems.

See the Solaris 2.5 PowerPC Edition release notes, installation notes, or device configuration document for information about configuring serial and parallel ports on PowerPC microprocessor-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 LaserWriter®I and LaserWriterII printers.

You can specify the printer type by using the `lpadmin -T` command or Admintool. With Admintool, you can specify the printer type only when you are installing a printer. If you want to change the type of an existing printer, you must delete the printer and reinstall it by using Admintool, otherwise change the printer type by using the `lpadmin` command.

Admintool enables you to 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.

Entries 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   4   7   A   M   a   d   g   j   m   p   s   u   x
2   5   8   B   P   b   e   h   k   n   q   t   v   y
3   6   9   H   S   c   f   i   l   o   r   ti  w   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    epson250       esprit
envision230   ep40          epson2500-80   ethernet
ep2500+basic  ep4000        epson2500-h    ex3000
ep2500+color  ep4080        epson2500-hi8  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        ncr7901        netty-Tabs     newhpkeyboard
ncr7900-na     nec            netty-vi       nuc
ncr7900i       net            network        nucterm
ncr7900i-na    netronics      netx
ncr7900iv      netty          newhp
$
```

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 may 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 may 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 1002 contains some useful tips.

See "Frequently Used Printers" on page 876 for information about the printer type and file content type for printers that are most commonly used with SunOS 5.x software.

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 Admintool. With Admintool, you can select a file contents 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 Admintool 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 file)

When submitting a file for printing, the user can indicate the content type of the file (`lp -T` *content-type*). Otherwise, a file is assumed to be `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.

Admintool 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. Table 47-2 describes the file content types you can choose with Admintool.

Table 47-2 Choosing File Content Type With Admintool

File Contents Choice	LP Print Service Name	Description
PostScript	`postscript`	PostScript files are not filtered. ASCII files are filtered.
ASCII	`simple`	PostScript files are filtered. ASCII files are not filtered.
Both PostScript and ASCII	`simple,postscript`	PostScript files and ASCII files are not filtered.
None	`""`	All files are filtered, except those matching the printer’s type.
Any	`any`	No filtering. 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 Admintool 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 SunOS 5.x 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.

Table 47-3 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 does not supply filtering software for the printers listed in Table 47-3, 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 does not supply filters, contact the printer manufacturer.

Table 47-3 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 1005.

Fault Notification

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 Admintool:

- 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.

Also, if you choose not to send any fault notification, you need a way to 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.

Default Printer Destination

You can specify a default printer destination for a system so you don'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 system's default printer destination by setting any of the following:

- `LPDEST` environment variable
- `PRINTER` environment variable
- 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. Table 47-4 shows the search order for a system’s default printer destination.

Table 47-4 Search Order for Default Printer Destinations

Search Order	**Using `/usr/bin/lp` Command**	**Using `lpr`, `lpq`, and `lprm` Commands**
First	`LPDEST` variable	`PRINTER` variable
Second	`PRINTER` variable	`LPDEST` variable
Third	System’s default printer	System’s default printer

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 can also have an optional title that helps 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 may 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 may 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 through the command line interface only. For step-by-step command-line instructions, see “How to Turn Off Banner Pages” on page 942.

Limiting User Access to a Printer

You may want to control which users can access some or all of the available printers. For example, you may want to 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. (Admintool 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 may access the printer.
Specify `all` in the allow list	All users may access the printer.
Specify `all` in the deny list	All users, except root and lp, 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.

It is best to create allow and deny lists on the print server only and not set up allow and deny lists on print clients. If you create allow and deny lists on the print server only, the print server exclusively controls user access to printers. The benefit of using this strategy is that you do not have to coordinate changes to the print server’s allow and deny lists with print clients.

Printer Class

The print service enables you to group several local 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. When a print client attempts to print to a class of printers on a print server, only the first printer defined in the class is used.

Class names, like printer names, must be unique and may 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.

Fault Recovery

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

Printer faults can be as simple as running out of paper or needing to replace a toner cartridge. Other more serious problems may 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 1022.

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

Administering Printers

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

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

See Chapter 49, "Administering Printers," for step-by-step instructions on how to perform the printer administration tasks.

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. If you delete a local printer from a print server, you should delete the remote printer entry from the print clients. If you move a printer to another print server, you need to delete the old remote print entry from the print clients and add access to the remote printer in its new location.

See "How to Delete a Printer and Remote Printer Access" on page 930 for detailed information on how to delete a local and remote printer. You can use Admintool to delete a local or remote printer; however, Admintool does not enable you to move queued print request to another printer.

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 may need to 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.

Restarting the Print Scheduler

The print scheduler, `lpsched`, handles print requests on both a print client and print server. However, there may 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.

47

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, move, hold, resume, or change the priorities of print requests. These actions can help you keep printing services operating smoothly.

The print commands enable you to perform all print request management tasks. Admintool enables you to perform some print request management tasks when you modify a print server. Table 47-5 lists the print request management tasks you can perform with Admintool.

Table 47-5 Print Request Management With Admintool

Task	**Can You Do With Admintool?**
Canceling a print request	No
Moving a print request	No
Changing priority of print requests	No
Accepting or rejecting print requests	Yes
Processing or stopping printing	Yes

Canceling a Print Request

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

- To cancel requests by request identification number (request ID)
- To cancel requests from a specific user on all, or specified, printers
- To 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
- Logged in as root 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 `lpadmin` command without any options.

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` 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.

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 in the following ways:

- You can put 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.
- You can move 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.
- You can change 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.

Accepting or Rejecting Print Requests

The accept and reject commands—or the Accept Print Requests field in Admintool's Modify Printer window—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. 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 47-6 compares the functions of the accept/reject and enable/disable commands.

Table 47-6 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 887, for information about disabling a printer.

If a print request is rejected, the print service writes 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 will be displayed on users' systems whenever they try to check the printer's queue.

Figure 47-3 shows the point at which processing of print requests is interrupted when a print queue rejects print requests.

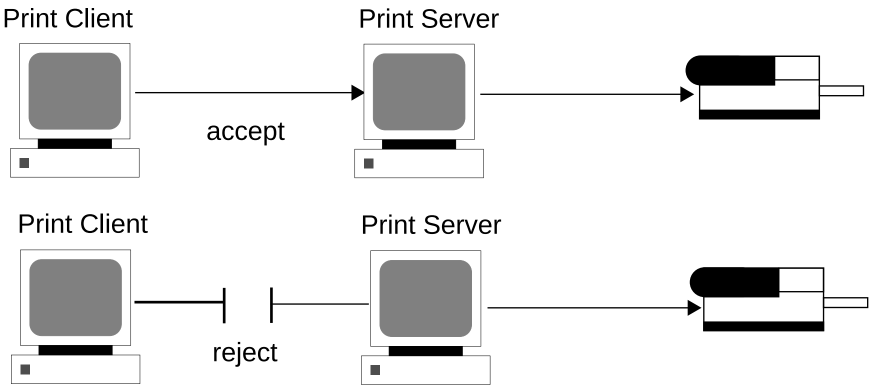


Figure 47-3 What Happens When a Print Queue Accepts or Rejects Requests

Processing or Stopping Printing

The `enable` and `disable` commands—or the Process Print Requests field on Admintool's Modify Printer window—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 886 for information about rejecting print requests.)

You must enable the printer whenever it has been disabled, which may 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.

Figure 47-4 shows the point at which processing of print requests is interrupted when a printer is disabled.

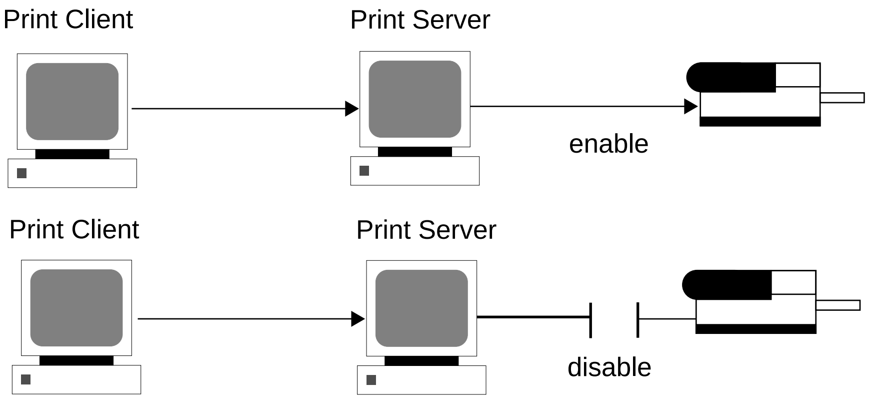


Figure 47-4 What Happens When a Printer Is Enabled or Disabled

Administering Character Sets, Filters, Forms, and Fonts

Depending on your site's requirements and the types of printers you have on the network, you may 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 50, "Managing Character Sets, Filters, Forms, and Fonts" for background information and step-by-step instructions on how to set up and administer character sets, print filters, forms, and fonts.

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 may provide, and the hardware (the printer, system, and network connections).

The Structure of the LP Print Service

This section provides information about the directory structure, files, logs, and commands of the LP print service.

Directories

The many files of the LP print service are distributed among seven directories, as shown in Table 47-7.

Table 47-7 Directories for the LP Print Service

Directory	**Contents**
`/usr/bin`	The LP print service user commands
`/etc/lp`	A hierarchy of LP 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
	`lpNet` – Messages from `lpNet`
	`lpsched.`*n* – Messages from `lpsched`
	`requests.`*n* – Information about completed print requests

`/var/spool/lp`	The spooling directory where files are queued for printing

Configuration Files

The scheduler stores configuration information in LP configuration files located in the `/etc/lp` directory, as described in Table 47-8.

Caution – The configuration files listed in Table 47-8 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 47-8 Contents of the `/etc/lp` Directory

File	Type	Description
`Systems`	ASCII file	Names of systems defined using the `lpsystem` command. Includes every remote system with which the local system can exchange print requests.
`classes`	Directory	Files identifying classes provided by the `lpadmin -c` command.
`default`	ASCII file	Name of the default destination provided by the `lpadmin -d` 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 (remote or 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 the function of the /etc/printcap file in SunOS 4.1.

Note – You can check the contents of the configuration files, but you should not edit them directly. Instead, use the lpadmin(1M)command 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 printer (local or remote) 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
```

Within each of the printer-specific directories, the following files can describe the printer:

- 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 873 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 1002.

Daemons and LP Internal Files

The /usr/lib/lp directory contains daemons and files used by the LP print service, as described in Table 47-9.

Table 47-9 Contents of the /usr/lib/lp Directory

File	Type	Description
bin	Directory	Contains files for generating printing alerts, slow filters, and queue management programs.
lpNet	Daemon	Controls LP requests for network printing.
lpdata	ELF executable file	Lists LP print service configuration information (interactive command).
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 SunOS 5.x 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.

Log Files

The LP print service maintains two sets of log files:

- A list of current requests that are in the print queue (`/var/spool/lp`)
- An ongoing history of print requests (`/var/lp/logs/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 three log files: lpNet, 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
lpNet       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" in Chapter 68, "Scheduling System Events," for suggestions on modifying the cron job for cleaning out the requests log.

The two most important log files for troubleshooting are:

- The lpNet log, which contains information about network printing
- 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, Mon Jun 20 01:24:01 EST
1995
z slw2
C 1
D slw2
F /etc/motd
P 20
t simple
U irving
s 0x0100
#
```

Table 47-10 shows the letter codes and the content of their corresponding lines in the LP `requests` log.

Table 47-10 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.

Table 47-10 Letter Codes in the LP `requests` Log *(Continued)*

Letter	Content of Line
`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 `-r` option of the `lp` 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 Table 47-11.
`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.
`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.

Table 47-11 shows the outcome codes in the LP `requests` log and their descriptions.

Table 47-11 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.

Table 47-11 Outcome Codes in the LP requests Log

Outcome Code	Description
0x0080	The request will print next.
0x0100	The request failed filtering or printing.
0x0200	The request is in transit to a remote printer.
0x0400	The user will notified.
0x0800	A notification is running.
0x1000	A remote system has accepted the request.
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 may be only seconds. Table 47-12 shows the contents of the /var/spool/lp directory.

Table 47-12 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.
fifos	Directory	Pipes that convey networked print requests to and from the lpNet daemon.
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.

Table 47-12 Contents of the `/var/spool/lp` Directory *(Continued)*

File	Type	Description
`system`	Directory	A print status file for the system.
`temp`	Link	Link to `/var/spool/lp/tmp/`*printer-name*, 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

Table 47-13 lists frequently used LP print service commands. You must be root or lp to use the `1M` commands.

Table 47-13 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
`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

Table 47-13 Quick Reference to LP Print Service Commands *(Continued)*

Command	**Task**
`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
- 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 Structure of the LP Print Service" on page 889 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 also schedules all local print requests, as shown in Figure 47-5 on page 900, regardless of whether users issue the requests from an application or from the command line. In addition, the scheduler tracks the status of printers and filters. When a printer finishes printing a request, the scheduler schedules the next request, if there is one in the queue.

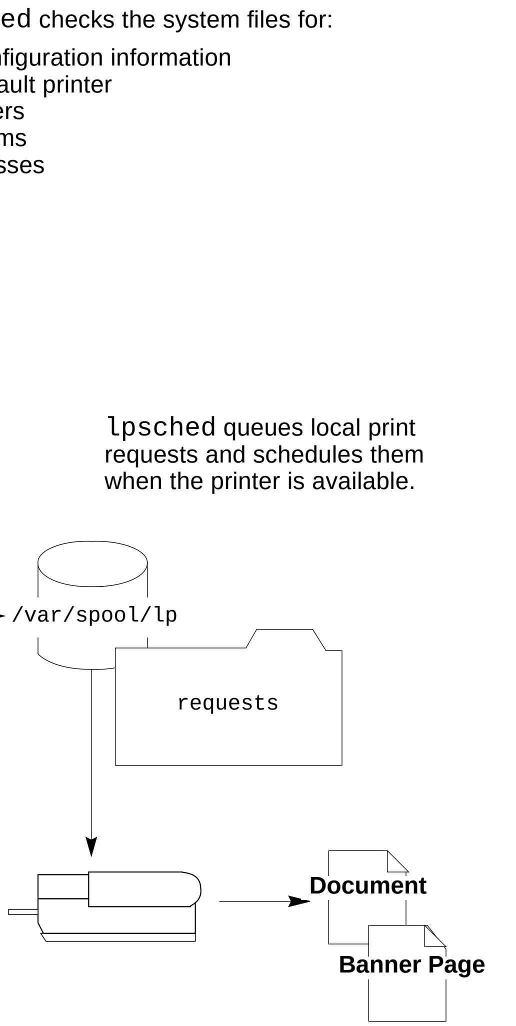


Figure 47-5 The `lpsched` Daemon Schedules Local Print Requests

Each print client and 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 and print server must have one or more lpNet daemons. The lpNet daemon schedules network print requests. The lpNet daemon is started when a system is booted. If you stop and restart the scheduler (using the lpshut and lpsched commands), the lpNet daemon is also stopped and restarted.

The lpNet daemon needs a configured port monitor and registered listen services to handle incoming network requests on each print server running SunOS 5.x system software.

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. SunOS 5.x system software provides print filters in the /usr/lib/lp/postscript directory that cover most PostScript printing 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.

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. Or users can submit print requests specifying a particular form and whether the form is mounted. When the LP print service receives the request, it sends an alert message to root requesting that you mount the form.

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.

Receiving Printing Problem Alerts

The LP print service performs sophisticated error checking. If a printing problem occurs, alerts are sent to the originator of a print request, or to the system administrator, depending on the nature of the problem and what is required to correct it. Users are notified when a print request cannot be completed. If users request it, they are notified by email when a job is successfully completed. The LP print service alerts administrators of problems with printers, and of requests for filters, forms, or character sets.

For problems that require an administrator's attention, the LP print service default is to write an alert message to the console window of the system on which root is logged in.

As the system administrator, you can change the alert policy to receive alert messages via:

- Electronic mail
- A program of your choice

Cleaning Out Log Files

The `lpsched`, `lpNet`, 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 Nov 18 01:24:01 EST
1992
z slw2
C 1
D slw2
F /etc/motd
P 20
t simple
U irving
s 0x1010
#
```

How Local Printing Works

Figure 47-6 on page 905 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; no network printing software is used.

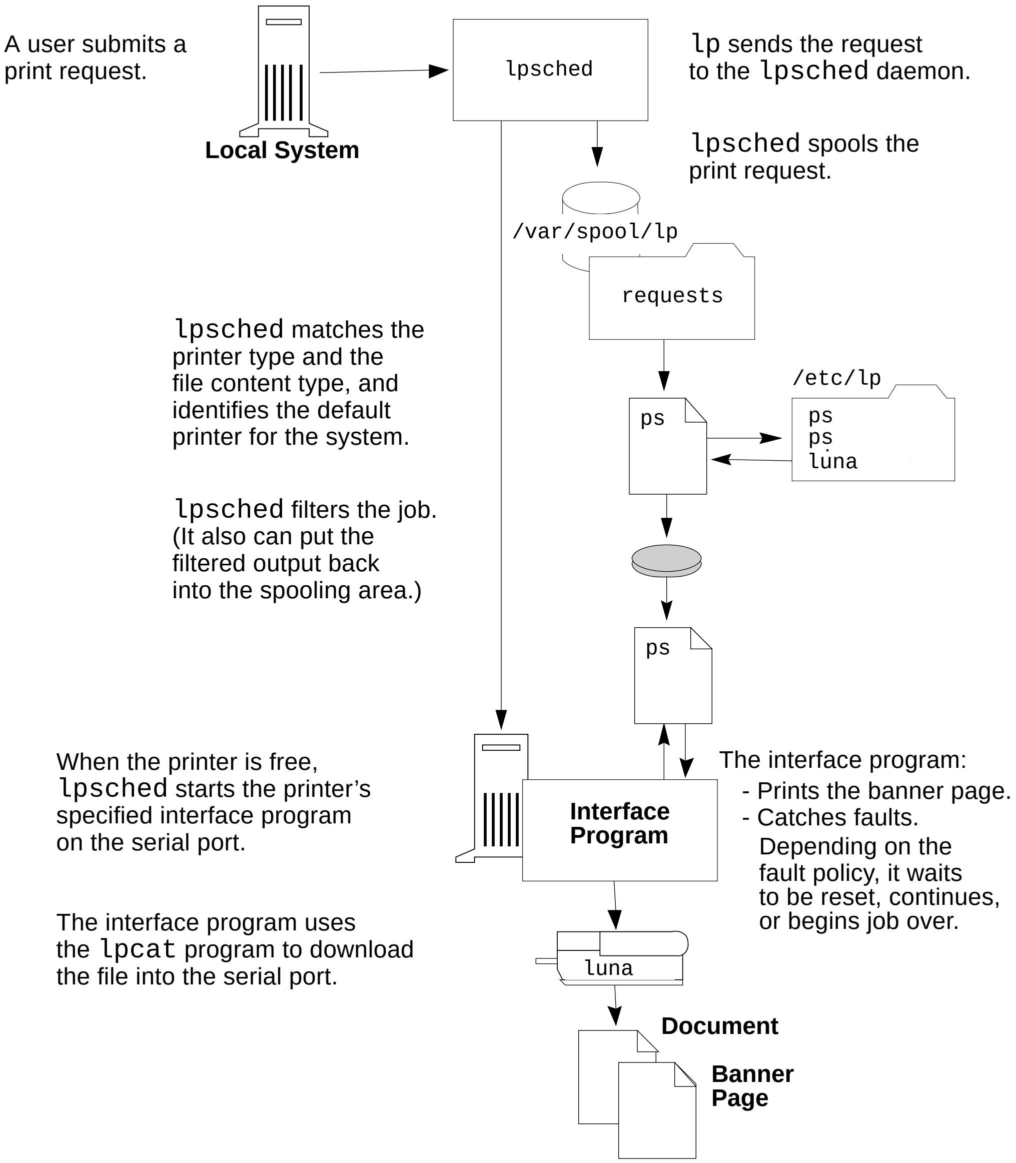
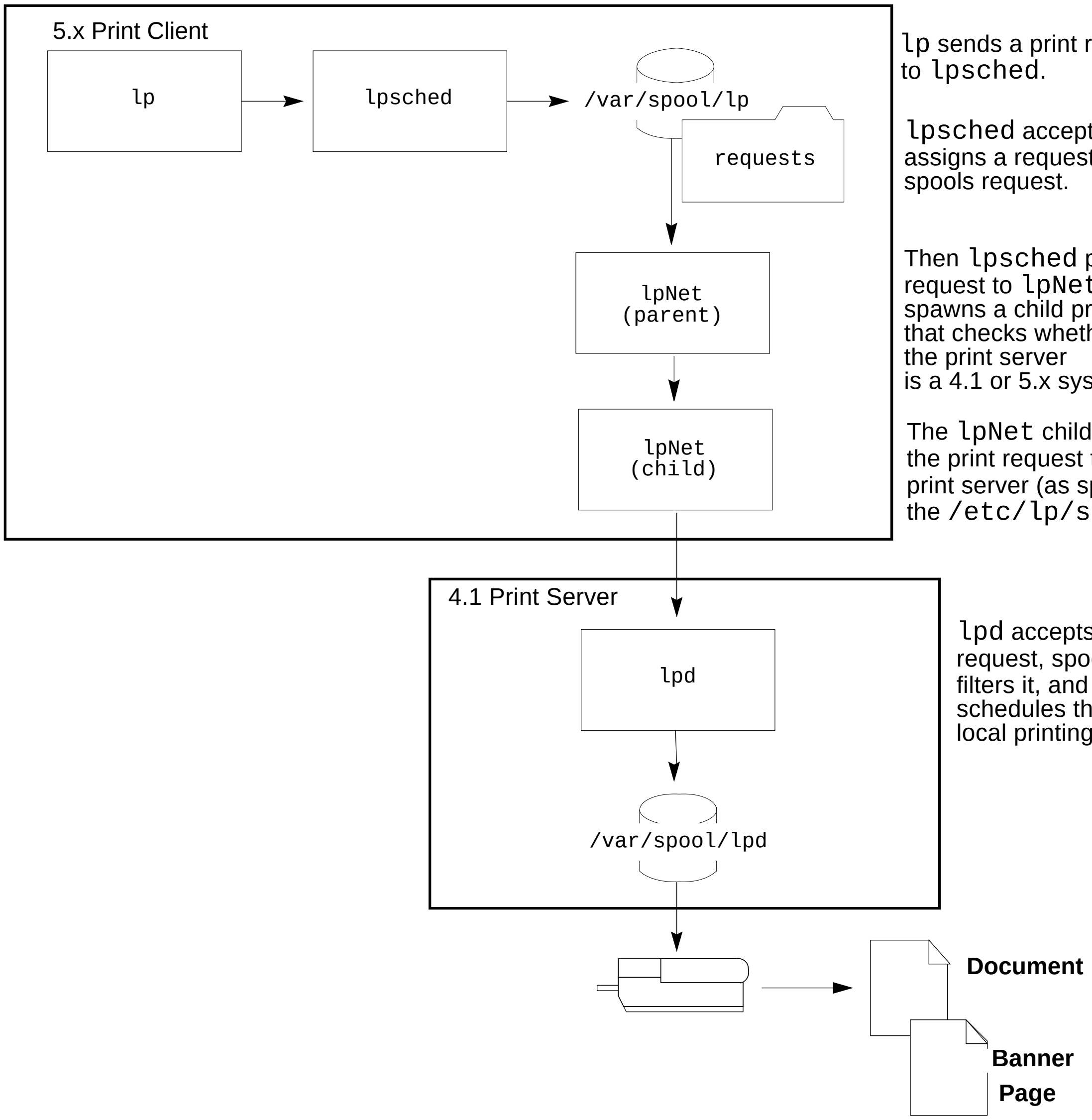

A user submits a print request.
Local System
lpsched
lp sends the request to the lpsched daemon.
lpsched spools the print request.
/var/spool/lp
requests
lpsched matches the printer type and the file content type, and identifies the default printer for the system.
/etc/lp
ps
ps
ps
luna
lpsched filters the job. (It also can put the filtered output back into the spooling area.)
ps
When the printer is free, lpsched starts the printer's specified interface program on the serial port.
Interface Program
The interface program:
- Prints the banner page.
- Catches faults.
Depending on the fault policy, it waits to be reset, continues, or begins job over.
The interface program uses the lpcat program to download the file into the serial port.
luna
Document
Banner Page

Figure 47-6 The Local Printing Process

How Remote Printing Works

Figure 47-7 on page 907 shows what happens when a user on a SunOS 5.x print client submits a print request to a SunOS 4.1 print server. The `lpsched` daemon handles the local part of the print request, and the `lpNet` daemon and its child process handle the network communication between the two systems.



`lp` sends a print request to `lpsched`.

`lpsched` accepts the request assigns a request ID, and spools request.

Then `lpsched` passes the request to `lpNet`. `lpNet` spawns a child process that checks whether the print server is a 4.1 or 5.x system.

The `lpNet` child transmits the print request to the print server (as specified in the `/etc/lp/system` file).

`lpd` accepts the request, spools it, filters it, and schedules the local printing.

Figure 47-7 Network Printing Between a SunOS 5.x Print Client and a SunOS 4.1 Print Server

Figure 47-8 on page 909 shows a SunOS 4.1 print client submitting a print request to a SunOS 5.x print server. The lpd daemon handles the local part of the print request and the connection to the print server. The network listen process, which resides in the Service Access Facility on the server, waits for network printing requests and sends them to the lpNet daemon. The lpNet daemon and its child processes hand the request over to the lpsched daemon, which processes the request on the print server.

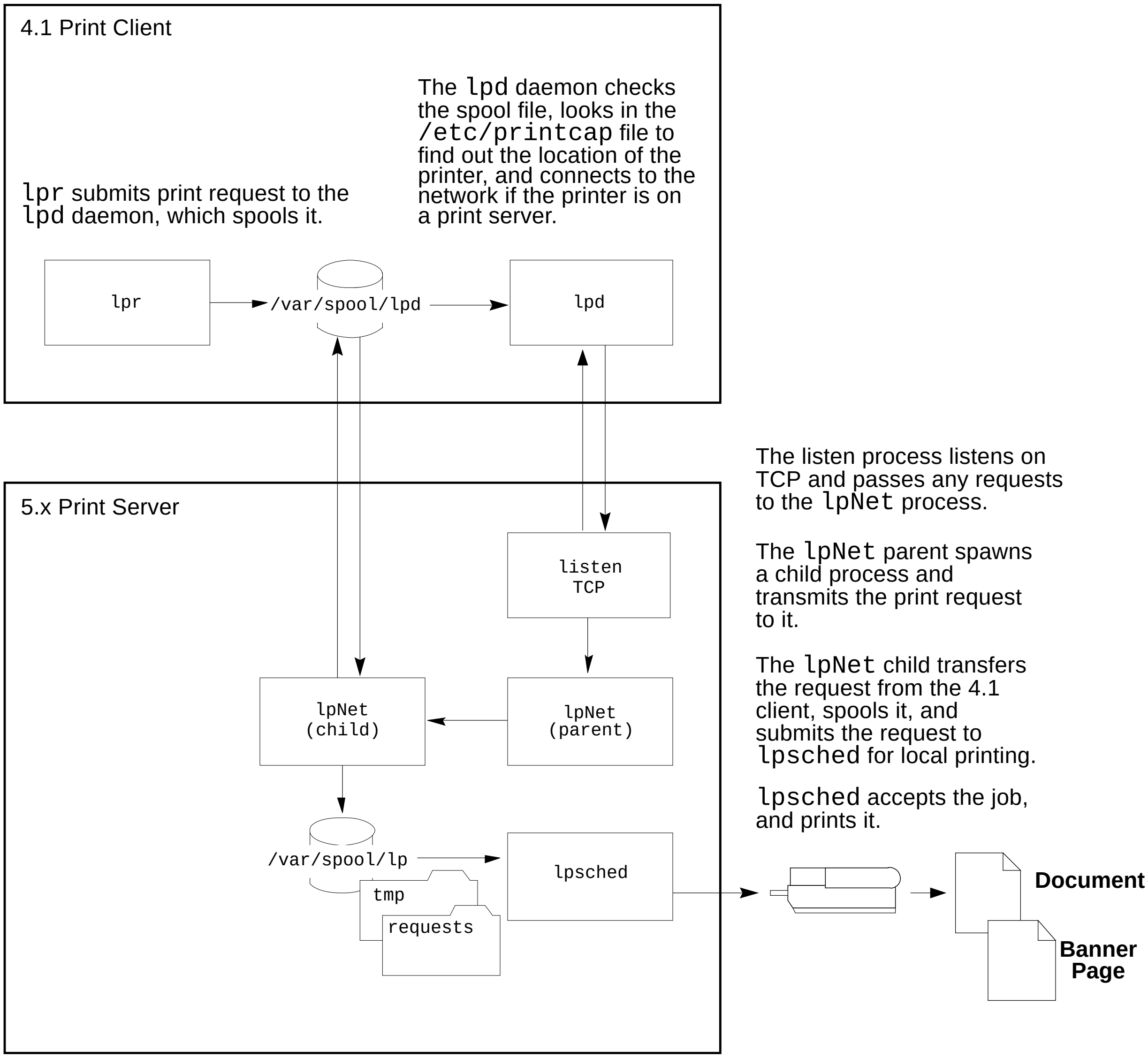

4.1 Print Client
lpr submits print request to the lpd daemon, which spools it.
The lpd daemon checks the spool file, looks in the /etc/printcap file to find out the location of the printer, and connects to the network if the printer is on a print server.
lpr
/var/spool/lpd
lpd
5.x Print Server
The listen process listens on TCP and passes any requests to the lpNet process.
The lpNet parent spawns a child process and transmits the print request to it.
The lpNet child transfers the request from the 4.1 client, spools it, and submits the request to lpsched for local printing.
lpsched accepts the job, and prints it.
listen TCP
lpNet (child)
lpNet (parent)
/var/spool/lp
tmp
requests
lpsched
Document
Banner Page

Figure 47-8 Network Printing Between a SunOS 4.1 Print Client and a SunOS 5.x Print Server

Figure 47-9 on page 911 shows what happens when a user of a SunOS 5.x print client submits a print request to a SunOS 5.x print server. The `lpsched` daemon on the print client handles the local part of each print request. Then `lpsched` passes the request to the `lpNet` daemon on the print client, which spawns a child process that communicates with the print server.

The Service Access Facility network listen service on the print server monitors network printing requests and sends them to the `lpNet` daemon on the print server. The `lpNet` daemon and its child processes send the request to the `lpsched` daemon on the print server, which processes the print request.

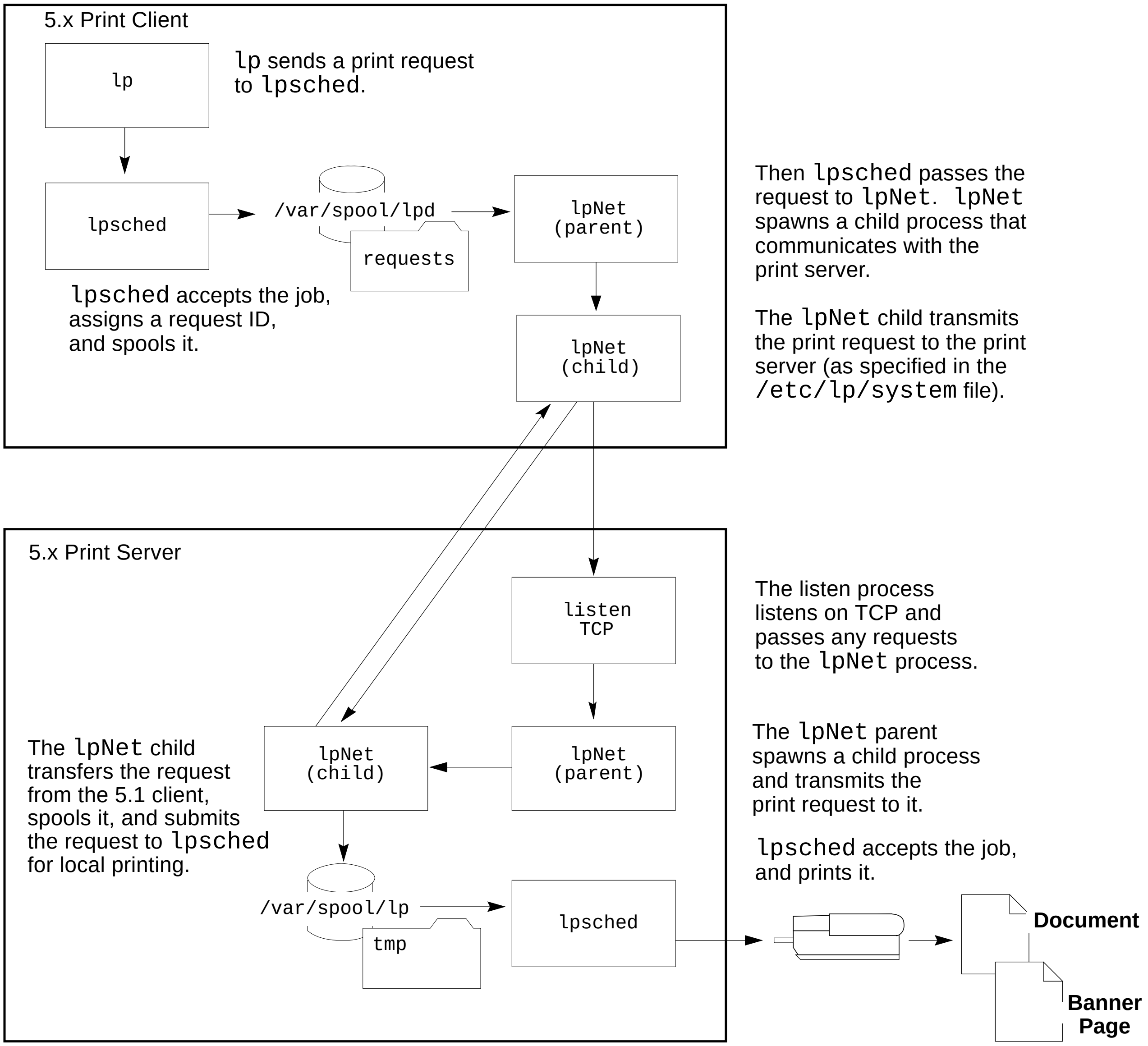

5.x Print Client
lp
lp sends a print request to lpsched.
lpsched
/var/spool/lpd
requests
lpNet (parent)
lpsched accepts the job, assigns a request ID, and spools it.
lpNet (child)
Then lpsched passes the request to lpNet. lpNet spawns a child process that communicates with the print server.
The lpNet child transmits the print request to the print server (as specified in the /etc/lp/system file).
5.x Print Server
listen TCP
The listen process listens on TCP and passes any requests to the lpNet process.
The lpNet child transfers the request from the 5.1 client, spools it, and submits the request to lpsched for local printing.
lpNet (child)
lpNet (parent)
The lpNet parent spawns a child process and transmits the print request to it.
lpsched accepts the job, and prints it.
/var/spool/lp
tmp
lpsched
Document
Banner Page

Figure 47-9 Network Printing Between a SunOS 5.x Print Client and a SunOS 5.x Print Server

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 may have a printing request that is not accommodated by the standard features of the LP print service. Or you may 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 51, "Customizing the LP Print Service," for detailed descriptions and step-by-step instructions to customize the LP print service.

Setting Up Printers 48

This chapter explains how to set up a printer and make it accessible to systems on the network. You can perform most printer setup tasks by using Admintool; however, you can add a network printer only from the command line.

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

How to Start Admintool	*page 915*
How to Add a Local Printer	*page 917*
How to Add Access to a Remote Printer	*page 919*
How to Add a Network Printer	*page 921*

For overview information about printers, see Chapter 47, “Overview of Print Management.”

Note – The SunSoft print client software and the Printer Manager application in Solstice AdminSuite offer the best solution for setting up and managing printers on a network. The advantage of the SunSoft print client software is that it supports a name service (NIS or NIS+), which enables you to centralize print administration for a network.

You can also set up a printer by using the LP print service commands instead of Admintool, although the manual process is more complicated. See “Setting Up a Printer With the LP Print Service Commands” on page 923 for detailed examples.

48

Setting Up Printing

Table 48-1 provides an overview of the tasks necessary to set up local and network printers and print clients.

Table 48-1 Task Map: Setting Up Printing

Activity	Description	For Examples, Go To	
Add a Printer	**Local Printer** After physically attaching the printer to a system, you must use Admintool to make the printer available for printing. The system to which the printer is connected becomes a print server.	▼ How to Add a Local Printer	page 917
	Network Printer After physically connecting the printer to the network, you must add a vendor-supplied SVR4 printer interface program or printing program to a system that becomes a print server for the printer.	▼ How to Add a Network Printer	page 921
Set Up Fault Recovery	Optional. Admintool does not enable you to set up how a printer should recover after it faults. By default, a printer tries to continue printing at the top of the page where printing stopped.	▼ How to Set Printer Fault Recovery	page 944
Set Up Fault Alerts	Optional. You can set up more specific fault alerts for the printer than Admintool provides.	▼ How to Set Fault Alerts for a Printer	page 945
Turn Off Banner Pages	Optional. You can turn off banner pages so a banner page is never printed on the printer. Admintool only enables you to make banner pages optional.	▼ How to Turn Off Banner Pages	page 942
Limit Access to the Printer	Optional. Admintool enables you to set up an allow list, but if you want to limit a few users' access to the printer, you may want to set up a deny list.	▼ How to Limit User Access to a Printer	page 947

Activity	Description	For Instructions, Go To
Add Access to the Printer	You must individually configure each system on the network to access the new printer. The systems become print clients for that printer.	▼ How to Add Access to a Remote Printer page 919

▼ How to Start Admintool

1. **Verify that the following prerequisites are met. To use the Admintool software, 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.
   - OpenWindows software. Start this software with the following command:

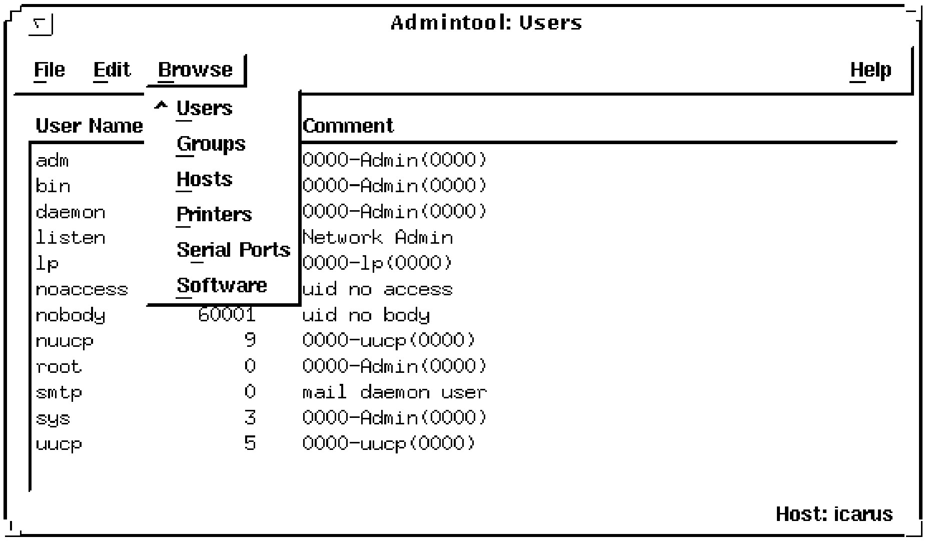
   ```
   $ /usr/openwin/bin/openwin
   ```

   - Membership in the `sysadmin` group (group 14).
2. **Log in on the system where you want to set up the printer.**
3. **Start Admintool with the following command:**

   ```
   $ admintool &
   ```

   The Admintool main window is displayed.

4. **Select Printers from the Browse menu.**



User Name		Comment
adm		0000-Admin(0000)
bin		0000-Admin(0000)
daemon		0000-Admin(0000)
listen		Network Admin
lp		0000-lp(0000)
noaccess		uid no access
nobody	60001	uid no body
nuucp	9	0000-uucp(0000)
root	0	0000-Admin(0000)
smtp	0	mail daemon user
sys	3	0000-Admin(0000)
uucp	5	0000-uucp(0000)

The Printers window is displayed.

Setting Up a Print Server

When you add a local printer to a system through Admintool, the printer is made accessible to the local system. The system on which you install the printer becomes the *print server.*

▼ How to Add a Local Printer

1. **Connect the printer to a system and turn on the power to the printer.**
   Consult the printer vendor’s installation documentation for information about the hardware switches and cabling requirements.

2. **Start Admintool on the system where you connected the printer.**
   See the procedure on page 915 for detailed information.

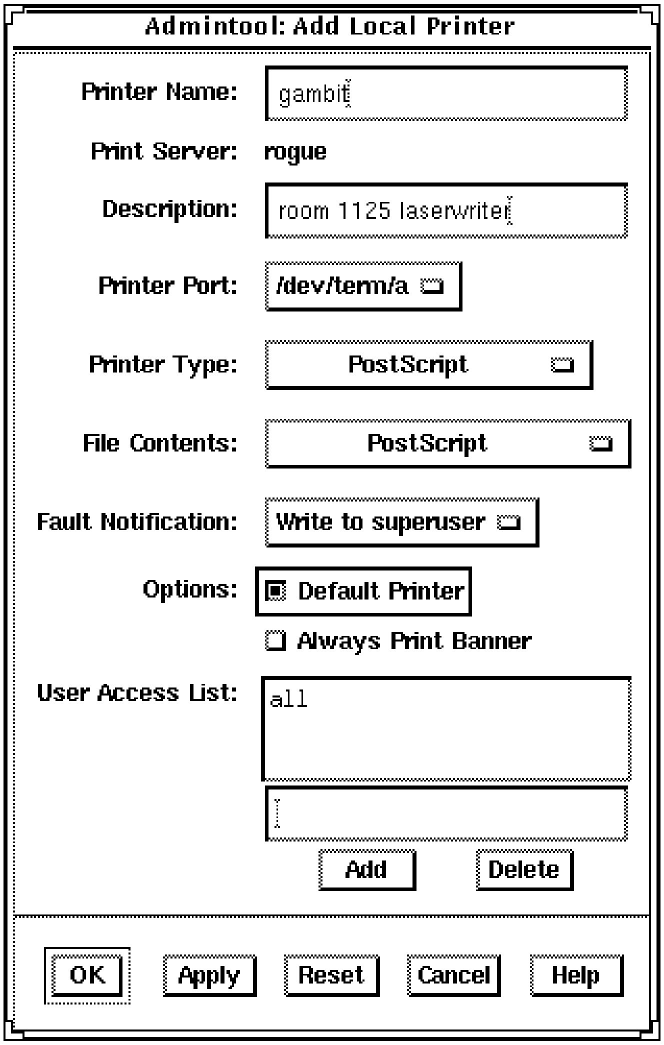
3. **Select Add Local Printer from the Edit menu.**
   The Add Local Printer window is displayed.

4. **Fill in the window.**
   If you need information to complete a field, click on the Help button to see field definitions for this window.

5. **Click on OK.**
   The printer is displayed in the Admintool Printers window. The printer is entered in the `/etc/lp/printers` directory of the print server.

Example—Completed Add Local Printer Window

In the following example, the printer `gambit` is added on the print server `rogue`.



Admintool: Add Local Printer

Printer Name: gambit

Print Server: rogue

Description: room 1125 laserwriter

Printer Port: /dev/term/a

Printer Type: PostScript

File Contents: PostScript

Fault Notification: Write to superuser

Options: ☒ Default Printer

□ Always Print Banner

User Access List: all

Setting Up a Print Client

When you give a system access to a remote printer through Admintool™, it makes that system a *print client*. A remote printer is any printer that is installed on a print server.

▼ How to Add Access to a Remote Printer

1. **Start Admintool on the system where you want to add access to a remote printer.**
   See the procedure on page 915 for detailed information.

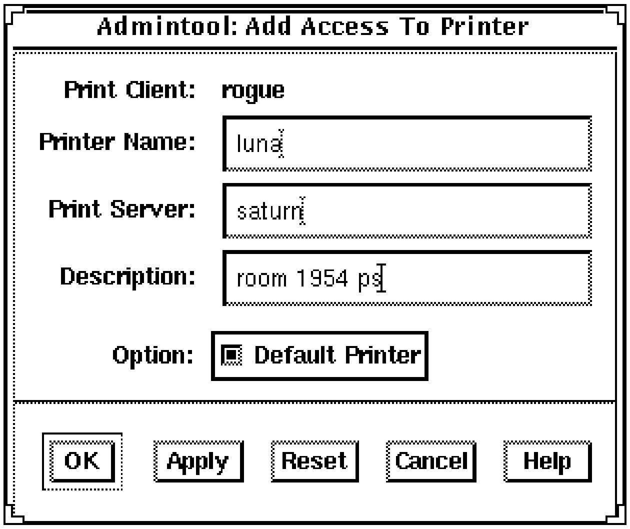
2. **Select Add Access to Remote Printer from the Edit menu.**
   The Add Access to Remote Printer window is displayed.

3. **Fill in the window.**
   If you need information to complete a field, click on the Help button to see field definitions for this window.

4. **Click on OK.**
   The printer is displayed in the Admintool Printers window. The printer is entered in the client's `/etc/lp/printers` directory.

Example—Completed Add Access To Printer Window

In the following example, the print client `rogue` is given access to the printer `luna` on the print server `saturn`.



Admintool: Add Access To Printer

Print Client: rogue

Printer Name: luna

Print Server: saturn

Description: room 1954 ps

Option: ■ Default Printer

Adding a Network Printer

A *network printer* is a hardware device that provides printing services to print clients without being connected to a print server. It has its own system name and IP address, and is connected directly to the network. Even though a network printer is not connected to a print server, it is a good idea to set up a print server for it. A print server provides queuing capabilities and printing administration for the network printer.

Network printers provide one or more special protocols that require a vendor-supplied printing program. The procedures to set up the vendor-supplied printing program can vary.

The vendor might supply an SVR4 printer interface script to replace the standard printer interface script. If so, the 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.

▼ How to Add a Network Printer

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.

2. **Follow the printer vendor instructions to add the network printer to a SunOS 5.x system that has an SVR4 LP print spooler.**
   If you are adding a network printer to a system that is not set up as a print server, see "Setting Up a Printer With the LP Print Service Commands" on page 923 in addition to the vendor instructions.

3. **See the following table to find your next step in this procedure.**
   Your next step depends on whether the network printer has an SVR4 printer interface script.

If the Printer ...	Then ...
Has an SVR4 interface script	Go to Step 4.
Does not have an SVR4 interface script	Go to Step 6.

4. **Set up the printer server to use the SVR4 printer interface script to call the vendor-supplied printing program.**

```
# lpadmin -p printer-name -i vendor-interface
```

In this command,

printer-name Is the name of the network printer.

vendor-interface Is the name of the vendor-supplied SVR4 interface script for the network printer.

5. **Go to “Where to Go From Here” on page 923.**
   Step 6 applies only when the printer has no vendor-supplied SVR4 interface script.

6. **Change the `FILTER` variable in the standard interface script, `/etc/lp/interfaces/`*printer-name*, to include the name of the vendor-supplied printing program instead of the standard printing program.**
   If your printer type is PS (PostScript), change the following `FILTER` variable.

   ```
   FILTER="/usr/lib/lp/postscript/postio"
   ```

   If your printer type is PSR (Reverse PostScript), change the following `FILTER` variable.

   ```
   FILTER="/usr/lib/lp/postscript/postreverse | \
      /usr/lib/lp/postscript/postio"
   ```

   If the printer is non-PostScript, change the following `FILTER` variable.

   ```
   FILTER="${LPCAT} 0"
   ```

   The standard interface script will use the value of the `FILTER` variable in the following line as the program to run to send print jobs to the device.

   ```
   0<${file} eval ${FILTER} 2>&1 1>&3
   ```

   The vendor-supplied printing program can be placed in the `/etc/lp/interfaces` directory, and it can be called from the standard interface script using that path name.

Verification—Adding a Network Printer

Submit a print request to the network printer from any system on the network and check for output.

Where to Go From Here

There are several optional tasks you may want to complete when setting up a network printer. See “Task Map: Setting Up Printing” on page 914 for pointers to the remaining tasks.

Setting Up a Printer With the LP Print Service Commands

Setting up a printer using the LP print service commands is complicated and error-prone. Admintool offers an easier and more reliable solution. Through Admintool, you just supply the information and let the tool run the required commands.

This section does not include detailed procedures on how to set up a printer using the LP print service commands, but it does provide annotated examples. The examples show the sequence of commands needed to set up a printer. You should use the examples in this chapter only when you have special needs that require the command-line approach—for example, when you need to write scripts to perform batch setup.

Regardless of the method, the decisions you make and information you need to supply during printer setup are similar, whether you use Admintool or the LP print service commands. Consequently, most of the conceptual and reference information you may need is described in “Administering Character Sets, Filters, Forms, and Fonts” on page 888 and “The Structure of the LP Print Service” on page 889.

Table 48-2 shows the high-level steps to set up a printer with the LP print service commands.

Table 48-2 Steps Required to Set Up a Printer by Using Commands

Activity	Description	For Annotated Examples, Go To	
Initialize System as Print Server	For a system to manage one or more printers, you must initialize the system as a print server. This includes configuring a systems’s port monitor and registering the network listening service.	▼ Example—Initializing a Print Server	page 925
Add a Local Printer	After you physically connect a local printer to the print server, you need to make the printer available for printing.	▼ Example—Adding a Local Printer	page 927
Add Access to the Printer	You must individually configure each system on the network to access the new printer. The systems become print clients for that printer.	▼ Example—Adding Access to a Remote Printer	page 928

Example—Initializing a Print Server

Initializing a system as a print server includes:

- Configuring the port monitor.
- Registering the network listen service.

This example shows how to use the sacadm and pmadm commands to initialize a print server.

```
❶ # sacadm -a -p tcp -t listen -c "/usr/lib/saf/listen tcp" -v ‘nlsadmin -V‘ -n 999
❷  # pmadm -a -p tcp -s lp -i root -m ‘nlsadmin -o \
   > /var/spool/lp/fifos/listenS5‘ -v ‘nlsadmin -V`
   # u_addr=‘lpsystem -A‘
❸  # pmadm -a -p tcp -s lpd -i root -m ‘nlsadmin -o /var/spool/lp/fifos/listenBSD -A \
   > "\\x${u_addr}"` -v ‘nlsadmin -V‘
   # new_addr=‘lpsystem -A | cut -b1-4‘
   # tail=‘lpsystem -A | awk '{pos = index($0, "0203")+4
   > print substr($0, pos, length($0)-pos+1)}'‘
   # new_addr=‘echo ${new_addr}0ACE${tail}‘
❹  # pmadm -a -p tcp -s 0 -i root -m ‘nlsadmin -c /usr/lib/saf/nlps_server -A \
   > "\\x${new_addr}"` -v ‘nlsadmin -V‘
❺  # cat /var/saf/tcp/log
   10/28/91 10:22:51; 178; @(#)listen:listen.c     1.19.9.1
   10/28/91 10:22:51; 178; Listener port monitor tag: tcp
   10/28/91 10:22:51; 178; Starting state: ENABLED
   10/28/91 10:22:51; 178; Service 0: fd 6 addr \ \x00020ACE000000000000000000000000
   10/28/91 10:22:51; 178; Service lpd: fd 7 addr \x00020203000000000000000000000000
   10/28/91 10:22:52; 178; Net opened, 2 addresses bound, 56 fds free
   10/28/91 10:22:52; 178; Initialization Complete
```

❶ Configures the port monitor to accept service requests.

❷ Registers the network System V listen service.

❸ Registers the network BSD listen service.

❹ Registers the network Service 0 listen service.

❺ Checks to make sure that the print services are enabled and initialized.

Configuring the Port Monitor

For print clients to access a print server, the port monitor on the print server must accept service requests and notify the LP print service of such requests. In addition, the port monitor on print clients must be running to receive messages from the server. See Chapter 56, “Setting Up Terminals and Modems With the Service Access Facility,” for a complete discussion of port monitors and the Service Access Facility.

Registering the Network Listen Service With the Port Monitor

The LP print service uses a connection-oriented protocol to establish connections for incoming requests from remote systems. When the port monitor is configured, the following listen services are registered:

- `listenBSD`
- `listenS5`
- `Service 0`

These services “listen” for print requests from print clients or confirmations from the server. When a communication is detected, the service hands over the process to the `lpNet` daemon.

To configure the network listen process to listen for print requests from other systems, you must register the *universal address* of the print server with the LP print service. The universal address is the Internet Protocol (IP) address in hexadecimal form.

You can obtain the universal address by using the `lpsystem -A` command. The universal address has four parts, as shown in Figure 48-1. The third part, the IP address, consists of zeros. The zeros represent a special IP address that refers to the local host. The last part, RFU, means Reserved for Future Use, and could be used for other families of addresses (for example, Open Systems Interface) in the future.

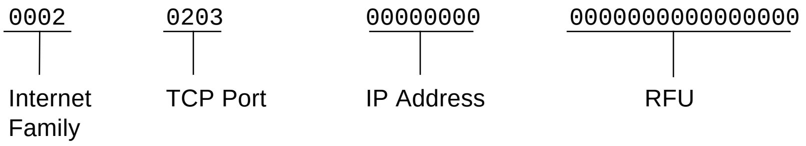


Figure 48-1 Parts of the Universal Address

Example—Adding a Local Printer

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 in the example and may change depending on your situation:

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

```
❶ # chown lp /dev/term/b
  # chmod 600 /dev/term/b
❷ # lpadmin -p luna -v /dev/term/b
❸ # lpadmin -p luna -T PS
❹ # lpadmin -p luna -I postscript,simple
  # cd /etc/lp/fd
❺ # for filter in *.fd;do
  > name=‘basename $filter .fd‘
  > lpfilter -f $name -F $filter
  > done
  # lpfilter -f postio -F postio.fd
  # lpfilter -f postior -F postior.fd
  # lpfilter -f postprint -F postprint.fd
  # lpfilter -f postreverse -F postreverse.fd
❻ # accept luna
  destination “luna” now accepting requests
  # enable luna
  printer “luna” now enabled
❼ # lpadmin -p luna -D "PostScript Laser printer in Building 5, Room 262"
❽ # lpadmin -d luna
❾ # lpstat -t
  scheduler is running
  system default destination: luna
  device for luna: /dev/term/b
  luna accepting requests since Mon Mar  4 14:37:55 PST 1995
  printer luna is idle. enabled since Mon Mar  4 14:37:59 PST 1995. available.
```

❶ Gives `lp` ownership and sole access to a port device.

❷ Defines the printer name and the port device the printer will use.

❸ Sets the printer type of the printer.

❹ Specifies the file content types to which the printer can print directly.

❺ Adds print filters to the print server.

❻ Accepts print requests for the printer and enables the printer.

❼ Adds a description for the printer.

❽ Specifies this printer as the system’s default printer destination.

❾ Verifies that the printer is ready.

Example—Adding Access to a Remote Printer

If a system needs to print to a remote printer, you must add access to the remote printer. This example shows how to configure a SunOS 5.x system to access a printer named `luna`, which is connected to the SunOS 5.x print server `terra`. The system becomes a print client of the printer luna.

```
❶ # lpsystem -t bsd terra
  "terra" has been added.
❷ # lpadmin -p luna -s terra -T unknown -I any
❸ # accept luna
  destination “luna” now accepting requests
  # enable luna
  printer “luna” now enabled
❹ # lpadmin -p luna -D "PostScript Laser printer in Building 5, Room 262"
❺ # lpadmin -d luna
❻ # lpstat -t
  scheduler is running
  system default destination: luna
  system for luna: terra
  luna accepting requests since Mon Mar  4 15:15:21 PST 1995
  printer luna is idle. enabled since Mon Mar  4 15:15:26 PST 1995. available.
```

❶ Identifies the print server system and its type (`bsd` for BSD or `s5` for System V).

❷ Identifies the printer on the printer server.

❸ Specifies that the print client can send print requests to the printer.

❹ Adds a description for the printer.

❺ Sets the printer as the system’s default printer destination.

❻ Verifies that the printer is ready.

Administering Printers 49

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	*page 930*
How to Check the Status of Printers	*page 933*
How to Stop the Print Scheduler	*page 936*
How to Restart the Print Scheduler	*page 936*
How to Add a Printer Description	*page 939*
How to Set a System's Default Printer	*page 940*
How to Make Banner Pages Optional	*page 941*
How to Turn Off Banner Pages	*page 942*
How to Define a Class of Printers	*page 943*
How to Set Printer Fault Recovery	*page 944*
How to Set Fault Alerts for a Printer	*page 945*
How to Limit User Access to a Printer	*page 947*
How to Check the Status of Print Requests	*page 949*
How to Accept or Reject Print Requests for a Printer	*page 951*
How to Enable or Disable a Printer	*page 953*
How to Cancel a Print Request	*page 954*
How to Cancel a Print Request From a Specific User	*page 956*
How to Move Print Requests to Another Printer	*page 957*
How to Change the Priority of a Print Request	*page 959*

For overview information about printing and the LP print service, see Chapter 47, “Overview of Print Management.”

Managing Printers and the Print Scheduler

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

▼ How to Delete a Printer and Remote Printer Access

1. **Log in as root 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
   ```

   In this command,

   printer-name Is the 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
   ```

   In this command,

   print-server Is the 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 1 through Step 3 on each print client that has access to the printer.**

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

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

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

In this command,

printer-name Is the name of the printer you want to delete.

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 951 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 953 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 957 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...]
```

In this command,

print-client Is the 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.

Verification—Deleting a Printer and Remote Printer Access

On the print client, make sure that the information for the printer has been deleted. You should receive an error indicating that the printer does not exist in the output of the following command.

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

On the print server, make sure that the configuration information for the printer has been deleted. You should receive an error indicating that the printer does not exist in the output of the following command.

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

Verify that any print clients that will no longer use the print server have been deleted from the server. The print clients should not be listed in the output of the following command.

```
print-server# lpsystem -l
```

Otherwise, submit a print request to the printer from a deleted print client. You should receive an error message stating that the destination printer is unknown to the LP print service.

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”.
jupiter# lpadmin -x luna
jupiter# lpsystem -r terra
Removed “terra”.
```

▼ 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 the `lpstat(1)` man page for other options.

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

In this command,

`-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 Wed Oct 12 10:28:33 MDT 1995. 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 Tue Nov  1 12:41:17 MST 1995. available.
    unable to print: paper misfeed jam

    Description: Printer by break room.
printer luna is idle. enabled since Tue Nov  1 12:41:17 MST 1995. available.
    Description:
```

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

```
$ lpstat -p luna -l
printer luna is idle. enabled since Tue Aug 30 11:05:33 MDT 1995. available.
    Content types: any
    Printer types: unknown
    Description:
    Users allowed:
    (all)
    Forms allowed:
    (none)
    Banner not required
    Character sets:
    (none)
    Default pitch:
    Default page size:
```

▼ How to Stop the Print Scheduler

1. **Log in as root 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 root 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 Print Definitions

See “Setting Definitions for Printers” on page 870 for more information on setting print definitions.

Table 49-1 lists the fault recovery values you can set for a printer with the `lpadmin -F` command. See “Fault Recovery” on page 882 for more information.

Table 49-1 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.

Table 49-2 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. See “Fault Notification” on page 878 for more information.

Table 49-2 Values for 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.

Table 49-2 Values for Alerts *(Continued)*

Value for `-A` *alert*	**Description**
`'`*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.

Table 49-3 lists the values you can add to an allow or deny list to limit user access to a printer. See "Limiting User Access to a Printer" on page 880 for more information.

Table 49-3 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
system`!all`	All users on *system*
`!all`	All users on local system

▼ How to Add a Printer Description

1. **Log in as root or lp on the print server.**
2. **Add a printer description by using the `lpadmin` command.**

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

In this command,

printer-name Is the name of the printer for which you are adding a description.

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.

Verification—Adding a Printer Description

Check the information following the `Description` heading in the output of the following command.

```
$ 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"
```

▼ How to Set a System’s Default Printer

1. **Log in as root 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]
```

In this command,

printer-name Is the 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.

Verification—Setting a System’s Default Printer

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
```

▼ How to Make Banner Pages Optional

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

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

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

In this command,

printer-name Is the name of the printer for which you are making banner pages optional.

`-o nobanner` 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` option.

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

Verification—Making Banner Pages Optional

The output from the following command should contain 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
```

▼ How to Turn Off Banner Pages

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

2. **Change directory to the** `/etc/lp/interfaces` **directory.**

```
# cd /etc/lp/interfaces
```

3. **Edit the file that has the name of the printer for which you want to turn off banner pages.**

4. **Change the** `nobanner` **variable to** `yes`**.**

```
nobanner="yes"
```

Change the `nobanner` variable to `no` if you want to turn banner pages on again.

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

Verification—Turning Off Banner Pages

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

▼ How to Define a Class of Printers

1. **Log in as root or lp on the print server.**
2. **Define a class of printers by using the `lpadmin` command.**

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

In this command,

printer-name Is the name of the printer you are adding to a class of printers.

printer-class Is the 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.

Verification—Defining a Class of Printers

List 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
```

▼ How to Set Printer Fault Recovery

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

2. **Set up fault recovery for the printer with the `lpadmin` command.**

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

In this command,

printer-name Is the name of the printer for which you are specifying fault recovery.

recovery-options Is one of the three valid recovery options: `beginning`, `continue`, or `wait`.

See Table 49-1 on page 937 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.

Verification—Setting Printer Fault Recovery

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
```

▼ How to Set Fault Alerts for a Printer

1. **Log in as root 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]
```

In this command,

printer-name Is the name of the printer for which you are specifying an alert for printer faults.

alert Specifies what kind of alert will occur when the printer faults. See Table 49-2 on page 937 for detailed information about the valid values for *alert*. Some valid values are `mail`, `write`, and `quiet`.

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.

Verification—Setting Fault Alerts for a Printer

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
```

▼ How to Limit User Access to a Printer

1. **Log in as root 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
```

In this command,

printer-name Is the name of the printer to which the allow or deny user access list applies.

user-list Represents user names to be added to the allow or 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 49-3 on page 938 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
```

Verification—Limiting User Access to a Printer

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

If you need overview information on managing print requests, see "Managing Print Requests" on page 884.

Table 49-4 lists the values for changing the priority of a print request with the `lp -H` command. See "Changing the Priority of Print Requests" on page 885 if you need further information.

Table 49-4 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 the `lpstat(1)` man page for other valid options.

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

In this command,

`-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    Mar 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   Jan 14 09:07
moon-79    root     1024   Jan 14 09:08
```

▼ How to Accept or Reject Print Requests for a Printer

1. **Log in as root or lp on the print server.**
2. **Stop accepting print requests for the printer by using the** `reject` **command.**

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

In this command,

`-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 Is the 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 953.

3. **Start accepting print requests for the printer by using the** `accept` **command.**

```
# accept printer-name
```

Verification—Accepting or Rejecting Print Requests for a Printer

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
```

▼ How to Enable or Disable a Printer

1. **Log in as root 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
```

In this command,

`disable` With no options, 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 Is the 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
```

Verification—Enabling or Disabling a Printer

```
$ 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
```

▼ How to Cancel a Print Request

1. **If you are going to cancel print requests of other users, become root 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 949 for more details.

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

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

In this command,

request-id Is the 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.

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. **If you are going to cancel print requests of other users, become root or lp.**
2. **Cancel a print request from a specific user with the `cancel` command.**

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

In this command,

`-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—Cancelling 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
```

▼ How to Move Print Requests to Another Printer

1. **Log in as root or lp on the print server.**
2. **(Optional) Check the request IDs of the print requests on the original printer.**

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

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

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

   In this command,

   printer-name2 Is the 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
```

In this command,

printer-name1 Is the name of the printer from which all print requests will be moved.

printer-name2 Is the 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
```

Verification—Moving Print Requests to Another Printer

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
```

▼ How to Change the Priority of a Print Request

1. **Log in as root 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 949 for more information.
3. **Change the priority of a print request by using the `lp` command.**

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

In this command,

request-id Is the 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.

change-priority Is one of the three ways to change the priority of a print request: `hold`, `resume`, `immediate`.

See Table 49-4 on page 949 for detailed information about valid values for *change-priority*.

You can also use the -q option of the lp 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
```

Managing Character Sets, Filters, Forms, and Fonts 50

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	*page 965*
How to Unmount and Mount a Print Wheel or Font Cartridge	*page 966*
How to Set an Alert to Mount a Print Wheel or Font Cartridge	*page 968*
How to Set Up an Alias for a Selectable Character Set	*page 970*
How to Add a Print Filter	*page 975*
How to Delete a Print Filter	*page 976*
How to View Information About a Print Filter	*page 977*
How to Add a Form	*page 981*
How to Delete a Form	*page 982*
How to Unmount and Mount a Form	*page 983*
How to Set an Alert to Mount a Form	*page 986*
How to View Information About a Form	*page 988*
How to View the Current Status of a Form	*page 989*
How to Limit User Access to a Form	*page 990*
How to Limit Printer Access to a Form	*page 991*
How to Install Downloaded PostScript Fonts	*page 996*
How to Install Host-Resident PostScript Fonts	*page 996*

For overview information about printing, see Chapter 47, “Overview of Print Management.”

Managing Character Sets

Printers differ in the method they use to print text in various font styles. For instance, 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 may have defined 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 csnm n
```

The `csnm` 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 may 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 992 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.

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, may not be the same character set.

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 “Fault Notification” on page 878 for general information about alerts.

▼ How to Define a Print Wheel or Font Cartridge

1. **Log in as root 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...]
```

In this command,

printer-name Is the name of the printer for which you are defining a print wheel or font cartridge.

hard-charset Is the 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 root 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.

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 and Step 4 for each print client that may need to use the print wheel or font cartridge.**

Verification—Defining a Print Wheel

On both the print server and the print client, check the information following the `Character sets` heading in the output of the following command:

```
$ 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 root 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
```

In this command,

printer-name Is the name of the printer on which you are unmounting a print wheel or font cartridge.

`-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
   ```

   In this command,

   printer-name Is the printer on which you are mounting a print wheel or font cartridge.

   hard-charset Is the 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.

Verification—Unmounting and Mounting a Print Wheel

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 root or lp on the print server.**

2. **Set an alert to mount a print wheel or font cartridge by using the** `lpadmin` **command.**

```
# lpadmin -S hard-charset -A alert [-Q requests] [-W minutes]
```

In this command,

hard-charset Is the hardware character set name of the print wheel or font cartridge for which you want to set an alert.

alert Specifies what kind of alert will occur when a print wheel or font cartridge is requested. See Table 49-2 on page 937 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.

requests Specifies the number of print requests that require the print wheel or font cartridge 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.

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.

Verification—Setting an Alert to Mount a Print Wheel or Font Cartridge

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 10 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 names for the selectable character sets are adequate. See "Managing Print Requests" on page 884 for more information.

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

2. **Display the names of the selectable character sets for the specified printer type by using the** tput **command.**

   ```
   # tput -T printer-type csnm n
   ```

   In this command,

   printer-type Is a printer type found in the terminfo database. See "Printer Type" on page 873 for information on entries in the terminfo database.

   n Is a 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, csnm 1 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...]
```

In this command,

printer-name Is the printer on which you are setting up aliases for selectable character sets.

select-charset Is the selectable character set name for which to set an alias. The name can be found in Step 2.

alias Is the 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 root 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.

The alias is added in the print client's `/etc/lp/printers/`*printer-name*`/configuration` file.

6. **Repeat Step 4 and Step 5 for each print client that may need to use the alias.**

Verification—Setting Up an Alias for a Selectable Character Set

On the print server and print clients, verify that the alias for the selectable character set is listed in the output of the following command.

```
$ 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 csnm 0
usascii# tput -T ln03 csnm 1
english# tput -T ln03 csnm 2
finnish# tput -T ln03 csnm 3
japanese# tput -T ln03 csnm 4
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 50-1 on page 974. 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. Admintool automatically enables the supplied filters. However, if you administer other printers, you may 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 1011 for background information and step-by-step instructions.

Adding, Changing, Removing, and Restoring Print Filters

Print filters are added, changed, or removed on both the print server and the print clients.

You use the `lpfilter` 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 may 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 may 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 5.x system software provides a default set of PostScript filters, which Admintool automatically adds to a print server. Some of the TranScript filters used with SunOS 4.1 have SunOS 5.x equivalents, but others do not. Table 50-1 lists the default PostScript filters and identifies the TranScript filters, where applicable.

Table 50-1 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
postprint	simple to PostScript	enscript
postreverse	Reverse or select pages	psrev
posttek	TEK4014 to PostScript	ps4014

SunOS 5.x does not provide the following filters:

- TEX
- `oscat` (NeWSprint `opost`)
- Enscript

The postreverse, postprint, postio, and dpost filters are provided in place of Enscript.

Admintool 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 1022 for information about writing a custom print filter.

How to Add a Print Filter

1. **Log in as root 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
   ```

   In this command,

   filter-name Is a name you choose for the print filter.

   filter-def Is name of the print filter definition.

   The print filter is added in the print server’s `/etc/lp/filter.table` file.

Verification—Adding a Print Filter

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 root or lp on the print server.**
2. **Delete the print filter by using the `lpfilter` command.**

```
# lpfilter -f filter-name -x
```

In this command,

filter-name Is the name of the print filter to be deleted.

The print filter is deleted from the print server's `/etc/lp/filter.table` file.

Verification—Deleting a Print Filter

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 root or lp on the print server.**
2. **Request information about a print filter by using the** `lpfilter` **command.**

```
# lpfilter -f filter-name -l
```

In this command,

filter-name Is the print filter for which you want to view information. Specify `all` for *filter-name* to view information about all the available print filters.

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
(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 “How to Create a New Form Definition” on page 1027 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 may 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. Then 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.

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 “Fault Notification” on page 878 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.

- Use the `lpforms` command to show the attributes of the form. You can also redirect the output of the command into a file to save it for future reference.
- Use the `lpstat` command to display the current status of the form. 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 may want only the people in the payroll or accounts payable department to be able to print check forms. In addition, you may 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 990. To limit printer access to a form, see "How to Limit Printer Access to a Form" on page 991.

▼ How to Add a Form

1. **Log in as root 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
```

In this command,

form-name	Is the name you choose for the form.
`/etc/lp/forms/`*form*	Is the name of the form definition.

The form is added in the print server’s `/etc/lp/forms/`*form-name*`/describe` file.

Verification—Adding a Form

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 granted access to the form. See “How to Limit Printer Access to a Form” on page 991.

How to Delete a Form

1. **Log in as root or lp on the print server.**
2. **Delete the form by using the `lpforms` command.**

```
# lpforms -f form-name -x
```

In this command,

form-name Is the form to be deleted.

The form is deleted from `/etc/lp/forms/`*form-name* file.

Verification—Deleting a Form

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 root 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
```

In this command,

printer-name Is the name of the printer on which you are unmounting a form.

New print requests (which may 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.

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]
```

In this command,

printer-name Is the printer on which you are mounting a form.

form-name Is the 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.

Verification—Unmounting and Mounting a Form

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 root 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]
```

In this command,

form-name Is the form for which you want to set a request alert.

alert Specifies what kind of alert will occur when a form is requested. See Table 49-2 on page 937 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.

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.

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.

Verification—Setting an Alert to Mount a Form

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 root or lp on the print server.**
2. **Request information about a form by using the `lpforms` command.**

```
# lpforms -f form-name -l
```

In this command,

`-f` *form-name* Is the form for which you want to view information. Specify `all` for *form-name* to view information about all the available forms.

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` command.**

```
$ lpstat -f form-name
```

In this command,

`-f` *form-name* Is the 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,payroll
form medical is available to you
```

▼ How to Limit User Access to a Form

1. **Log in as root 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
```

In this command,

form-name — Is the name of the form for which the allow or deny user access list is being created.

user-list — Represents users to be added to the allow or 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 49-3 on page 938 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
/etc/lp/forms/form-name/deny
```

Verification—Limiting User Access to a Form

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 root 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
```

In this command,

printer-name Is the name of the printer for which the allow or deny forms list is being created.

form-list Are 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`

Verification—Limiting Printer Access to a Form

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 payroll"
```

Managing Fonts

If you have a laser printer, you may need to install and maintain PostScript fonts. You may 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 may need to 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 may 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 may be a print server or a print client. Each user may 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 may 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` 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 may also be used to send fonts to a print server. To do this, you may create a new filter table entry that calls the download filter as a slow filter by using the `-y` option of the `lp` command. Alternatively, you may 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 root 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
```

In this command,

printer-name Is the name of the printer on which you want to install downloaded PostScript fonts.

3. **Create the** `residentfonts` **file, if it does not already exist.**
   This file may not exist if this is the first time you are adding permanently downloaded fonts.

4. **Edit the** `residentfonts` **file by adding all the printer-resident fonts and fonts to be permanently downloaded.**
   You can use any text editor, such as vi.

5. **Save the file.**

▼ How to Install Host-Resident PostScript Fonts

1. **Log in as root 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
# 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. **Type** `cd /usr/share/lib/hostfontdir` **and press Return.**

   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.**

Customizing the LP Print Service 51

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	*page 1001*
How to Add a terminfo Entry for an Unsupported Printer	*page 1005*
How to Set Up a Custom Printer Interface Program	*page 1010*
How to Create a New Print Filter	*page 1022*
How to Create a New Form Definition	*page 1027*

For overview information about printers, see Chapter 47, “Overview of Print Management.”

51

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.

Table 51-1 shows the default `stty` settings used by the LP print service.

Table 51-1 `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 root or lp on the print server.**
2. **Adjust the printer port characteristics by using the `lpadmin` command.**

```
# lpadmin -p printer-name -o "stty=options"
```

In this command,

printer-name
Is the 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 quotes and use a space to separate the options.

See `stty(1)` for a complete list of options. Table 51-1 on page 1000 shows the default `stty` settings used by the LP print service.

Verification—Adjusting the Printer Port Characteristics

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 5.x system are PS (for PostScript) and PSR (for PostScript Reverse).

If you cannot find a terminfo entry for your printer, you still may be able to use the printer with the LP print service without the automatic selection of page size, pitch, and character sets. However, you may 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. Table 51-2 lists the required terminfo items for a printer.

Table 51-2 Required terminfo Items for a Printer *(1 of 3)*

Item		**Meaning**
Booleans:		
	cpix	Changing character pitch changes resolution
	daisy	Printer requires an operator to change character set
	lpix	Changing line pitch changes resolution

Table 51-2 Required `terminfo` Items for a Printer *(2 of 3)*

Item		Meaning
Numbers:		
	`bufsx`	Number of bytes buffered before printing
	`cols`	Number of columns in a line
	`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
	`if`	Name of initialization file
	`iprog`	Path name of initialization program
	`is1`	Printer initialization string
	`is2`	Printer initialization string
	`is3`	Printer initialization string

Table 51-2 Required `terminfo` Items for a Printer *(3 of 3)*

Item	**Meaning**
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
`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 root or lp on the print server.**

1. **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.

2. **Create a `terminfo` entry file for the printer.**
   Table 51-2 on page 1002 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 the `terminfo(4)` man page.

   To help you start writing a new `terminfo` entry, use the `infocmp` command to save an existing `terminfo` entry to a file. This is very 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
   ```

3. **Compile the `terminfo` entry file into the `terminfo` database.**

   ```
   # tic terminfo_entry
   ```

   In this command,

terminfo_entry	Is the `terminfo` entry file you created.

Verification—Adding a terminfo Entry for an Unsupported Printer

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.

Caution – A customized printer interface program must not terminate the connection to the printer or "uninitialize" the printer in any way.

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.x LP print service. However, several `-o` options have been standardized in the SunOS 5.x LP print service and will be passed to every printer interface program. These options may 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 51-3.

Table 51-3 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.

Table 51-4 shows the exit codes and how they are interpreted by the LP print service.

Table 51-4 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:

```
# Here’s where we set up the $LPTELL program to capture fault
messages
```

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 the kill(1) and signal(3B) man pages.) 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 root or lp on the print server.**
2. **Find 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.
Already have a custom printer interface program	Go to Step 5.

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 1007 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
```

In this command,

printer-name Is the printer that will use the custom printer interface program.

custom-interface The 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.

Verification—Setting Up a Custom Printer Interface Program

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 SunOS 5.x 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 may 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 may 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 may want to 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 “Fault Notification” on page 878.

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 may 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 may 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
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 Table 51-5 are assigned by default. They are not very useful, which is why you should specify explicit values.

Table 51-5 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 may 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 may 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 may 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 may 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 may 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 may, however, have some printers that are or inappropriate for the output that the filter produces. For example, you may want to dedicate one printer for fast turnaround, only sending files that require no filtering to that printer. Other printers of identical type may 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 1012.

Slow filters that are invoked by printing modes (using the `-y` option of the `lp` 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 may 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 may include none, all, or any subset of the 13 keywords. In addition, a single keyword may be defined more than once, if multiple definitions are required for a complete filter definition. Table 51-6 contains descriptions of the 13 keywords available for defining `Options` in a print filter definition.

Table 51-6 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`

Table 51-6 Print Filter Options Keywords *(Continued)*

Characteristic	**Keyword**	**Possible Patterns**	**Example**
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:

keyword pattern = 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 -f option of the lp 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 `-y` option of the `lp` command. Because a user can specify several `-y` options, there may 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 root or lp on the print server.**
2. **Create a print filter program.**
   See “Writing a Print Filter Program” on page 1011 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 1015 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 975.

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 `–y` option to the `lp` 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 for two purposes:

- To initialize the printer so that printing is done properly on the form
- To 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
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 may 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 Table 51-7, the default values are assigned. Before running the `lpforms` command, gather the following information about the new form:

Table 51-7 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.
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 may 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.

Table 51-7 Default Form Values *(Continued)*

Item	Default	Description
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 root or lp on the print server.**
2. **Create a form definition file.**
   See "Creating a New Printer Form" on page 1025 for an overall 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 981.

Troubleshooting Printing Problems 52

This chapter explains how to troubleshoot printing problems that may occur when you set up or maintain printing services.

If you want to skip the background information that explains the concepts of troubleshooting printing problems and proceed directly to step-by-step instructions, use the following list to find the page where the instructions for a specific task begin.

How to Troubleshoot No Output	*page 1037*
How to Troubleshoot Incorrect Output	*page 1060*
How to Unhang the LP Commands	*page 1065*
How to Troubleshoot an Idle (Hung) Printer	*page 1066*
How to Resolve Conflicting Status Messages	*page 1068*

For pointers on troubleshooting print problems, see "Tips on Troubleshooting" on page 1030.

For additional information about printing, refer to the following chapters:

- See Chapter 47, "Overview of Print Management," for overview information about printing and the LP print service.
- See Chapter 48, "Setting Up Printers," for information about setting up printers and print clients using the Admintool application or the LP print service's command-line interface.

- See Chapter 49, “Administering Printers,” for instructions on how to administer printing services after you have set up the printers at your site. This chapter contains instructions for performing ongoing tasks such as setting definitions for your printers and managing print requests.
- See Chapter 50, “Managing Character Sets, Filters, Forms, and Fonts,” for task information on managing character sets, filters, forms, and fonts that may be required to meet the printing needs at your site.
- See Chapter 51, “Customizing the LP Print Service,” for instructions on how to customize the LP print service.

Tips on Troubleshooting

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 1033.

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 in 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 *NIS+ and FNS Administration Guide* in the *Solaris 2.5 System Administrator AnswerBook* 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 936.

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, examine the `lpNet` log (`/var/lp/logs/lpNet`).
- 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 1037 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.x print client using a SunOS 5.x print server (for instructions, see page 1044)
- SunOS 5.x print client using a SunOS 4.1 print server (for instructions, see page 1049)
- SunOS 4.1 print client using a SunOS 5.x print server (for instructions, see page 1054)

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 on a SunOS 5.x print client or a SunOS 5.x print server, 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 Print Service 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 936 and "How to Restart the Print Scheduler" on page 936 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 SunOS 5.x network software was not set up properly.

While you are tracking down the source of the problem, you should stop new requests from being added to the queue.

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 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 (page 1037).
- Check the network (page 1039).
- Check the LP print service basic functions (page 1040).
- Check printing from a SunOS 5.x print client to a SunOS 5.x print server (page 1044).
- Check printing from a SunOS 5.x print client to a SunOS 4.1 print server (page 1049).
- Check printing from a SunOS 4.1 print client to a SunOS 5.x print server (page 1054).

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 1060.

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. Table 52-1 shows the pin configuration for NULL modem cables.

Table 52-1 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 1060.

To check the network:

1. **On a print client, type** `ping` *print-server* **and press Return. On the print server, type** `ping` *print-client* **and press Return.**
   The `ping` command helps you check that the network link between the print server and the print client is set up correctly.

   ```
   # ping neptune
   neptune is alive
   # ping jupiter
   jupiter 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. **Check that the port monitor is configured correctly on the print server.**
   See "Setting Up a Printer With the LP Print Service Commands" on page 923.

3. **Check that the network listen services are registered with the port monitor on the print server.**
   See "Setting Up a Printer With the LP Print Service Commands" on page 923.

To check the basic functions of the LP print service:

1. **On both the print server and print client, make sure that the LP print service is running.**

   a. **Type** `lpstat -r` **and press Return.**
   This command shows whether the LP scheduler is running.

```
# lpstat -r
scheduler is running
```

   b. **If the scheduler is not running, become root or lp, type** `/usr/lib/lp/lpsched`**, and press Return.**
   If you have trouble starting the scheduler, see "How to Unhang the LP Commands" on page 1065.

2. **On both the print server and print client, make sure that the printer is accepting requests.**

   a. **Type** `lpstat -a` **and press Return.**
   This command verifies that the LP system is accepting requests for each printer configured for the system.

```
# lpstat -a
red accepting requests since Wed Mar 13 20:37:07 PST 1995
luna not accepting requests since Wed Apr 17 19:10:55 PDT 1995
unknown reason
```

   b. **If the printer is not accepting requests, become root or lp, type** `accept` ***printer-name*** **and press Return.**
   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. **Type** `lpstat -p` *printer-name* **and press Return.**
   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.

```
# lpstat -p luna
printer luna disabled since Wed Apr 17 19:13:33 PDT 1995.
available.
unknown reason
```

   b. **If the printer is disabled, become root or lp, type** `enable` *printer-name* **and press Return.**

```
# 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. **Type** `lpstat -t` **and press Return.**
   This command tells you the port to which the printer is connected. In the following example, the printer is connected to `/dev/term/a.`

```
# 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.

   b. **Become root or lp.**

c. **Type** `chown lp` *device-filename* **and press Return.**
This command assigns the special user `lp` as the owner of the device file. In this command, *device-filename* is the name of the name of the device file.

d. **Type** `chmod 600` *device-filename* **and press Return.**
This command allows only root 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. **Type** `lpstat -p` *printer-name* `-l` **and press Return.**
The following 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.

```
# lpstat -p luna -l
printer luna is idle. enabled since Wed Feb  4
20:17:21 PST 1970. available.
        Content types: postscript
        Printer types: PS
```

b. **If the printer type or file content type is incorrect, type** `lpadmin -p` *printer-name* `-T printer-type -I` *file-content-type* **and press Return.**
On the print client, try setting the print type to `unknown` and the content type to `any`.

6. **On the print server, make sure that the printer is not waiting because of a printer fault.**

a. **Type** `lpadmin -p` *printer-name* `-F continue` **and press Return.**
This command instructs the LP print service to continue if it is waiting because of a fault.

b. **Type** `enable` *printer-name* **and press Return.**
This command forces an immediate retry.

c. **(Optional) Type** `lpadmin -p` *printer-name* `-A 'write root'` **and press Return.**
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. **Type** `ps -ef` **and press Return.**
In the output from this command, look for the printer port entry. In the following 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.

```
# 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   944   939  0 19:30:47 ?        0:00 lpNet
    root   859   858  0 19:18:54 term/a   0:01 /bin/sh -c /etc/lp/interfaces/luna \
luna-294 rocket!smith “passwd\n##
#
```

b. **Type** `cancel` *request-id* **and press Return.**
In this command, *request-id* is the request ID number for a print request to be canceled.

```
# cancel luna-294
```

The print request is canceled.

c. **Type** `lpadmin -p` *printer-name* `-h` **and press Return.**
This command sets the printer port to be a nonlogin device.

d. **Type** `ps -ef` **and press Return.**
Check the output from this command 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.x client to a SunOS 5.x 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 page 1040. 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 page 1040. 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, type** `ping` *print-server* **and press Return.**
   This command sends an “are you there?” request to the system you specify.

```
# ping neptune
neptune is alive
# ping jupiter
jupiter not available
```

If you receive the message *system* `not available`, you have a network problem.

4. **On the print client, make sure that the print server is identified as type `s5`.**

   a. **Type** `lpsystem -l` **and press Return.**
   The following example shows a print server, `neptune`, that is properly identified as type s5 (SunOS 5.x).

```
# lpsystem -l
System:                       neptune
Type:                         s5
Connection timeout:           never
Retry failed connections:     after 10 minutes
Comment:                      none
#
```

   b. **If the print server is incorrectly identified, type** `lpsystem -t S5` *print-server* **and press Return.**

```
# lpsystem -t S5 neptune
```

5. **On the print client, check the print queue.**

   a. **Type** `cd /var/spool/lp/requests/`*print-client* **and press Return.**
   A record of print requests still in the queue is kept in this directory. For more information about the content of the request logs, see “Log Files” on page 893.

   b. **Type** `ls -l` **and press Return.**
   A list of the print requests in the queue is displayed.

c. **For the print request you want to check, type** `lpstat -o` ***request-id*** **and press Return.**
The following example shows a print request that is queued successfully.

```
# cd /var/spool/lp/requests/neptune
# ls -l
total 12
-rw-rw----   1 lp       lp           43 May 22 19:44 11-0
# lpstat -o luna-11
luna-11             root               364   May 22 19:59
#
```

If the print request is not queued successfully, the client/server connection may be faulty.

6. **Make sure that the client/server connection is not faulty.**

a. **On the print client, type** `tail /var/lp/logs/lpNet` **and press Return.**
This command shows you if `lpNet` can connect to the print server. The following example shows the log for a print request that could not connect to the print server.

```
# tail /var/lp/logs/lpNet
05/21/95 19:36 p  1780 <none> Starting.
05/21/95 19:36 p  1780 <none> Starting lpNetParent.
05/21/95 19:36 p  1780 <none> Initialized & Polling.
05/21/95 19:36 c  1781 neptune Starting.
05/21/95 19:36 p  1780 <none> Started child for neptune, pid =
1781
***05/21/95 19:36 c  1781 neptune Could not connect to remote
child.
05/21/95 19:36 c  1781 neptune Normal process termination.
#
```

b. **If the connection is *not* being made, type** `lpstat -t` **on the print server and press Return.**
This command shows you whether the print server is operating properly. A connection cannot be made otherwise.

The following example shows a print server up and running.

```
# lpstat -t
scheduler is running
system default destination: luna
device for luna: /dev/term/a
luna accepting requests since Thu May 23 20:56:26 PDT 1995
printer red is idle. enabled since Sun May 19 17:12:24 PDT 1995.
available.
printer luna now printing luna-314. enabled since Fri May 24
16:10:39 PDT 1995. available.
luna-129          root            488   May 23 20:43 filtered
#
```

c. **If the print server is not operating properly, go back to Step 1, otherwise go on to Step 6d.**

d. **On the print server, type** `tail /var/lp/logs/lpNet` **and press Return.**
Examine the `lpNet` log to see if the print server is connecting to the client. If there is no entry, `lpNet` is not transmitting correctly. The following example shows the log for a print request that connected to the print server.

```
# tail /var/lp/logs/lpNet
08/10/95 15:17 p   708 <none> Normal process termination.
08/10/95 15:17 p  3802 <none> Starting.
08/10/95 15:17 p  3802 <none> Starting lpNetParent.
08/10/95 15:17 p  3802 <none> Initialized & Polling.
08/10/95 15:17 p  3802 <none> Started child for saturn, pid =
3804
08/10/95 15:17 c  3804 saturn lpd starting (active)
08/10/95 15:17 c  3804 saturn lpd connected to saturn
08/10/95 15:17 c  3804 saturn lpd disconnecting from saturn
08/10/95 15:17 c  3804 saturn lpd connected to saturn
08/10/95 15:17 c  3804 saturn lpd disconnecting from saturn
```

7. **On the print server, make sure that the print client is correctly specified as an s5 system.**

   a. **Type** `lpsystem -l` **and press Return.**
   The following example shows a print client, `neptune`, that is configured correctly.

```
# lpsystem -l
System:                       neptune
Type:                         s5
Connection timeout:           never
Retry failed connections:     after 10 minutes
Comment:                      none
```

   b. **If the print client configuration is incorrect, type** `lpsystem -t s5` ***print-client*** **and press Return.**

```
# lpsystem -t s5 neptune
```

8. **On the print server, make sure that the port monitor and network listen services are set up properly.**

   a. **Type** `sacadm -l` **and press Return.**
   The following example shows a print server configured correctly.

```
# sacadm -l
PMTAG          PMTYPE         FLGS RCNT STATUS     COMMAND
tcp            listen         -    9999 ENABLED    /usr/lib/saf/listen tcp #
#
```

b. Type `pmadm -l` **and press Return.**
The following example shows a print server configured for all three types of services.

```
# pmadm -l
PMTAG          PMTYPE         SVCTAG         FLGS ID       <PMSPECIFIC>
tcp          listen       lp           -    root    - - p - /var/spool/lp/fifos/listenS5 #
tcp           listen        lpd           -    root     \x00020203819514 3a0000000000000000
- p - /var/spool/lp/fifos/listenBSD #
tcp           listen        0             -    root     \x00020ACE8195143a0000000000000000
- c - /usr/lib/saf/nlps_server #
```

If either Step 8a or Step 8b shows a problem, see "Setting Up a Printer With the LP Print Service Commands" on page 923 for information on setting up the port monitor and network listen services.

To check printing from a SunOS 5.x 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 page 1040.

2. **Make sure that the print server is accessible.**

   a. On the print client, type `ping` ***print-server*** **and press Return.**
   An "are you there?" request is sent to the system you specify.

```
# ping neptune
neptune is alive
# ping jupiter
jupiter not available
```

If you receive the message *system* `not available`, you have a network problem.

3. **Make sure that the** `lpd` **daemon on the print server is running.**

   a. **On the print server, type** `ps -ax | grep lpd` **and press Return.**
   If the `lpd` daemon is running, a line is displayed, as shown in the following example. If it is not running, no process information is shown.

```
$ ps -ax | grep lpd
  126 ?  IW    0:00 /usr/lib/lpd
  200 p1 S     0:00 grep lpd
$
```

   b. **If** `lpd` **is not running on the print server, become root on the print server, type** `/usr/lib/lpd &` **and press Return.**

4. **Make sure that the remote** `lpd` **daemon is configured properly.**

   a. **On the print server, become root, and type** `/usr/etc/lpc` **and press Return.**
   The `lpc>` prompt is displayed.

   b. **Type** `status` **and press Return.**
   Status information is displayed. In the following example, the daemon is not running and needs to be restarted.

```
# /usr/etc/lpc
lpc> status
red:
queuing is enabled
printing is enabled
no entries
no daemon present
lpc>
```

   c. **If no daemon is present, at the** `lpc>` **prompt, type** `restart` **and press Return.**
   The daemon is restarted.

   d. **Type** `status` **and press Return.**
   Check the information displayed to verify that the `lpd` daemon has started.

   e. **Type** `quit` **and press Return.**
   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 4.1 print server.**
   On a 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 root, and type** `ps -ef | grep lp` **and press Return.**
   The `lpNet` and `lpsched` daemons should be running, as shown in the following example.

```
# ps -ef | grep lp
   root   162   154 51   Jan 07 ?        0:01 lpNet
   root   154     1 80   Jan 07 ?        0:02 /usr/lib/lpsched
```

   If the `lpNet` daemon is running, skip to Step 7.

   b. **Type** `lpshut` **and press Return.**
   The LP print service is stopped.

   c. **Type** `/usr/lib/lp/lpsched` **and press Return.**
   The LP print service is restarted, including the `lpNet` daemon.

7. **Make sure that the remote print server is identified correctly as a SunOS 4.1 system.**

   a. **On the print client, become root, type** `lpsystem -l` **and press Return.**
   The following example shows a SunOS 4.1 print server, `jupiter`, that is specified correctly, as shown by `Type` being set to `bsd`.

```
# lpsystem -l
System:                     jupiter
Type:                       bsd
Connection timeout:         never
Retry failed connections:   after 10 minutes
Comment:                    none
```

   b. **If the print server is incorrectly identified, type** `lpsystem -t bsd` *print-server* **and press Return.**

8. **Make sure that the print client is** *not* **having trouble connecting to the print server.**

   a. **On the print client, type** `tail -100 /var/lp/logs/lpNet` **and press Return.**
   By examining the `lpNet` log, you can tell if the print client (for example, `jupiter`) is reaching the print server. Normally, the contents will be similar to the following:

```
# tail -100 /var/lp/logs/lpNet
04/18/95 09:40 p  1097 <none> Starting.
04/18/95 09:40 p  1097 <none> Starting lpNetParent.
04/18/95 09:40 p  1097 <none> Initialized & Polling.
04/17/95 19:32 c   965 jupiter lpd connected to jupiter
04/17/95 19:32 c   965 jupiter lpd disconnecting from jupiter
#
```

If the results appear normal, skip the last steps in this procedure. If there is a problem, you will see retries to the BSD system, as shown in the following example.

```
# tail -100 /var/lp/logs/lpNet
05/23/95 14:39 c   120 jupiter lpd retrying connection to
jupiter
05/23/95 14:51 c   120 jupiter lpd retrying connection to
jupiter
05/23/95 15:02 c   120 jupiter lpd retrying connection to
jupiter
#
```

b. On the print client, type `lpsystem -l` *print-server* **and press Return.**
This command shows you the retry and time-out parameters currently set.

c. Type `lpsystem -T {n,0,`*N*`} -R {n,0,`*N*`}` *print-server* **and press Return.**
The `-T` option specifies the length of time a network connection can be idle before it is dropped. Choose either `n` (never time out), `0` (drop immediately), or enter a number (wait *N* minutes, then drop connection). The default is `n`. The `-R` option specifies the length of time to wait before trying to re-establish a connection. Choose either `n` (don’t retry until there is more work), or `0` (try to reconnect immediately), or enter a number (wait *N* minutes before trying to reconnect). The default is wait 10 minutes before trying to reconnect.

```
# lpsystem -T n -R 0 saturn
"saturn" has been modified.
```

To check printing from a SunOS 4.1 client to a SunOS 5.x 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 page 1040. 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 root or lp on the system specified in the following steps.

---

2. **Make sure that the print client is accessible.**

   a. **On the SunOS 5.x print server, become root, then type** `ping` *print-client* **and press Return.**

```
# ping rocket
rocket is alive
```

   If you receive the message *system* `not available`, you have a network problem.

3. **On the print client, become root, then type** `lpr -P` *printer-name filename* **and press Return.**
   This command shows whether the print client is working. The following example shows that the print client is not working correctly.

```
# lpr -P luna /etc/fstab
lpr: cannot access luna
#
```

4. **Make sure that the `lpd` daemon is running on the print client.**

   a. **Type `ps -ax | grep lpd` and press Return.**
   This command shows if the `lpd` daemon is running on the print client. The following example shows that the daemon is running.

```
# ps -ax | grep lpd
  118 ?  IW    0:02 /usr/lib/lpd
#
```

   b. **On the print client, type `/usr/lib/lpd &` and press Return.**

5. **On the print client, make sure that there is a `printcap` entry identifying the print server.**

   a. **Type `lpr -P` *printer-name filename* and press Return.**
   The following example shows that the `/etc/printcap` file does not have an entry for the specified printer.

```
# lpr -P mercury /etc/fstab
lpr: mercury: unknown 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 root on the print client, type** `lpc` **and press Return.**
      The `lpc>` prompt is displayed.

   b. **Type** `restart` *printer-name* **and press Return.**

   c. **Type** `quit` **and press Return.**
      The shell prompt is redisplayed.

```
# lpc
lpc> restart luna
luna:
       no daemon to abort
luna:
      daemon started
#quit
terra$
```

7. **Check the connection to the print server.**

   a. **On the print client, become root, type** `more /var/spool/lpd/`*printer-name*`/log` **and press Return.**
      Frequently, no information is displayed.

   b. **Type** `more /var/spool/lpd/`*printer-name*`/status` **and press Return.**

```
# more /var/spool/lpd/luna/status
waiting for neptune to come up
#
```

c. If the connection is all right, on the print serve, type `lpstat -t` **and press Return.**

This command shows you if the print server is operating properly. The following example shows a print server that is up and running.

```
# lpstat -t
scheduler is running
system default destination: luna
device for luna: /dev/term/a
luna accepting requests since Thu May 23 20:56:26 PDT 1995
printer uranus is idle. enabled since Sun May 19 17:12:24 PDT
1995.
available.
printer luna now printing luna-314. enabled since Fri May
24 16:10:39 PDT 1995. available.
luna-129          root             488   May 23 20:43 filtered
#
```

If the print server is not running, go back to Step 1 before continuing.

d. On the print server, type `tail /var/lp/logs/lpNet` **and press Return.**

Examine the messages to see if the connection is being made from the print client to the `lpNet` process on the print server. The following example shows a print client that is transmitting correctly.

```
# tail /var/lp/logs/lpNet
# tail /var/lp/logs/lpNet
05/24/95 16:26 c  3651 rocket lpd exiting, status=0
05/24/95 16:33 c  3727 rocket lpd starting (passive)
05/24/95 16:33 c  3727 rocket rocket requests recvjob luna
05/24/95 16:33 c  3727 rocket lpd exiting, status=0
05/24/95 16:43 c  3835 rocket lpd starting (passive)
05/24/95 16:43 c  3835 rocket rocket requests recvjob luna
05/24/95 16:43 c  3835 rocket lpd exiting, status=0
#
```

If there is no entry, `lpNet` is not transmitting correctly. If the connection is being made, it implies that the problem is on the print client, and you should check the basic functions of the LP print service on the print client before continuing.

8. **On the print server, make sure that the print client is a BSD system.**

   a. **Type** `lpsystem -l` **and press Return.**
   The following example shows a print client, `rocket`, that is configured correctly.

```
# lpsystem -l
System:                      rocket
Type:                        bsd
Connection timeout:          never
Retry failed connections:    after 10 minutes
Comment:                     none
#
```

   b. **If the print client is not specified correctly,**
   type `lpsystem -t bsd` *print-client* **and press Return.**

9. **On the print server, make sure that the port monitor and network listen services are set up properly.**

   a. **Type** `sacadm -l` **and press Return.**
   The following example shows a print server that is configured correctly.

```
# sacadm -l
PMTAG          PMTYPE         FLGS RCNT STATUS     COMMAND
tcp            listen         -    9999 ENABLED    /usr/lib/saf/listen tcp #
#
```

b. Type `pmadm -l` **and press Return.**
The following example shows a server that is configured for all three services.

```
# pmadm -l
PMTAG          PMTYPE         SVCTAG         FLGS ID       <PMSPECIFIC>
tcp           listen        lp            -    root    - - p - /var/spool/lp/fifos/listenS5 #
tcp            listen        lpd           -    root     \x00020203819514 3a0000000000000000
- p - /var/spool/lp/fifos/listenBSD #
tcp            listen        0             -    root     \x00020ACE8195143a0000000000000000
- c - /usr/lib/saf/nlps_server #
```

If either Step 9a or Step 9b shows a problem, see “Setting Up a Printer With the LP Print Service Commands” on page 923 for instructions on how to set up the port monitor and network listen services.

▼ How to Troubleshoot Incorrect Output

1. **Log in as root 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 873.

   a. **On the print server and print client, type** `lpstat -p` *printer-name* `-l` **and press Return.**
   This command lists the characteristics of the printer.

```
mars$ lpstat -p luna -l
printer luna is idle. enabled since Wed Jan  2 18:20:22 PST 1995. available.
        Content types: simple,postscript
        Printer types: PS
        Description:
        Users allowed:
                (all)
        Forms allowed:
                (none)
        Banner not required
        Character sets:
                (none)
        Default pitch:
        Default page size:
mars$
```

   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 type `lpadmin -p` *printer-name* `-T` *printer-type* and press Return.**
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 1005.

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 type**
`lpadmin -p` *printer-name* `-I` *file-content-type(s)* and press Return.
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. To examine the current settings, type `stty -a` < *device-name* **and press Return.**
This command shows the current `stty` settings for the printer port.

```
mars# 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
mars#
```

Table 52-2 shows the default `stty` options used by the LP print service's standard printer interface program.

Table 52-2 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
`-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

Table 52-2 Default `stty` Settings Used by the Standard Interface Program *(Continued)*

Option	Meaning
`nl0`	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. **To change the `stty` settings, type `lpadmin -p` *printer-name* `-o "stty=`*options*`"` and press Return.**
Use Table 52-3 to choose `stty` options to correct various problems affecting print output.

Table 52-3 `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`
`evenp`
`-parity`
	Sets odd parity
	Sets even parity
	Sets no parity
		Missing or incorrect characters appear randomly

`-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. **Type** `lp -d` *printer-name filename* **and press Return.**
   Verify that the document prints correctly.

▼ How to Unhang the LP Commands

1. **Log in as root or lp.**

2. **Type** `lpshut` **and press Return.**
   If this command hangs, press Control-c and proceed to the next step. If this command succeeds, skip to step 4.

3. **Type** `ps -el | grep lp` **and press Return.**

```
# ps -el | grep lp
   103 ?        0:00 lpNet
   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. **Type** **`kill -15`** *pid1 pid2…* **and press Return.**
   This should stop the LP print service processes. If the processes do not stop, as a last resort go to step 5.

```
# kill -15 103 134
```

5. **Type** `kill -9` *pid1 pid2...* **and press Return.**
   All the `lp` processes are terminated.

6. **Type** `rm /usr/spool/lp/SCHEDLOCK` **and press Return.**
   This command removes the `SCHEDLOCK` file so you can restart the LP print service.

7. **Type** `/usr/lib/lp/lpsched` **and press Return.**
   The LP print service should restart. If you are having trouble restarting the scheduler, see "How to Restart the Print Scheduler" on page 936.

▼ 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. **Type** `lpstat -p` *printer-name* **and press Return.**
   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 Commands" on page 1065.

2. **If the printer is not available (not accepting requests), type** `accept` *printer-name* **and press Return.**
   The printer begins to accept requests into its print queue.

3. **If the printer is disabled, type** `enable` *printer-name* **and press Return.**
   This command re-enables the printer so that it will act on the requests in its queue.

To check for print filtering:

Type `lpstat -o` *printer-name* and press Return.

See if the first waiting request is being filtered. If the output looks like the following example, the file is being filtered; the printer is not hung, it just is taking a while to process the request.

```
terra$ 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
terra$
```

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. **Type** `enable` *printer-name* **and press Return.**
   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, type** `reject` *printer-name* **and press Return.**
   This command stops further queuing of print requests from the print client to the print server.

2. **Type** `ping` *print-server* **and press Return.**
   The information displayed indicates whether the print server and the network between the print client and the print server are up.

3. **Type** `more /var/lp/logs/lpNet` **and press Return.**
   The information displayed may help you pinpoint what is preventing the transmission of print requests from the print client to the print server.

4. **After you fix the problem, type** `accept` *printer-name* **on the print client and press Return.**
   This command allows new print requests to be queued.

5. **If necessary, type** `enable` *printer-name* **on the print client and press Return.**
   This command enables the printer you specify.

To free print requests from a print client that back up in the print server queue:

1. **On the print server, type** `reject` *printer-name* **and press Return.**
   This command stops further queuing of print requests from any print client to the print server.

2. **Type** `more /var/lp/logs/lpsched` **and press Return.**
   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, type** `accept` *printer-name* **on the print server and press Return.**
   This allows new print requests to be queued.

4. **If necessary, type** `enable` *printer-name* **on the print server and press Return.**

▼ How to Resolve Conflicting Status Messages

1. **Type** `lpstat -p` *printer-name* **and press Return.**
   Check that the printer connected to the print server is enabled and is accepting requests. 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, type** `lpstat -p -l` ***printer-name*** **and press Return.**
   This command checks that the definition of the printer on the print client matches the definition of the printer on the print server. 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.

Part 11 —Working With Remote Systems

This part provides instructions for working with remote systems in the Solaris environment.

53 **Working With Remote Systems**
Step-by-step instructions for working with remote systems using `rlogin`, `ftp`, `rcp`, and remote authorization and authentication.

Working With Remote Systems 53

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	*page 1080*
How to Find Out If a Remote System Is Operating	*page 1080*
How to Find Who Is Logged In to a Remote System	*page 1081*
How to Log In to a Remote System (rlogin)	*page 1082*
How to Log Out From a Remote System (exit)	*page 1083*
How to Open an ftp Connection to a Remote System	*page 1086*
How to Close an ftp Connection to a Remote System	*page 1087*
How to Copy Files From a Remote System (ftp)	*page 1087*
How to Copy Files to a Remote System (ftp)	*page 1090*
How to Copy Files Between a Local and a Remote System (rcp)	*page 1096*

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 Figure 53-1:

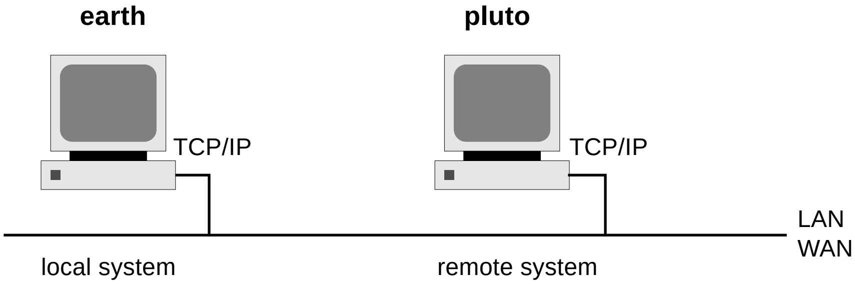


Figure 53-1 A Remote System

On Solaris 2.x systems, TCP/IP configuration is established automatically during start-up. For more information, see the *TCP/IP and Data Communications Administration Guide.*

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. Figure 53-2 shows a simplified illustration to describe authentication for remote logins.

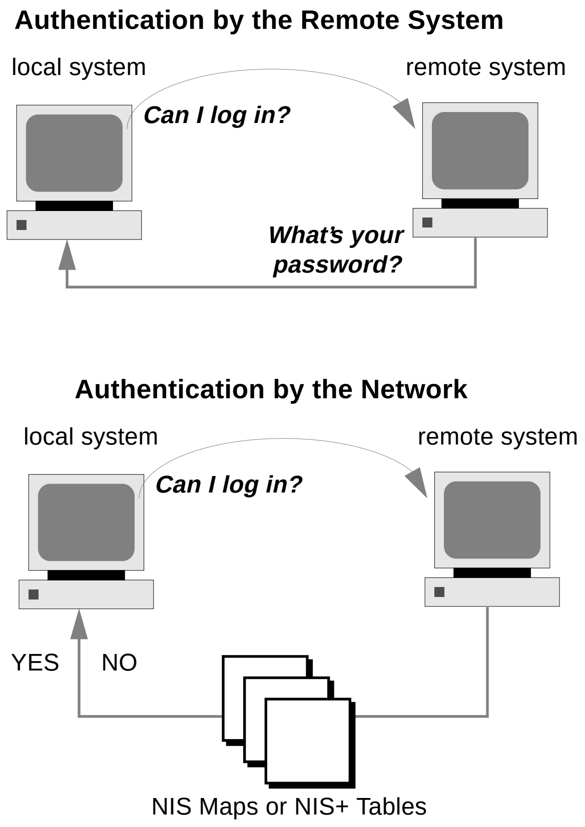


Figure 53-2 Authentication for Remote Logins (`rlogin`)

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.

Caution – 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.

Caution – 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.

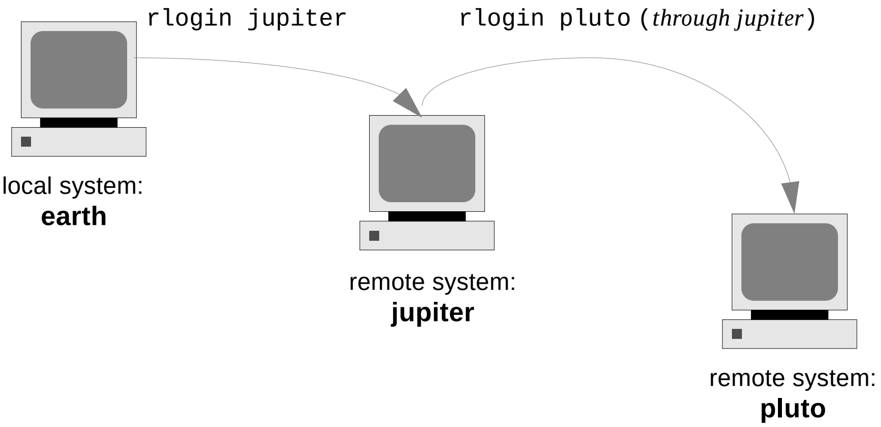
Users can create `.rhosts` files in their home directories. Using the `.rhosts` file is another way to allow trusted access between their own accounts on different systems without using the `/etc/hosts.equiv` file.

Caution – 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 may create a `.rhosts` file granting access to whomever the user chooses without the system administrator’s knowledge.

The only secure way to manage `.rhosts` files is to completely disallow them. See “How to Search for and Remove .rhosts Files” on page 1080 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 may 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`:

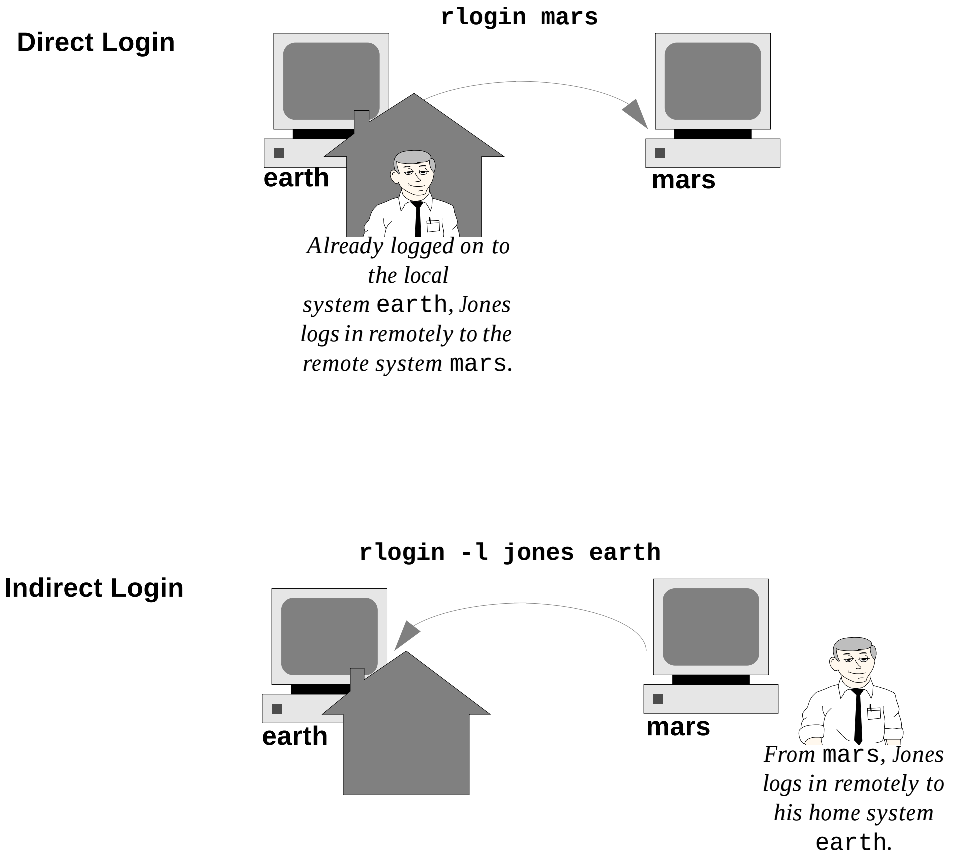


Of course, the user could have logged out of `jupiter` and then logged in directly to `pluto`, but this type of linking can be more convenient.

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 Figure 53-3 on page 1077.



Already logged on to the local system `earth`*, Jones logs in remotely to the remote system* `mars`*.*

From `mars`*, Jones logs in remotely to his home system* `earth`*.*

Figure 53-3 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 Table 53-1.

Table 53-1 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 /
pluto(/)
```

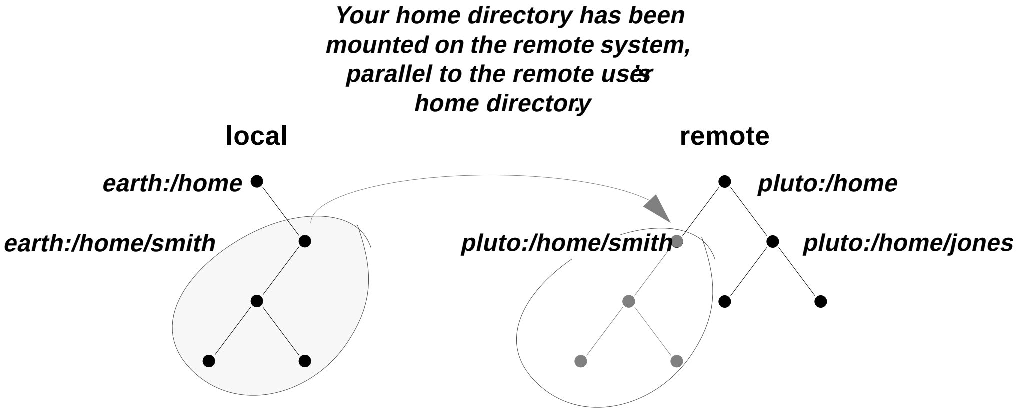
However, if the `rlogin` command finds your home directory, it sources both your `.cshrc` and `.login` files. Therefore, after a remote login, your prompt is your standard login prompt, and the current directory is the same as when you log in locally. For example, if your usual prompt displays your system name and working directory, and if upon login your working directory is your home directory . . .

```
earth(/home/smith):
```

. . . 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 root.**
2. **Search for and remove `.rhosts` files by using the `find(1)` command.**

```
# find home-directories -name .rhosts -print | xargs -i -t rm{}
```

In this command,

home-directories Is the path to a directory where user's home directories are located.

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(1M)` command.

```
$ ping system-name | ip-address
```

In this command,

system-name Is the name of the remote system.

ip-address Is 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 may 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 the `ping` man page.

▼ 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
```

In this command,

`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 named `pluto`. The first user logged in from the system console on November 18 and has been logged on for 4 hours and 10 minutes. The second user logged in from a remote system, `mars`, on the same date, and has been logged on for a similar amount of time.

```
$ rusers -l pluto
smith     pluto:console      Nov 18 09:19     4:10
jones     mars:console       Nov 18 09:20     4:11   (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
```

In this command,

`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 pluto
Last login: Thu Oct 27 15:38:59 from earth
Sun Microsystems Inc.,   SunOS 5.4   Generic September 1994
You have mail.
```

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: Thu Oct 27 15:38:59 from earth
Sun Microsystems Inc.,   SunOS 5.4   Generic September 1994
You have mail.
```

▼ 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: smith logged out at Mon Oct 31 10:10:54 PST 1994
Goodbye!
Connection closed.
```

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 Table 53-2 on page 1085.

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 53-2 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 the task titled "How to Open an ftp Connection to a Remote System," on page 1086.
`open`	Logs in to the remote system from the command interpreter
`close`	Logs out of the remote system and returns to the command interpreter
`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 the `ftp(1)` man page.

▼ 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 1084.

2. **Open a connection to a remote system by using the `ftp(1)` 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 53-2 on page 1085.

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 (UNIX(r) System V Release 4) 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
```

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 1086.

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 file-name
```

6. **To copy multiple files at once, use the `mget` command.**

```
ftp> mget file-name [file-name ...]
```

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 Smith has an open `ftp` connection with the system belonging to the user Jones, and uses the `get` command to copy a single file from Jones' home directory to his own home directory:

Before opening a connection to Jones' system, Smith `cd`'s to the the `tmp` directory where the files from the remote system will be copied.

```
$ cd /home/smith/tmp
ftp> pwd
257 "/home/smith" is current directory
```

Smith changes the remote working directory to `/home/jones`

```
ftp> cd ../jones
250 CWD command successful
ftp> pwd
257 "/home/jones" is current directory
```

Smith uses the `ls -l` command to make sure he has Read access to the file he wants to copy.

```
ftp> ls -l
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.135.331 ...
total 122
-rw-rw-rw-   1 jones   staff   30720  Dec 19 15:14  JonesFile.txt
-rw-rw-rw-   1 jones   staff   27450  Dec 19 15:20  GoodFile.txt
```

Smith uses the `get` command to copy `JonesFile.txt` from the remote working directory to the local working directory.

```
ftp> get JonesFile.txt
200 PORT command successful.
150 ASCII data connection for JonesFile.txt (129.135.331....
226 ASCII Transfer complete.
local: JonesFile.txt remote: JonesFile.txt
25700 bytes received in 0.32 seconds (77 Kbytes/s)
ftp> bye
```

In this example, the same user Smith uses the `mget` command to copy a set of files from Jones' home directory to his own home directory. Note that Smith can accept or reject individual files in the set.

Smith uses a wildcard to copy all the files in Jones' home directory that end with `.txt`.

The `mget` command asks for confirmation for each file. Smith declines the first file and accepts the second.

```
ftp> mget *.txt



mget JonesFile.txt? n
mget GoodFile.txt? y
200 PORT command successful.
150 ASCII data connection for JonesFile.txt (129.135.331....
226 ASCII Transfer complete.
local: JonesFile.txt remote: JonesFile.txt
25700 bytes received in 0.32 seconds (77 Kbytes/s)
ftp> bye
```

▼ 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 1086.

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 file-name
```

6. **To copy multiple files at once, use the `mput` command.**

```
ftp> mput file-name [file-name ...]
```

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 Smith opens an `ftp` connection from the `/home/smith/transfer` directory into the `/home/jones/transfer` directory, and uses the `put` command to copy a file from his system to Jones' system:

Before opening a connection to Jones' system, Smith `cd`'s to the proper source directory on the local system

```
$ cd /home/smith/transfer
/home/smith/transfer

$ ftp pluto
   .
   .
   .
230 User smith logged in.
ftp> pwd
257 "/home/smith" is current directory
```

Smith changes the remote working directory to `/home/jones` and uses the `ls -l` command to make sure he has Write access to target directory

```
ftp> cd ../jones
250 CWD command successful
ftp> ls -l
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.135.331 ...
total 122
drwxrwxrwx   1 jones   staff     512  Dec 19 10:45  transfer
-rw-rw-rw-   1 jones   staff   30720  Dec 19 15:14  JonesFile.txt
-rw-rw-rw-   1 jones   staff   27450  Dec 19 15:20  GoodFile.txt
```

After verifying that he has Write permission to the target directory, Smith `cd`'s to it.

```
ftp> cd transfer
```

Smith uses the `put` command to copy `ToJones.txt` from the local working directory (source) to the remote working directory (target).

```
ftp> put ToJones.txt
200 PORT command successful.
150 ASCII data connection for ToJones.txt (129.135.331....
226 Transfer complete.
local: ToJones.txt remote: ToJones.txt
25700 bytes sent in 0.32 seconds (77 Kbytes/s)
ftp> bye
```

In this example, the same user Smith uses the mget command to copy a set of files from Jones' home directory to his own home directory. Note that Smith can accept or reject individual files in the set.

Smith uses a wildcard to copy all the files in Jones' home directory that end with .txt.

The mget command asks for confirmation for each file. Smith declines the first file and accepts the second.

```
ftp> mget *.txt



mget JonesFile.txt? n
mget GoodFile.txt? y
200 PORT command successful.
150 ASCII data connection for JonesFile.txt (129.135.331....
226 ASCII Transfer complete.
local: JonesFile.txt remote: JonesFile.txt
25700 bytes received in 0.32 seconds (77 Kbytes/s)
ftp>
```

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 may 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 1078.

Table 53-3 provides a representative 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 53-3 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 1082. 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
```

In this command,

`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 53-3 on page 1095, and as illustrated in the examples below.

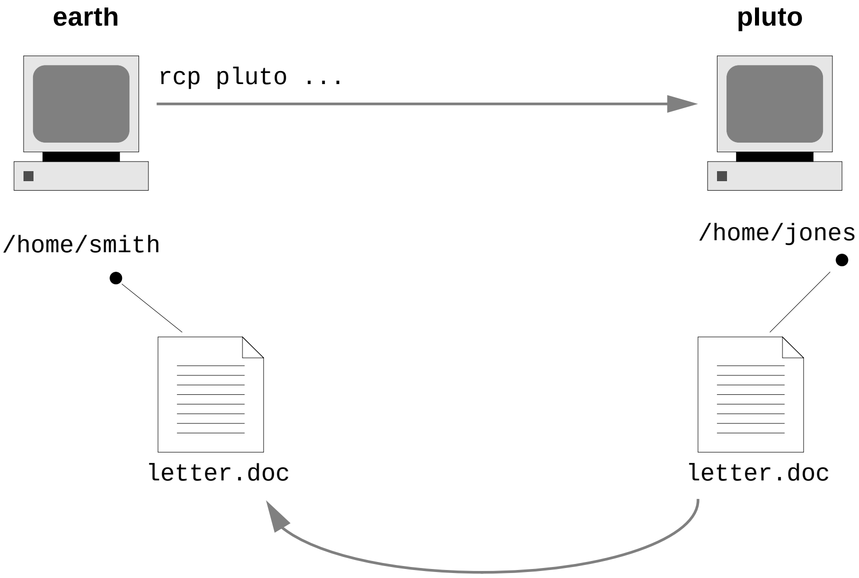
You can use the "~" and "`.`" characters to specify the path portions of the local file or directory names. Note, however, that "~" applies to the current user, not the remote system, and that "`.`" applies to system you are logged into. For explanations of these symbols, see Table 53-3 on page 1095.

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 .
```



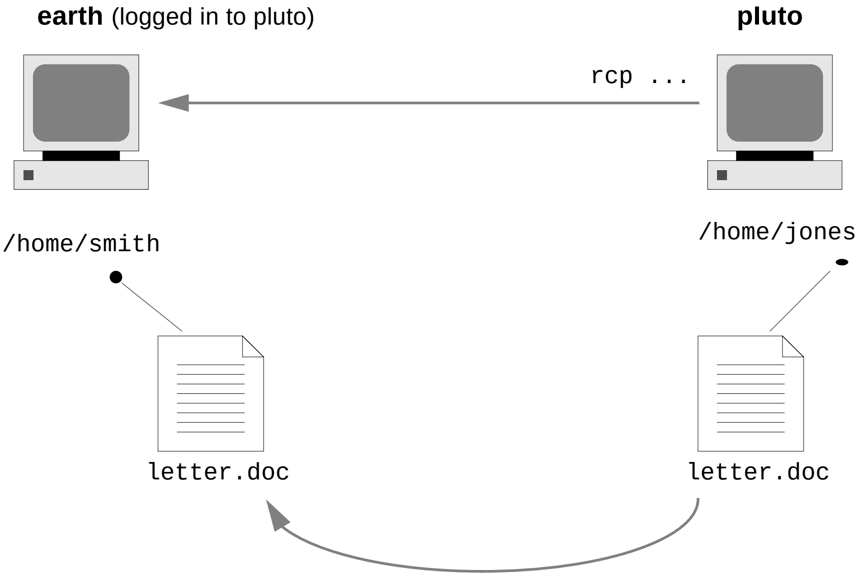
Since the `rcp` operation is performed without a remote login, the "`.`" symbol applies to the local system, not the remote system.

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:

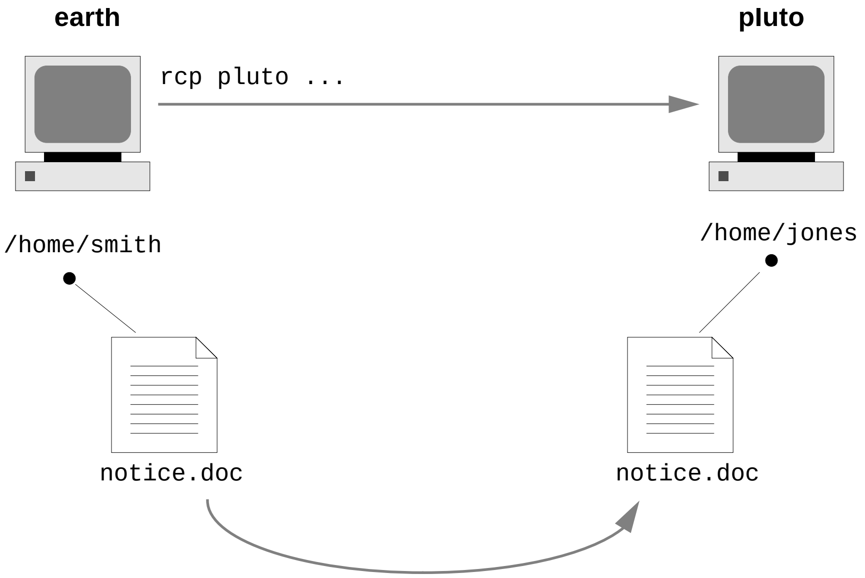
```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/jones): rcp letter.doc ~
```



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
```



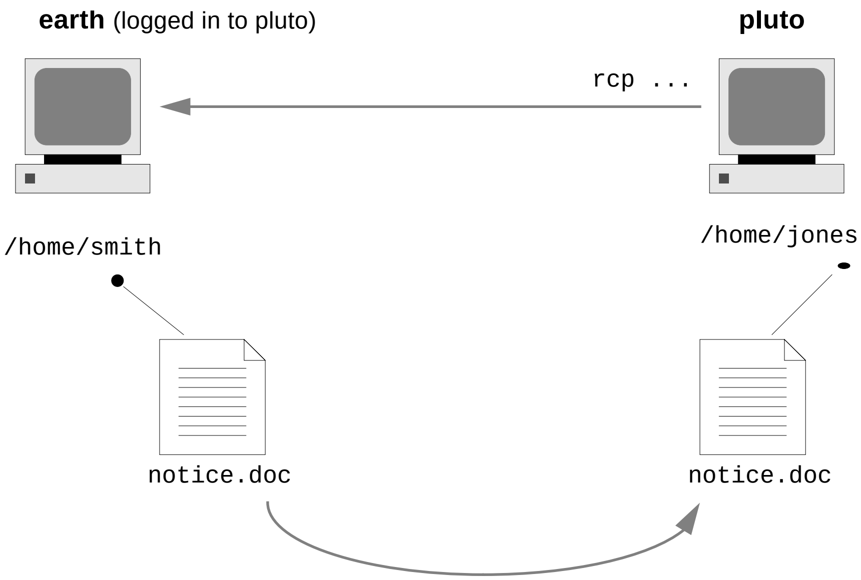
Because no remote filename is provided, the file `notice.doc` is copied into the `/home/jones` directory with the same name.

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
```

Part 12 — Managing Terminals and Modems

This part provides instructions for managing terminals and modems.

54 **Overview of Managing Terminals and Modems**
Provides overview information about terminals and modems.

55 **Setting Up Terminals and Modems With Serial Port Manager**
Provides step-by-step instructions for setting up terminals and modems.

56 **Setting Up Terminals and Modems With the Service Access Facility**
Provides step-by-step instructions for using SAF commands to set up terminals and modems.

Overview of Managing Terminals and Modems 54

This chapter provides the overview information for managing terminals and modems by using the Solstice Serial Port Manager, a Solstice AdminSuite application; or the Service Access Facility (SAF). This is a list of the overview information in this chapter.

Terminals, Modems, Ports, and Services	*page 1103*
Tools for Managing Terminals and Modems	*page 1106*
Serial Port Manager	*page 1107*
Service Access Facility	*page 1108*

For step-by-step instructions about how to set up terminals and modems with Serial Port Manager, see Chapter 55, “Setting Up Terminals and Modems With Serial Port Manager.”

For step-by-step instructions about how to set up terminals and modems with the SAF, see Chapter 56, “Setting Up Terminals and Modems With the Service Access Facility.”

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 2.x environment.

Terminals

Note – 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 converse. 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, incorporates 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 may 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 may 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.

Table 54-1 describes the two types of port monitors included in the Solaris 2.x environment.

Table 54-1 Port Monitor Types

Port Monitor	Description
listen(1M)	Controls access to network services, handling remote print and file system requests. A common use of the listen port monitor is to listen for requests from the LP print service. For more information on the listen port monitor, see Chapter 56, “Setting Up Terminals and Modems With the Service Access Facility.” The listen port monitor is *not* used when you set up modems and alphanumeric terminals.
ttymon(1M)	Provides access to the login services needed by modems and alphanumeric terminals. Solstice Serial Port Manager automatically sets up a ttymon port monitor to process login requests from these devices. Using Solstice Serial Port Manager to set up terminals and modems is described in Chapter 55, “Setting Up Terminals and Modems With Serial Port Manager.”

You may 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

In previous Solaris releases, you may have used Administration Tool or Solaris commands to set up terminals and modems on a local system or remote systems. In the Solaris 2.5 release, there are three ways to manage terminals and modems:

- **Admintool –** A new tool to set up terminals and modems on a local system only.
- **Solstice AdminSuite –** Includes the tool, Serial Port Manager, to set up terminals and modems on local and remote systems.
- **Service Access Facility (SAF) –** A collection of background processes and administrative commands, pmadm and sacadm, used from the command line to set up port services and monitors.

Table 54-2 highlights some situations when you may choose to use one tool or the other.

Table 54-2 When to Use Solstice Serial Port Manager or Service Access Facility

Procedure	**Suggested Tool**	**Comment**
Set up terminals and modems	Serial Port Manager	Solstice Serial Port Manager quickly sets up typical port services for terminals and modems. Solstice Serial Port Manager provides most of the functionality of the `pmadm` command.
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 Solstice Serial Port Manager.
Not hanging up a modem 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.

Serial Port Manager

The Serial Port Manager 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

54

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.

Setting Up Terminals and Modems With Serial Port Manager

55

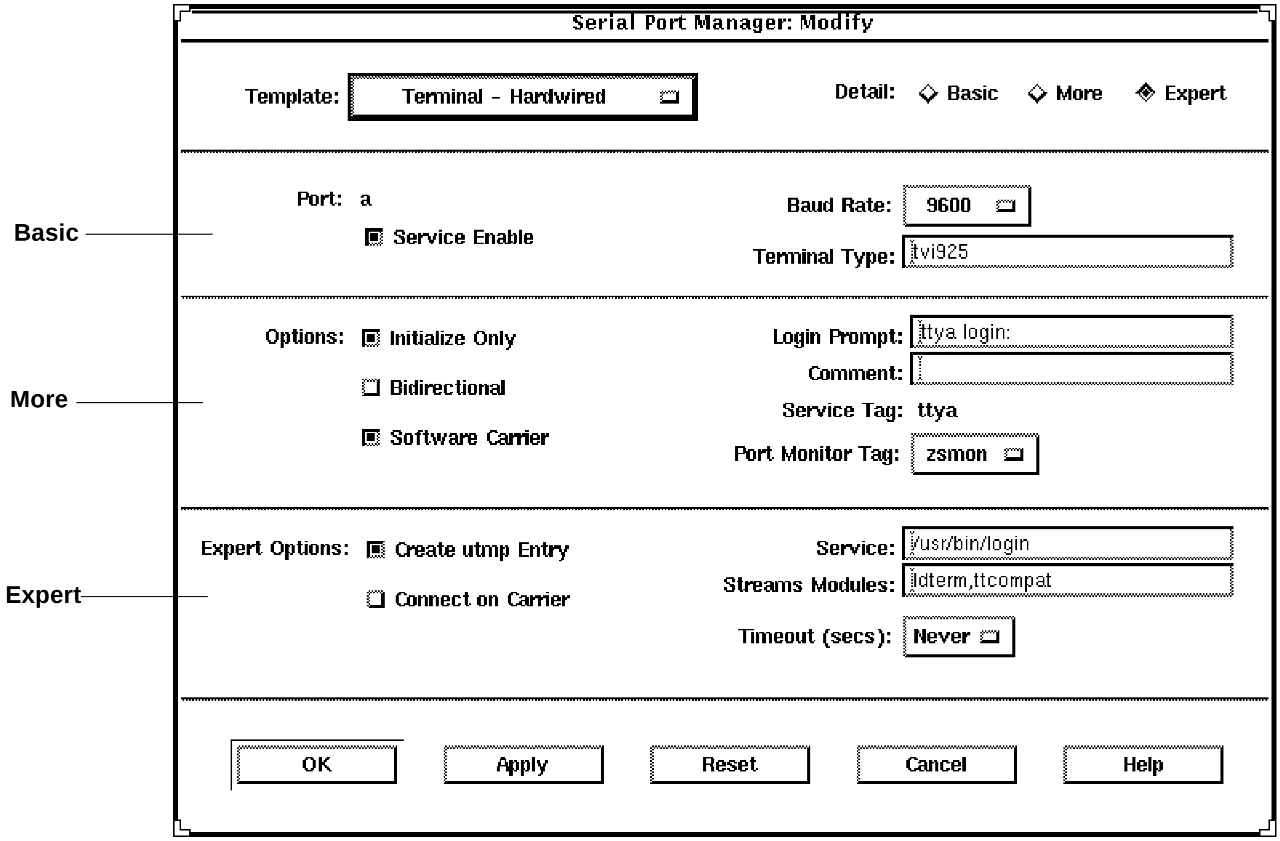
This chapter provides step-by-step instructions for setting up terminals and modems using Serial Port Manager, a Solstice AdminSuite application. This is a list of the step-by-step instructions in this chapter.

How to Start Serial Port Manager	*page 1115*
How to Set Up a Terminal	*page 1117*
How to Set Up a Modem	*page 1119*
How to Set Up a Modem for Use With UUCP	*page 1121*
How to Initialize a Port	*page 1123*
How to Disable a Port	*page 1125*
How to Remove a Port Service	*page 1126*
How To Troubleshoot Terminals and Modems	*page 1127*

For overview information about terminals and modems, see Chapter 54, “Overview of Managing Terminals and Modems.”

Setting Up Terminals and Modems

When setting up port information, choose Modify from the Edit menu to bring up the Modify Service window. This window provides access to the port templates and provides information on the port in three levels of detail—Basic, More, and Expert.



Template: Terminal - Hardwired

Detail: ◇ Basic ◇ More ◆ Expert

Port: a

☒ Service Enable

Baud Rate: 9600

Terminal Type: tvi925

Options: ☒ Initialize Only

□ Bidirectional

☒ Software Carrier

Login Prompt: ttya login:

Comment:

Service Tag: ttya

Port Monitor Tag: zsmon

Expert Options: ☒ Create utmp Entry

□ Connect on Carrier

Service: /usr/bin/login

Streams Modules: ldterm,ttcompat

Timeout (secs): Never

Note – The Modify Service 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 Service window are listed in Table 55-1.

Table 55-1 Modify Service Window Items

Detail	Item	Description
Basic	Port	Lists the port or ports you selected from Serial Port Manager's main window.
	Service	Specifies that the service for the port is turned on (enabled).
	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 `utmp` Entry	Specifies that a `utmp` 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

Table 55-2 describes the menu items (and their default values) when setting up a terminal using Serial Port Manager.

Table 55-2 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 `utmp` Entry	yes
	Connect on Carrier	no
	Service	`/usr/bin/login`
	Streams Modules	`ldterm,ttcompat`
	Timeout (secs)	Never

Setting Up Modems

Table 55-3 describes the three modem templates available when setting up a modem using Serial Port Manager.

Table 55-3 Modem Templates

Modem Configuration	**Description**
Dial-In Only	Users may dial in to the modem but cannot dial out.
Dial-Out Only	Users may dial out from the modem but cannot dial in.
Bidirectional	Users may either dial in or out from the modem.

Table 55-4 describes the default values of each template.

Table 55-4 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
	Service Tag	—	—	—
	Port Monitor Tag	`zsmon`	`zsmon`	`zsmon`
Expert	Create `utmp` 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

Table 55-5 describes the default values for the Initialize Only template.

Table 55-5 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
	Service Tag	—
	Port Monitor Tag	`zsmon`
Expert	Create `utmp` Entry	yes
	Connect on Carrier	no
	Service	`/usr/bin/login`
	Streams Modules	`ldterm,ttcompat`
	Timeout (secs)	Never

▼ How to Start Serial Port Manager

1. **Verify that the following prerequisites are met. To use Serial Port Manager, you must have:**
   - Solstice AdminSuite software installed.
   - A bit-mapped display monitor. The Solstice AdminSuite 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.

     If you want to perform administration tasks on a system with an ASCII terminal as the console, use Solaris commands instead.
   - OpenWindows™ software. Start this software with the following command:

```
$ /usr/openwin/bin/openwin
```

   - Membership in the `sysadmin` group (group 14) and the required access privileges for managing the NIS or NIS+ database.

Note – If your name service is NIS+, you must be a member of the NIS+ admin group.

The system being configured may be your local system or a remote system. Table 55-6 describes the required access privileges for setting up terminals and modem.

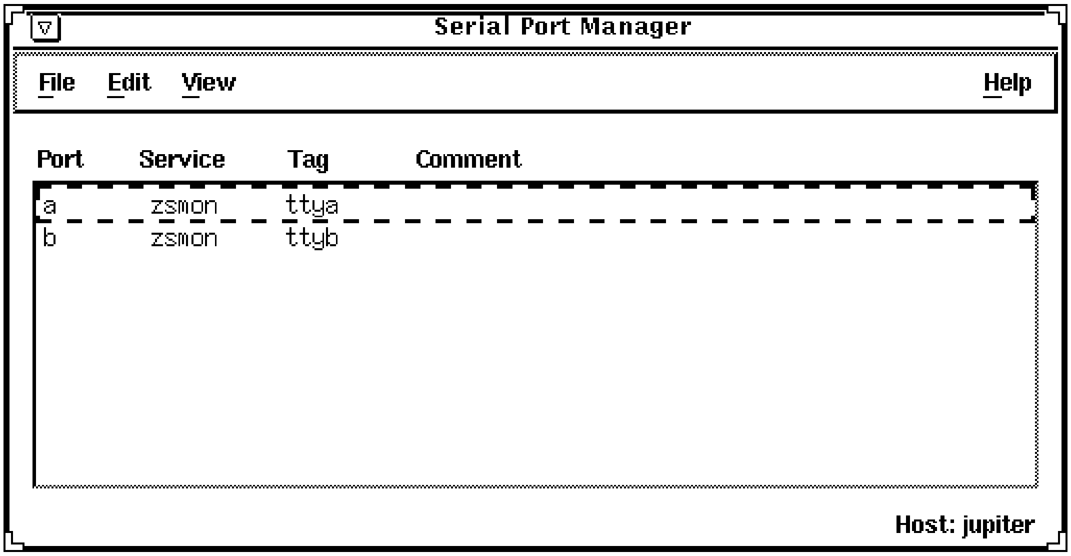
Table 55-6 Required Access Privileges for Setting Up Terminals and Modems

For a ...	You Must ...
Local system	Run Solstice AdminSuite as root, or Be a member of the UNIX sysadmin group (GID 14) for that system
Remote system	Be a member of the UNIX sysadmin group (GID 14) for that system

2. **Type** `solstice &` **from a Command or Shell Tool prompt and press Return.**
   The Solstice Launcher is displayed.

3. **Click on the Serial Port Manager icon.**
   The Serial Port Manager main window is displayed.

Example—Serial Port Manager Main Window



Port	Service	Tag	Comment
a	zsmon	ttya	
b	zsmon	ttyb	

Host: jupiter

▼ How to Set Up a Terminal

1. **Start Serial Port Manager from the Solstice Launcher, if not done already.**
   See “How to Start Serial Port Manager” on page 1115 for more information.

2. **Select the port or ports that will be used with a terminal.**

3. **Choose Modify from the Edit menu.**
   The Modify Service window appears in the Basic Detail mode. To enter additional details, select either the More or Expert Detail modes.

4. **Choose Terminal-Hardwired from the Use Template menu.**
   See Table 55-2 on page 1112 for a description of the Terminal–Hardware menu items.

5. **Change values of template entries if desired.**

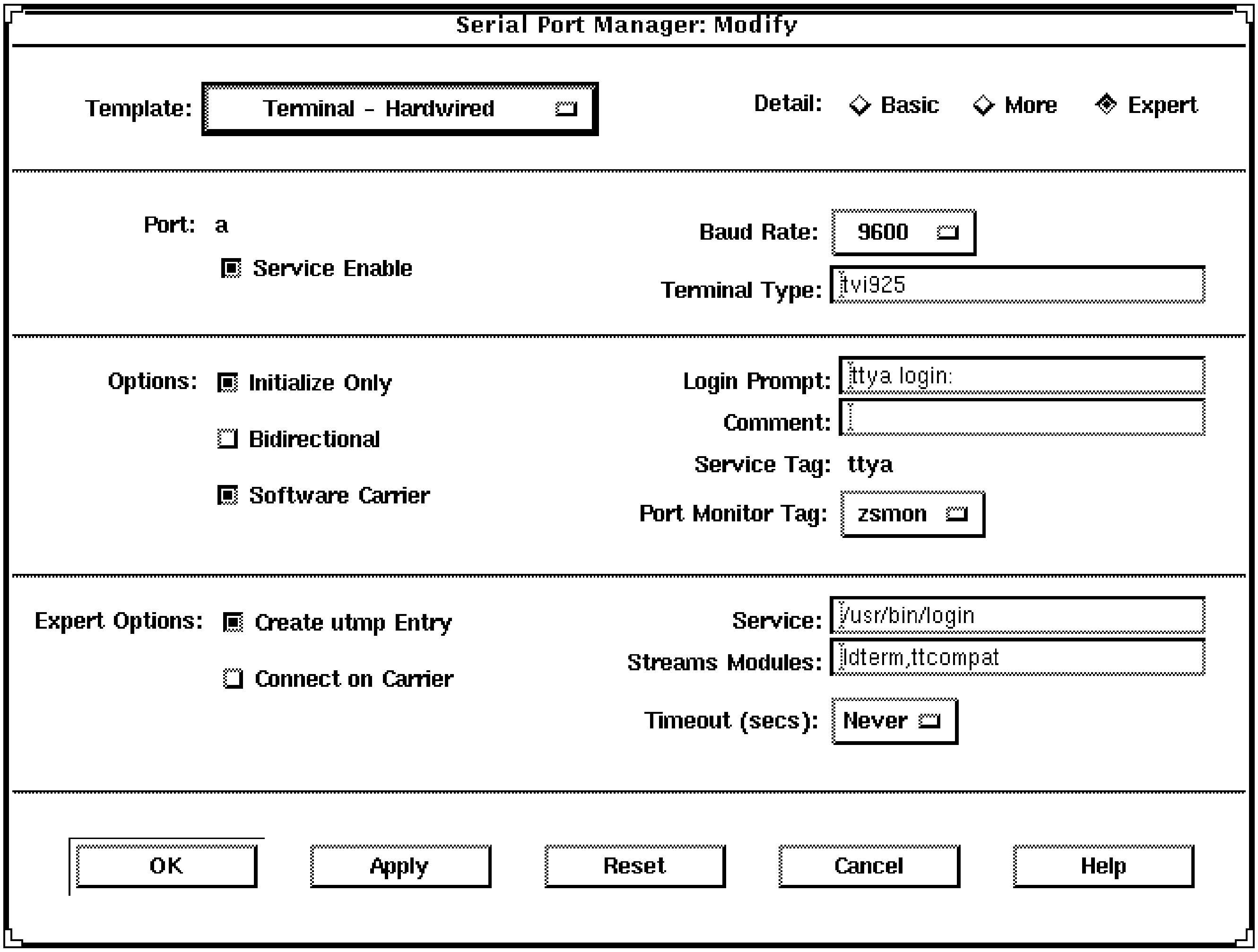
6. **Click on OK to configure the port.**

Verification—Setting Up a Terminal

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



Template: Terminal - Hardwired

Detail: ◇ Basic ◇ More ◆ Expert

Port: a

☒ Service Enable

Baud Rate: 9600

Terminal Type: tvi925

Options: ☒ Initialize Only

☐ Bidirectional

☒ Software Carrier

Login Prompt: ttya login:

Comment:

Service Tag: ttya

Port Monitor Tag: zsmon

Expert Options: ☒ Create utmp Entry

☐ Connect on Carrier

Service: /usr/bin/login

Streams Modules: ldterm,ttcompat

Timeout (secs): Never

▼ How to Set Up a Modem

1. **Start Serial Port Manager from the Solstice Launcher, if not done already.**
   See “How to Start Serial Port Manager” on page 1115 for more information.

2. **Select the port or ports that will be used with a modem.**

3. **Choose Modify from the Edit menu.**
   The Modify Service window appears in the Basic Detail mode. To enter additional details, select either the More or Expert Detail modes.

4. **Choose the modem configuration template from the Use Template menu that meets or most closely matches your modem service.**
   See Table 55-3 on page 1113 for a description of each template.

   See Table 55-4 on page 1113 for the default values of each template. If a UUCP service will be used to dial in to your modem on a Solaris 2.x system, see “How to Set Up a Modem for Use With UUCP” on page 1121” for the rest of the procedure.

5. **Change values of template entries if desired.**

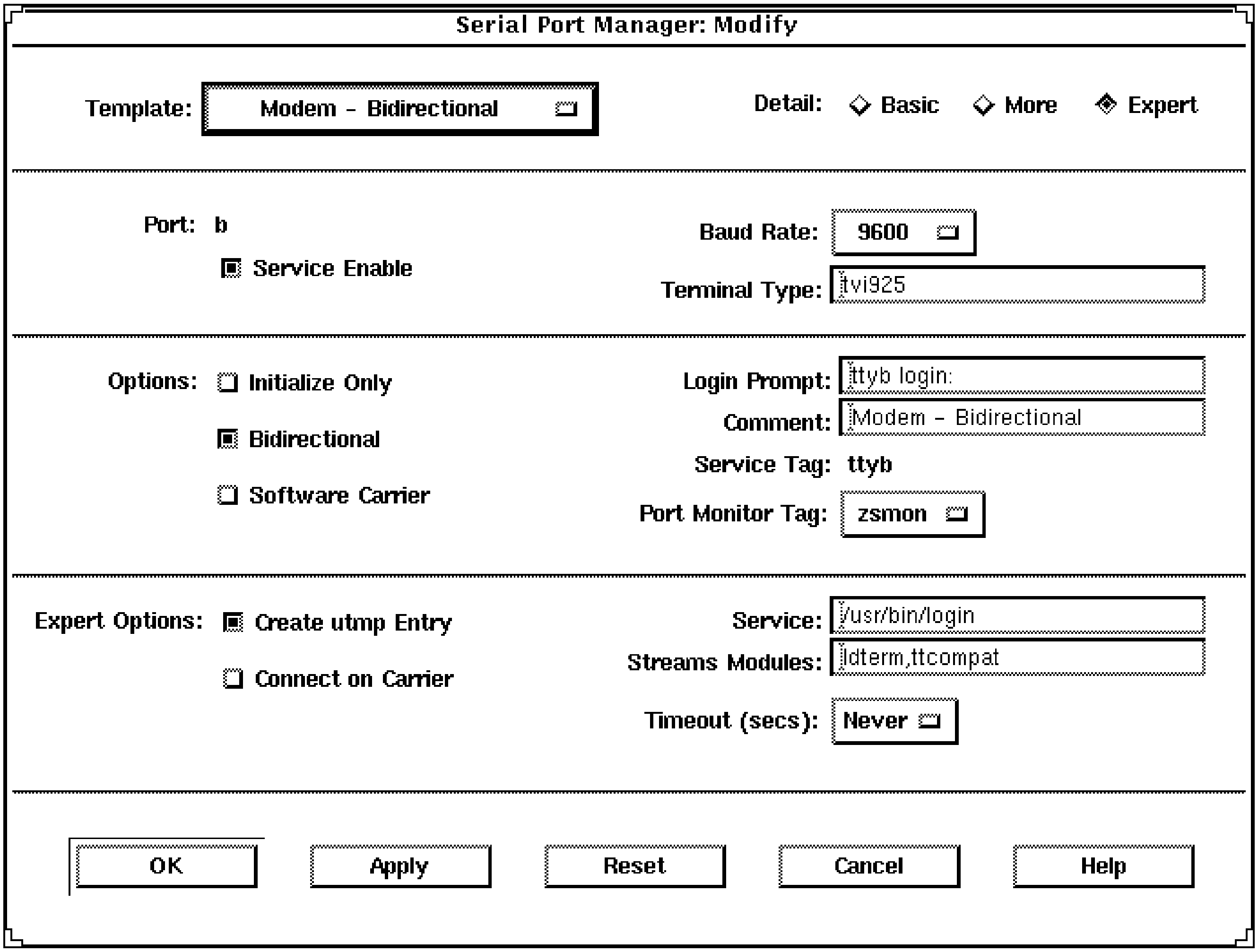
6. **Click on OK to configure the port.**

Verification—Setting Up a Modem

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



Serial Port Manager: Modify

Template: Modem - Bidirectional

Detail: ◇ Basic ◇ More ◆ Expert

Port: b

☒ Service Enable

Baud Rate: 9600

Terminal Type: tvi925

Options: □ Initialize Only

☒ Bidirectional

□ Software Carrier

Login Prompt: ttyb login:

Comment: Modem - Bidirectional

Service Tag: ttyb

Port Monitor Tag: zsmon

Expert Options: ☒ Create utmp Entry

□ Connect on Carrier

Service: /usr/bin/login

Streams Modules: ldterm,ttcompat

Timeout (secs): Never

▼ How to Set Up a Modem for Use With UUCP

UUCP sends information to a service using seven bits and even parity. Solaris 2.x 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 Serial Port Manager from the Solstice Launcher, if not done already.**
   See "How to Start Serial Port Manager" on page 1115 for more information.

2. **Select the port or ports that will be used with a modem.**

3. **Choose Modify from the Edit menu.**
   The Modify Service window appears in the Basic Detail mode. For additional details, select either the More or Expert Detail modes.

4. **Select Other from the Baud Rate menu.**
   A window listing baud rates from the `/etc/ttydefs` file is displayed.

5. **Enter a baud rate that provides seven bit, even parity service. Click on OK.**

6. **Change values of other template entries if desired.**

7. **Click on OK to configure the port.**

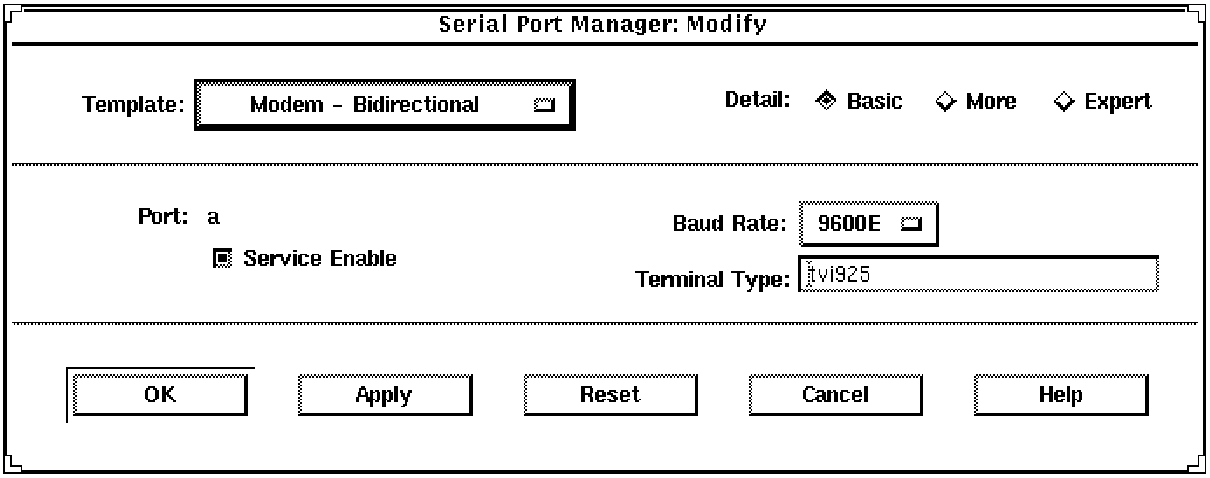
Verification—Setting Up a Modem for Use With UUCP

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.



Serial Port Manager: Modify

Template: Modem - Bidirectional

Detail: ● Basic ○ More ○ Expert

Port: a

☒ Service Enable

Baud Rate: 9600E

Terminal Type: tvi925

▼ How to Initialize a Port

1. **Start Serial Port Manager from the Solstice Launcher, if not done already.**
   See “How to Start Serial Port Manager” on page 1115 for more information.

2. **Select the port or ports that you want to initialize.**

3. **Choose Modify from the Edit menu.**
   The Modify Service window appears in the Basic Detail mode. To enter additional details, select either the More or Expert Detail modes.

4. **Choose Initialize Only - No Connection from the Use Template menu.**
   See Table 55-5 on page 1114 for a description of the Initialize Only - No Connection template.

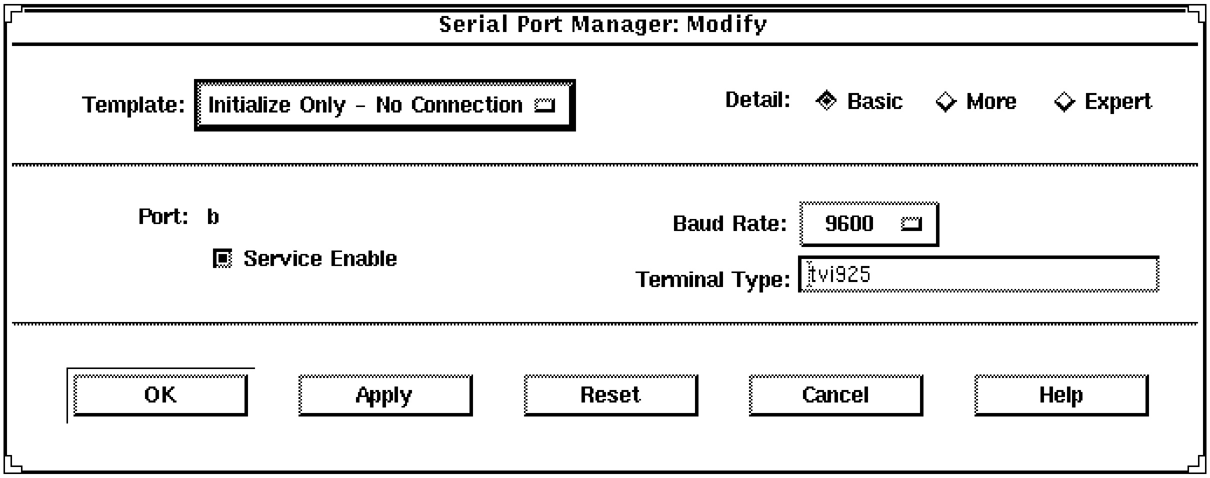
5. **Click on OK to initialize the port.**

Verification—Initializing a Port

Use the pmadm command to verify the port has been disabled.

```
$ pmadm -l -s ttyb
```

Example—Completed Modify Window to Initialize a Port



Template: Initialize Only - No Connection

Detail: ◆ Basic ◇ More ◇ Expert

Port: b

☒ Service Enable

Baud Rate: 9600

Terminal Type: tvi925

▼ How to Disable a Port

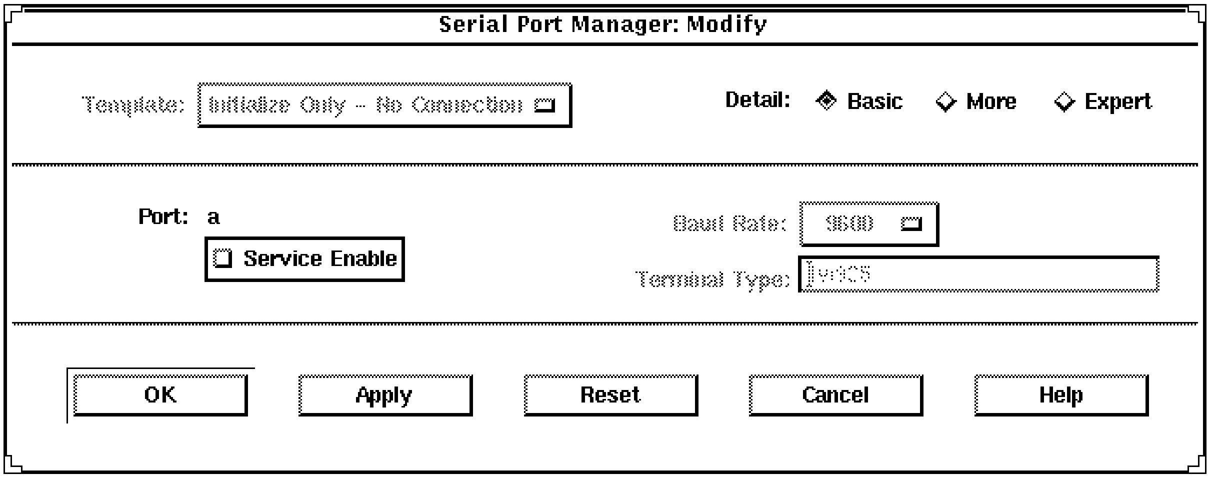
1. **Start Serial Port Manager from the Solstice Launcher, if not done already.**
   See "How to Start Serial Port Manager" on page 1115 for more information.
2. **Select the port or ports that you want to disable.**
3. **Choose Modify from the Edit menu.**
4. **Click on the Service Enable button to disable the port service in the Modify window.**
   This button acts as a toggle switch to enable or disable a port service.
5. **Click on OK to disable the port.**

Verification—Disabling a Port

Use the pmadm command to verify the port service has been disabled.

```
$ pmadm -l -s ttya
```

Example—Completed Modify Window to Disable a Port



Serial Port Manager: Modify

Template: Initialize Only - No Connection

Detail: ◆ Basic ◇ More ◇ Expert

Port: a

□ Service Enable

Baud Rate: 9600

Terminal Type: tvi925

▼ How to Remove a Port Service

1. **Start Serial Port Manager from the Solstice Launcher, if not done already.**
   See “How to Start Serial Port Manager” on page 1115 for more information.

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 may cancel the delete operation or continue with it.

Verification—Removing a Port Service

Use the `pmadm` command to verify the port service has been deleted.

```
$ pmadm -l -s ttya
```

▼ How To Troubleshoot Terminals and Modems

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.

1. Begin by checking 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.

2. 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*.

3. 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 `sacadm` 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.

4. 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

Setting Up Terminals and Modems With the Service Access Facility 56

This chapter explains in detail what a system or network administrator needs to know about the Service Access Facility (SAF) in the Solaris 2.x environment.

If you want to see examples of specific SAF commands, skip the first section, "Using the Service Access Facility," and use the following list to find the instructions you need.

Using the Service Access Facility	*page 1130*
Overall Administration: sacadm Command	*page 1131*
Port Monitor Service Administrator: pmadm Command	*page 1132*
Port Monitors: TTY Monitor and Network Listener	*page 1135*
Administering ttymon Port Monitors	*page 1137*
Administering ttymon Services	*page 1141*
Administering listen Port Monitors	*page 1144*
Administering listen Port Monitor Services	*page 1147*
Troubleshooting the Network Listener: listen Port Monitor	*page 1150*
Reference Material for Service Access Facility Administration	*page 1150*

For overview information about terminals and modems, see Chapter 54, "Overview of Managing Terminals and Modems."

56

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 may differ from network to network, SAC and the administrative programs `sacadm` and `pmadm` are network independent.

Table 56-1 illustrates 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 56-1 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` `listen`	Monitors serial port login requests 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

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 may 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 Figure 56-1 on page 1133, 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). Figure 56-1 shows only a single ttymon port monitor.

After ttymon has been started, it monitors the serial port lines for service requests.

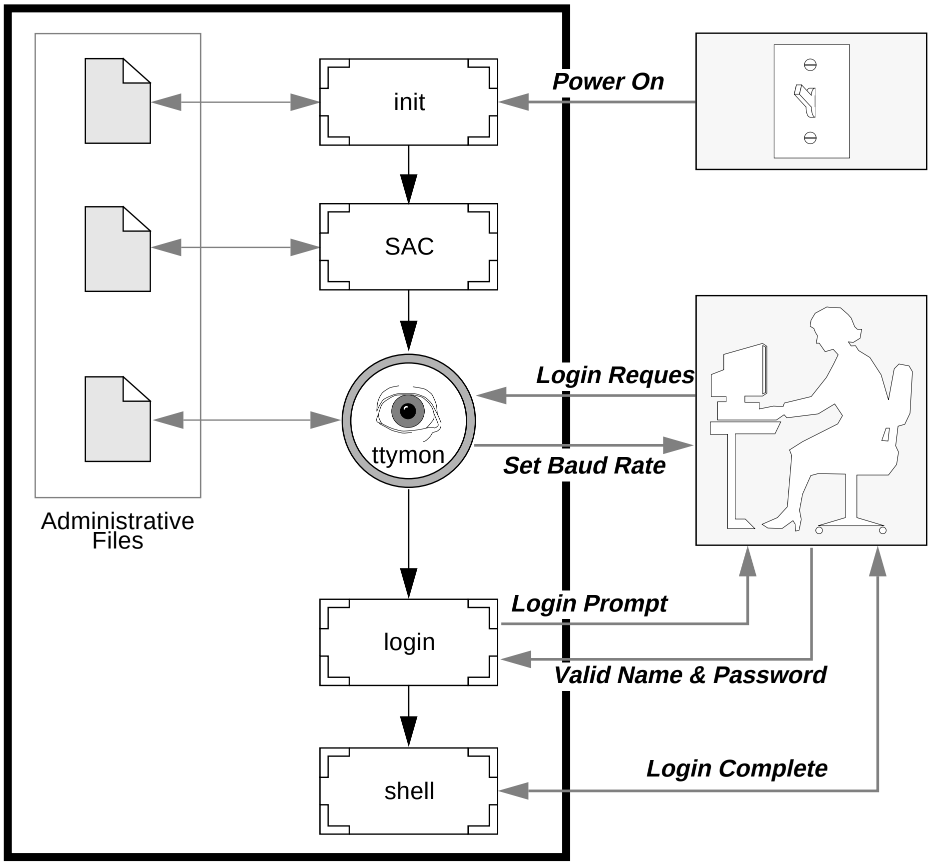


Figure 56-1 How `ttymon` Helps Process a Login Request

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/utmp entry if required, establishes the service environment, and invokes the service associated with the port.

After the service terminates, ttymon cleans up the /etc/utmp 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 2.x 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 2.x users the same services that `getty` did under previous versions of Solaris 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.

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 the `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 may be used with any connection-oriented transport provider that conforms to the Transport Layer Interface (TLI) specification. In the Solaris 2.x environment, `listen` port monitors provide additional network services not provided by `inetd`, such as print service.

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`. See `nlsadmin(1M)` for more details.

Each network has 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 a detailed description.

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"
```

In this command,

`-a`	is 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
```

In this command,

`-l` is the *list* port monitor status flag

`-p` specifies the *pmtag* `mbmon` as the port monitor tag

How to Stop a ttymon Port Monitor

To kill a `ttymon` port monitor, type:

```
# sacadm -k -p mbmon
```

In this command,

`-k` is 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
```

In this command,

`-s` is 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
```

In this command,

`-d` is 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
```

In this command,

`-e` is 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
```

In this command,

`-r` is 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`.

In this command,

`-a` is 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. In that embedded command:

`-b` is 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
```

In this command,

`-l` is the flag for a list of 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
```

In this command,

`-l` is the flag for a list of service information

Listing All Services of a Port Monitor

To list all services of a port monitor, type:

```
# pmadm -l -p mbmon
```

In this command,

`-l` is the flag for a list of service information

`-p` specifies the *pmtag* `mbmon` as the port monitor tag

▼ How to Enable a Port Monitor Service

To enable a disabled port monitor service, type:

```
# pmadm -e -p mbmon -s a
```

In this command,

`-e` is 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
```

In this command,

`-d` is the *disable* flag

`-p` specifies the *pmtag* `mbmon` as the port monitor tag

`-s` specifies the *svctag* `a` as the port monitor *service* tag

Administering `listen` *Port Monitors*

Use the `sacadm` command to add, list, kill, start, enable, disable, or remove a `listen` port monitor.

Note – You must be superuser to perform the following procedures.

▼ How to Add a `listen` Port Monitor

To add a `listen` port monitor, type:

```
# sacadm -a -p tcp -t listen -c /usr/lib/saf/listen -v ‘nlsadmin -V‘ -y "le0 ethernet"
```

In this command,

`-a` is the *add* port monitor flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

`-t` specifies the port monitor type as `listen`

`-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 `listen` Port Monitor Status

To list the status of a `listen` port monitor, type:

```
# sacadm -l -p tcp
```

In this command,

`-l` is the *list* port monitor status flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

How to Stop a `listen` Port Monitor

To kill a `listen` port monitor, type:

```
# sacadm -k -p tcp
```

In this command,

`-k` is the *kill* port monitor flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

▼ How to Start a `listen` Port Monitor

To start a `listen` port monitor, type:

```
# sacadm -s -p tcp
```

In this command,

`-s` is the *start* port monitor flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

▼ How to Enable a `listen` Port Monitor

To enable a `listen` port monitor, type:

```
# sacadm -e -p tcp
```

In this command,

`-e` is the *enable* port monitor flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

▼ How to Disable a `listen` Port Monitor

To disable a `listen` port monitor, type:

```
# sacadm -d -p tcp
```

In this command,

`-e` is the *disable* port monitor flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

How to Remove a `listen` Port Monitor

To remove a `listen` port monitor, type:

```
# sacadm -r -p tcp
```

In this command,

`-r` is the *remove* port monitor flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

Administering `listen` *Port Monitor Services*

Use the `pmadm` command to add, enable, disable, and list the services associated with a `listen` port monitor.

Note – You must be superuser to perform the following procedures.

How to Add a `listen` Port Monitor Service

To add a `listen` port monitor service, type:

```
# pmadm -a -p tcp -s lp -i root -v ‘nlsadmin -V‘ -m "‘nlsadmin -o
/var/spool/lp/fifos/listenS5‘"
```

In this example, a listen service is added for print requests from remote SunOS 5.x machines. This service does not listen for requests from SunOS 4.x machines.

Note – In this example, the input wraps to the next line. Do not put a Return or line feed after the `-o`.

In this command,

-a is the *add* port monitor service flag

-p specifies the *pmtag* tcp as the port monitor tag

-s specifies the *svctag* lp 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 listen-specific configuration data formatted by nlsadmin

The above pmadm command contains an embedded nlsadmin command. In that embedded command -o specifies the full path name of a first-in first-out (FIFO) or named STREAM through which a standing server will receive the connection.

▼ How to List listen Port Monitor Services

To list the services associated with a listen port monitor, type:

```
# pmadm -l -p tcp
```

In this command,

-l is the *list* port monitor service flag

-p specifies the *pmtag* tcp as the port monitor tag

How to Enable a `listen` Port Monitor Service

To enable a `listen` port monitor service, type:

```
# pmadm -e -p tcp -s lp
```

In this command,

`-e` is the *enable* flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

`-s` specifies the *svctag* `lp` as the port monitor *service* tag

How to Disable a `listen` Port Monitor Service

To disable a `listen` port monitor service, type:

```
# pmadm -d -p tcp -s lp
```

In this command,

`-d` is the *disable* flag

`-p` specifies the *pmtag* `tcp` as the port monitor tag

`-s` specifies the *svctag* `lp` as the port monitor *service* tag

Troubleshooting the Network Listener: `listen` Port Monitor

Use the following tips to remedy `listen` port monitor difficulties.

1. Begin with the network.

   The network listener is suspect when users report that they cannot print to a network printer. Begin by issuing the `/usr/sbin/ping` command from the print server to the client and back to determine if the network is up.

2. Check the `listen` port monitor.

   Use `sacadm` to check that the listener is starting. Use `pmadm` to check that the print service is configured correctly and that the service is enabled.

3. Check the print server's configuration.

   See Chapter 52, "Troubleshooting Printing Problems," for more information.

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

File Name	Description
/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

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.

State	Notes
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 may 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.
Network (`listen`) Port States	
Enabled	The `listen` port monitor scans the network for service requests and invokes services in response to those requests.
Disabled	If the `listen` port monitor is killed or disabled, the ports it controls are automatically disabled. If you specify this state, `listen` will send out the “disabled” message when it receives a connection request.

Part 13 —Managing System Security

This part provides instructions for managing system security in the Solaris environment.

57 **Overview of Managing System Security**
Provides overview information about file, system, and network security. It also includes information about the Automated Security Enhancement Tool (ASET).

58 **Securing Files**
Provides step-by-step instructions to display file information, change file ownership and permissions, set special permissions, and use access control lists (ACLs).

59 **Securing Systems**
Provides step-by-step instructions to check login statuses, set up dial-up passwords, restrict root access, and monitor root access and `su` attempts.

60 **Running ASET**
Provides 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.

61 **ASET Troubleshooting**
Provides recovery information about some of the ASET error messages.

Overview of Managing System Security 57

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.

Granting Access to a Computer System	*page 1156*
File Security	*page 1160*
System Security	*page 1165*
Network Security	*page 1173*
Automated Security Enhancement Tool (ASET)	*page 1177*

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.

Granting Access to a Computer System

The first line of defense is to control access to your system. You can control 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
- Monitoring setuid programs
- Tracking root login
- Installing a firewall

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 2.x system software restricts control of certain system devices to the user login account. Only a process running as root 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 may want to allow some people to read some files, and give other people permission to change or delete some files. You may have some data that you do not want anyone else to see. Chapter 58, "Securing Files," 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 may accidently 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, rather than just mistyped your password.

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 root 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.

`setuid` *Programs*

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.

Tracking Root Login

Your system requires a root password to boot into 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 root privileges on your system.

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 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 may 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 may be more appropriate to your particular needs. You can call CERT/CC at its 24-hour hotline: (412) 268-7090, or contact the team via email to `cert@cert.sei.cmu.edu`.

57

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. This section describes how to secure the files and directories in a file system.

User Classes

For each file, there are three classes of users:

- 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

Table 57-1 lists and describes the file permissions.

Table 57-1 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

Table 57-2 lists and describes the directory permissions.

Table 57-2 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 to 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 root has access to all files and directories on the system.

File Types

A file can be one of six types. Table 57-3 lists the possible file types.

Table 57-3 File Types

Symbol	Type
–	Text or program
d	Directory
b	Block special file
c	Character special file
p	Named pipe (FIFO)
l	Symbolic link

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 permissions 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 root permission by executing a program that sets the user ID to root. See “How to Set Special Permissions in Absolute Mode” on page 1209 for detailed instructions on setting special permissions.

You should monitor your system to stay aware of any unauthorized use of the `setuid` and `setgid` permissions to gain root privileges. See “How to Find Files With setuid Permissions Set” on page 1210 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 `bin` or `sys`. Only root can set these permissions.

`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 created the process. 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   1 root     sys       10332 May  3 08:23 /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.

`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 bin      mail      62504 May  3 07:58 /usr/bin/mail
```

When setgid permission is applied to a directory, files created in this directory belong to the group the directory belongs to, not the group the creating process belongs to. Any user who has write permission in the directory can create a file there—however, the file will not belong to the group of the user, but will belong to the group of the directory.

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 uucppublic:

```
drwxrwxrwt   2 uucp    uucp    512 May 24 09:48 /var/spool/uucppublic
```

Be sure to set the sticky bit manually when you set up a public directory on a tmpfs file system.

File Administration Commands

Table 57-4 lists the file administration commands that you can use on files or directories.

Table 57-4 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.

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 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.

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.

Table 57-5 shows some typical umask settings, and the effect on an executable file.

Table 57-5 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

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 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.

For example, assume you had a file that you wanted everyone in a group to be able to read. With that situation, you would 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 let you set that up. However, you could set up an ACL for that file to give only one person in the group write permissions on the file.

Table 57-6 lists the ACL commands that you can use on files or directories.

Table 57-6 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 root 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 root only. A would-be root 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 *NIS+ and FNS 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.

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 1226 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)`.

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 password to these login accounts.

Table 57-7 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, and these passwords should be distributed on a need-to-know basis.

Table 57-7 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.
`daemon`	`1`	Controls background processing.
`bin`	`2`	Owns most of the 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` to require a password. See `eeprom(1M)` for more information.

Passwords

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, and known only to users. You should ask your users to choose their passwords carefully, and 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” in the *System Administration Guide, Volume I* 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 User Account Manager or the `nispasswd(1)` 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 User Account Manager or the `yppasswd(1)` 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 root can read the `/etc/shadow` file. You can use User Account Manager or the `passwd(1)` command to change a user's password on a local system.

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 root 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 /etc/dialups file is a list of terminal devices, for example:

```
/dev/term/a
/dev/term/b
```

The /etc/d_passwd file has two fields. The first is the login shell that will require a password, and the second is the encrypted password, for example:

```
/usr/lib/uucp/uucico:encrypted_password:
/usr/bin/csh:encrypted_password:
/usr/bin/ksh:encrypted_password:
/usr/bin/sh:encrypted_password:
```

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.

The basic dial-up password sequence is shown in Figure 57-1.

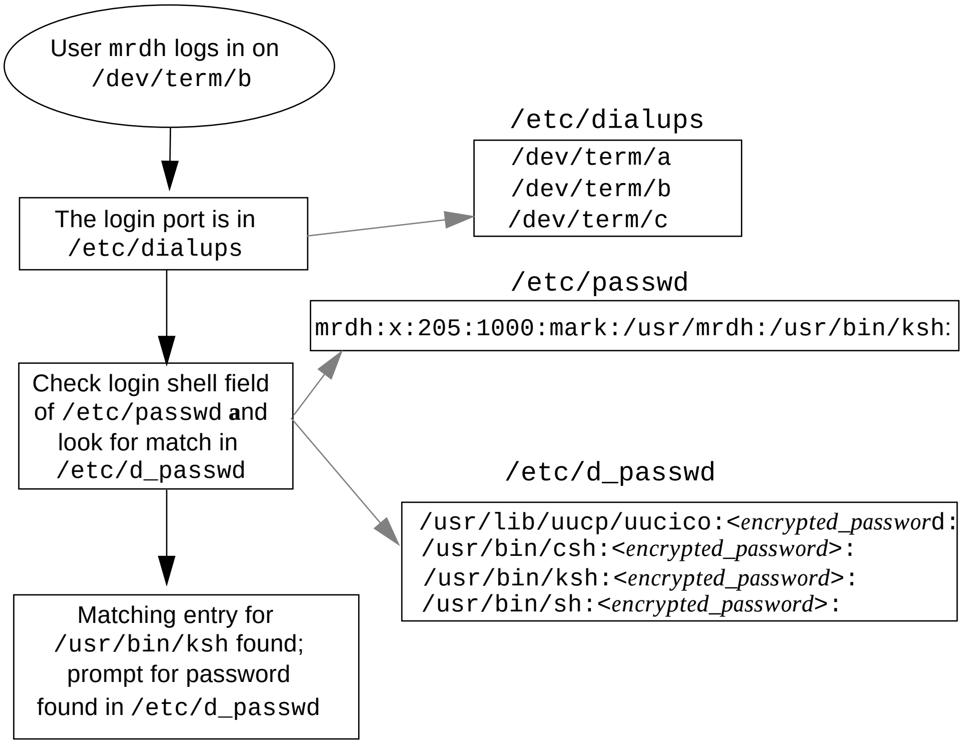

User mrdh logs in on /dev/term/b
The login port is in /etc/dialups
/etc/dialups
/dev/term/a
/dev/term/b
/dev/term/c
/etc/passwd
mrdh:x:205:1000:mark:/usr/mrdh:/usr/bin/ksh:
Check login shell field of /etc/passwd and look for match in /etc/d_passwd
/etc/d_passwd
/usr/lib/uucp/uucico:<encrypted_password:
/usr/bin/csh:<encrypted_password>:
/usr/bin/ksh:<encrypted_password>:
/usr/bin/sh:<encrypted_password>:
Matching entry for /usr/bin/ksh found; prompt for password found in /etc/d_passwd

Figure 57-1 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:*:

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 directory /usr/lib. (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 up 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 sh(1) for information about the restricted shell.

Root Access Restrictions

The root 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 root.

You can protect the root account on a system by restricting root access to a specific device through the `/etc/default/login` file. For example, if root access is restricted to the console, you can log in to a system as root 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` command to become root. See "How to Restrict Root Login to the Console" on page 1229 for detailed instructions.

Note – Restricting root login to the console is set up by default when you install a system.

Monitoring Who Is Using the `su` Command

You have to use the `su` command to change to another user, for example, if you want to become root. For security reasons, you may want to monitor who has been using the `su` command, especially those user's who are trying to gain root access.

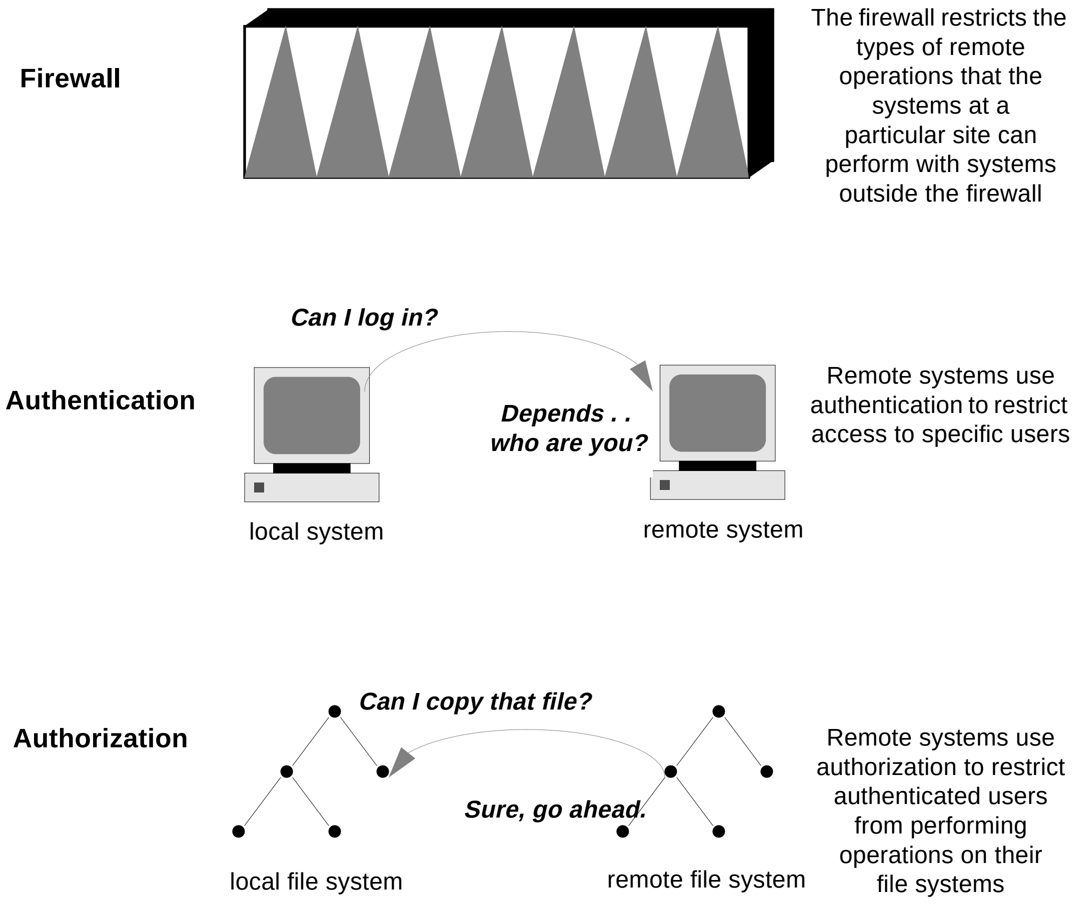
You can start monitoring `su` attempts through the `/etc/default/su` file. Through this file, you can enable the `/var/adm/sulog` file to monitor each time the `su` command is used to change to another user. See "How to Monitor Who Is Using the su Command" on page 1229 for detailed instructions.

The `sulog` file lists all uses of the `su` command, not only those used to switch user to `root`. 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 root access from a remote system. This is a good way to immediately detect someone trying to gain root access on the system you are currently working on. See "How to Display Root Access Attempts to the Console" on page 1230 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. Figure 57-2 describes the security restrictions you can impose on remote operations.



Firewall

The firewall restricts the types of remote operations that the systems at a particular site can perform with systems outside the firewall

Authentication

Remote systems use authentication to restrict access to specific users

Authorization

Remote systems use authorization to restrict authenticated users from performing operations on their file systems

Figure 57-2 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 must maintain strict and rigidly enforced security on the firewall, but since the other hosts on the network are protected, security on those systems 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.

The Automated Security Enhancement Tool (ASET) can be used to make a system into a firewall, and to enforce high security on a firewall system, which is described on page 1177.

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. Table 57-8 lists the types of authentications and authorizations that can help protect your systems on the network against unauthorized use.

Table 57-8 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.	*NIS+ and FNS 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.	*System Administration Guide, Volume II*
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. Secure RPC can also be used to provide additional security to the NFS™ environment, called Secure NFS.	*NFS Administration Guide*

Table 57-8 Types of Authentication and Authorization

Type	Description	Where to Find Information
Solstice AdminSuite™	The Solstice AdminSuite product provides authentication and authorization mechanisms to remotely manage systems with the AdminSuite tools.	*Solstice AdminSuite 2.1 User's 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 *NFS Administration Guide* for more information about sharing files.

Restricting Root Access

In general, root is not allowed root access to file systems shared across the network. Unless the server specifically grants root privileges, a user who is logged in as root 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 root privileges on a shared file system on a per-host basis, using the `root=`*hostname* option to the `share` command.

Alternative to Secure RPC

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 root 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 may 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”
```

Automated Security Enhancement Tool (ASET)

SunOS 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 598.)

ASET uses master files for configuration. Master files, reports, and other ASET files are in the directory `/usr/aset`. 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 typing the following command in a command shell:

```
$ aset
```

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 below:

- *Low security* – This level 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* – This level 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* – This level 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 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 1196 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
- An account `nobody`
- 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/utmp
- /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 1174. 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 `-n` option of the `aset` command directs the log to be delivered by electronic mail to a designated user. For a complete list of ASET options, refer to the `aset(1M)` man page.

Example of an 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
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 1179. 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 directory `/usr/aset/reports`. 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.

Figure 57-3 shows an example of the `reports` directory structure.

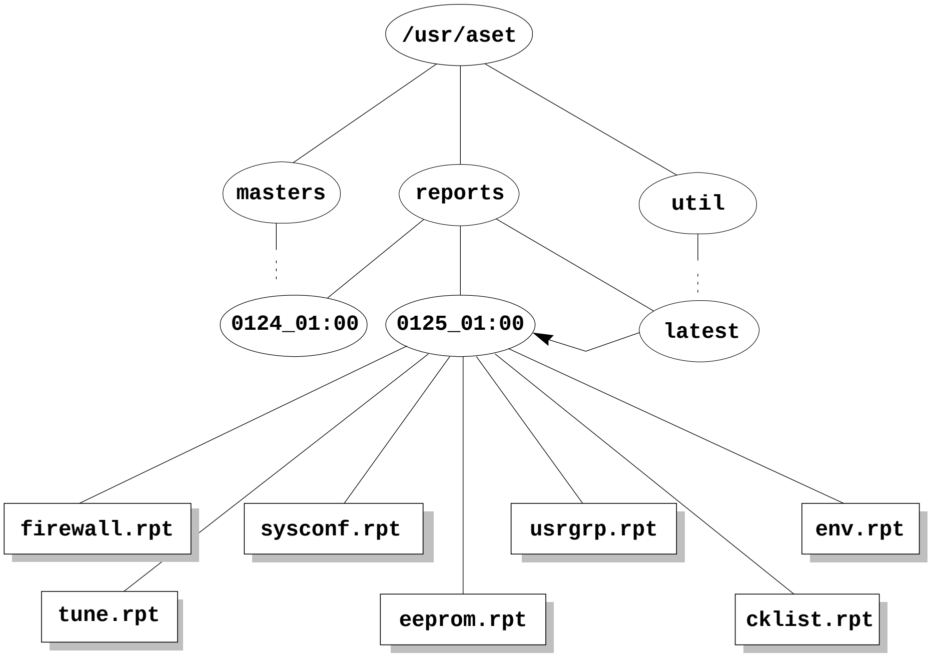


Figure 57-3 `reports` Directory Structure

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 directory `latest` 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 Report Files

Each report file is named after the task that generates it. See Table 57-9 for a list of tasks and their reports.

Table 57-9 ASET Tasks and Resulting Reports

Tasks	**Report**
System files 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 accidently 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 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 Report Files

After you monitor the report files for a period during which there are no configuration changes or system updates, you may find that the content of the reports begins 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 environment variables: `CKLISTPATH_LOW`, `CKLISTPATH_MED`, and `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 may 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 may 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 1234.

The format of PERIODIC_SCHEDULE follows the format of crontab entries. See the 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 1196.

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 the ASET operating directory, `/usr/aset`.

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 1235.
- 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 may 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 root 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 Reports on a Server" on page 1235.

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 root privilege or not. Alternatively, you can leave the reports directory on the local system when you want users to monitor their own ASET reports.

Environment Variables

Table 57-10 lists the ASET environment variables and the values that they specify.

*Table 57-10*Environment Variables and Their Meanings

Environment Variable	**Specifies**
`ASETDIR` (See below)	ASET working directory
`ASETSECLEVEL` (See below)	Security level
`PERIOD_SCHEDULE`	Periodic schedule
`TASKS`	Tasks to run
`UID_ALIAS`	Aliases file
`YPCHECK`	Extends check to NIS and NIS+
`CKLISTPATH_LOW`	Directory lists for low security
`CKLISTPATH_MED`	Directory list for medium security
`CKLISTPATH_HIGH`	Directory list for high security

The environment variables listed below are found in the file `/usr/aset/asetenv`. The `ASETDIR` and `ASETSECLEVEL` variables are optional and can be set only through the shell 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*

ASETSECLEVEL 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*"

- *minutes hours* Specifies start time in number of minutes after the hour (0-59) 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. Entries in all three tune files have the following format:

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 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 the reference page for the shell command sh(1).

- *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   staff  directory
/bin        00777  root   staff  symlink
/etc        02755  root   staff  directory
/etc/chroot 00777  root   staff  symlink
/etc/clri   00777  root   staff  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*=`...`

Where:

uid   is the shared user ID.

aliasn   is the 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
```

Securing Files 58

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	*page 1200*
How to Change the Owner of a File	*page 1202*
How to Change a Group Ownership of a File	*page 1203*
How to Change Permissions in Absolute Mode	*page 1206*
How to Change Permissions in Symbolic Mode	*page 1207*
How to Set Special Permissions in Absolute Mode	*page 1209*
How to Find Files With setuid Permissions Set	*page 1210*
How to Set ACL Entries on a File	*page 1215*
How to Check If a File Has an ACL	*page 1217*
How to Add or Modify ACL Entries on a File	*page 1218*
How to Delete ACL Entries From a File	*page 1219*
How to Display ACL Entries for a File	*page 1220*

For overview information about securing files, see “File Security” on page 1160.

58

Displaying File Information

How to Display File Information

Display information about all the files in a directory by using the `ls` command.

```
$ ls -la
```

In this command,

`-l`	Displays the long format.
`-a`	Displays all files, including files that begin with a dot (.).

Each line in the display has the following information about a file:

- Type of file and its permissions
- Number of hard links
- Owner of the file
- Group of the file
- Size of the file, in bytes
- Date the file was created or the 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 7504
drwxrwxr-x   2 root     sys          512 Mar  6  1994 .
drwxr-xr-x  24 root     root        1024 May 15 19:41 ..
-r-xr-xr-x   1 bin      bin       111632 Sep 27  1993 autopush
-rwxr-xr-x   1 root     other     258452 May 26  1993 bpgetfile
-r-xr-xr-x   1 bin      bin       305424 Sep 27  1993 hostconfig
-r-xr-xr-x   1 bin      bin       484464 Sep 27  1993 ifconfig
-r-xr-xr-x   1 root     sys       565204 Sep 27  1993 init
-r-xr-xr-x   2 bin      root      180264 Sep 27  1993 jsh
-r-xr-xr-x   1 bin      bin       137732 Sep 27  1993 mount
-r-xr-xr-x   1 root     sys         7140 Jan  1  1970 mountall
```

Changing File Ownership

▼ How to Change the Owner of a File

1. **If you are not the owner of the file or directory, become root.**
   Only the current owner or root can use the `chown` command to change the owner of a file or directory.

2. **Change the owner of a file by using the `chown` command.**

```
$ chown newowner filename
```

In this command,

newowner	Is the name of the new owner of the file or directory.
filename	Is the file or directory.

Verification—Changing the Owner of a File

```
$ ls -l
```

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  5  1994 myfile.doc
```

How to Change a Group Ownership of a File

1. **If you are not the owner of the file or directory, become root.**
   Only the current owner or root can use the `chgrp` command to change the group of a file or directory.

2. **Change the group owner of a file by using the `chgrp` command.**

```
$ chgrp group filename
```

In this command,

group Is the name of the new group of the file or directory.

filename Is the file or directory.

Verification—Changing a Group Ownership of a File

```
$ ls -g
```

Example—Changing a Group Ownership of a File

The following example sets the group ownership on `myfile` to the group `scifi`.

```
$ chgrp scifi myfile
$ ls –lg myfile
-rwxrw–rw– 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 root 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 and is 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.

Table 58-1 lists the octal values for setting file permissions in absolute mode.

Table 58-1 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

Table 58-2 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, or the permissions being assigned or changed.

Table 58-2 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

▼ How to Change Permissions in Absolute Mode

1. **If you are not the owner of the file or directory, become root.**
   Only the current owner or root 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
```

In this command,

nnn Specifies the octal values that change permissions on the file or directory. See Table 58-1 on page 1204 for the list of valid octal values.

filename Is the file or directory.

Verification—Changing Permissions in Absolute Mode

```
$ ls -l
```

Example—Changing Permissions in Absolute Mode

The following example sets `rwxr-xr-x` permissions on `myfile`.

```
$ chmod 755 myfile
```

How to Change Permissions in Symbolic Mode

1. **If you are not the owner of the file or directory, become root.**
   Only the current owner or root 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 perms   filename
   ```

   In this command,

   who operator perms Specifies the symbols that change the permissions on the file or directory. *who* specifies whose permissions are changed, *operator* specifies the operation to perform, and *perms* specifies what permissions are changed.

   See Table 58-2 on page 1205 for the list of valid symbols.

   filename Is the file or directory.

Verification—Changing Permissions in Symbolic Mode

```
$ ls -l
```

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
```

Setting and Searching for Special 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. Table 58-3 lists the octal values to set special permissions on a file.

Table 58-3 Setting Special Permissions in Absolute Mode

Octal Value	**Special Permissions Set**
`1`	Sticky bit
`2`	`setguid`
`4`	`setuid`

Table 58-2 on page 1205 lists the symbols to change the special permissions in symbolic mode.

How to Set Special Permissions in Absolute Mode

1. **If you are not the owner of the file or directory, become root.**
   Only the current owner or root 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
```

In this command,

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 58-3 on page 1208 for the list of valid octal values for the special permissions.

filename Is the file or directory.

Verification—Setting Special Permissions in Absolute Mode

```
$ ls -l
```

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     dbstaff       24576 May  6 09:30 dbprog
```

The following example sets sticky bit permission on the `pubdir` directory.

```
$ chmod 1777 pubdir
```

▼ How to Find Files With `setuid` Permissions Set

1. **Become root.**
2. **Find files with `setuid` permissions set by using the `find` command.**

```
# find directory -user root -perm -4000 -exec ls -ldb {}\; >/tmp/filename
```

In this command,

`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`, see “setuid Permission” on page 1162.

Example—Finding Files With `setuid` *Permissions Set*

```
    # 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
❶   ---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
    #
```

❶ 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.

Using ACLs

ACL entries are the way to define an ACL on a file, and they are set through the ACL commands. ACL entries consist of the following fields separated by colons:

```
entry_type:[uid|gid]:perms
```

In an ACL entry,

entry_type Is a 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 Is the user name or identification number.

gid Is the 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-
```

ACL Entries for Files

Table 58-4 lists the valid ACL entries. The first three ACL entries provide the basic UNIX file protection.

Table 58-4 ACL Entries for Files

ACL Entry	Meaning
`u[ser]::`*perms*	The owner's permissions.
`g[roup]::`*perms*	Permissions for the owner's group.
`o[ther]:`*perms*	Permissions for users other than the owner or members of the owner's group.
`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 may have write / execute permissions.
`u[ser]:`*uid*`:`*perms*	Permissions for a specific user.
`g[roup]:`*gid*`:`*perms*	Permissions for a specific group.

ACL Entries for Directories

In addition to the ACL entries described in Table 58-4, you can set default ACL entries on a directory that will apply to files created within the directory. Files created in a directory that has default ACL entries will have the same ACL entries as the default ACL entries. Table 58-5 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 owner, owner's group, others, and the mask (these are required and are the first four default ACL entries in Table 58-5).

Table 58-5 Default ACL Entries for Directories

Default ACL Entry	**Meaning**
`d[efault]:u[ser]::`*perms*	Default owner's permissions.
`d[efault]:g[roup]::`*perms*	Default permissions for the owner's group.
`d[efault]:o[ther]:`*perms*	Default permissions for users other than the owner or members of the owner's group.
`d[efault]:m[ask]:`*perms*	Default ACL mask.
`d[efault]:u[ser]:`*uid*`:`*perms*	Default permissions for a specific user.
`d[efault]:g[roup]:`*gid*`:`*perms*	Default permissions for a specific group.

How to Set ACL Entries on a File

Set ACL entries on a file by using the `setfacl` command.

```
$ setfacl -s user::perms,group::perms,other:perms,mask:perms,acl_entry_list filename1 [filename2...]
```

In this command,

`-s` Replaces the entire ACL with the new ACL entries, if an ACL already exists on the file.

`user::`*perms* Specifies the owner’s permissions.

`group::`*perms* Specifies the permissions for the owner’s group.

`other:`*perms* Specifies the permissions for users other than the owner or members of the owner’s 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 Is 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 58-4 and Table 58-5 show the valid ACL entries.

filename Is the file or directory on which to set the ACL entries.

Caution – If an ACL already exists on the file, the `-s` option will replace the entire ACL with the new ACL entries.

Verification—Setting ACL Entries on a File

To verify that an ACL was set on the file, see “How to Check If a File Has an ACL” on page 1217. To verify which ACL entries were set on the file, use the `getfacl` command.

```
$ getfacl filename
```

Examples—Setting ACL Entries on a File

The following example sets the user permissions to read / write, 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 is 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 user permissions to read / write / execute, group permissions to read only, and other permissions to none on the `ch2.doc` file. In addition, users in the `sysadmin` group are given read / write permissions on the file, and the ACL mask permissions is set to read / write.

```
$ setfacl -s u::7,g::4,o:0,g:sysadmin:6,m:6 ch2.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
```

In this command,

filename Is the file or directory that you want to check.

A '+' to the right of the mode field indicates the file has an ACL.

Example—Checking If a File Has an ACL

The following example shows that `ch1.doc` has an ACL.

```
$ ls -l ch1.doc
-rwxr-----+  1 nathan   sysadmin      167 Nov 11 11:13 ch1.doc
```

How to Add or Modify ACL Entries on a File

Add or modify ACL entries on a file by using the `setfacl` command.

```
$ setfacl -m acl_entry_list filename1 [filename2...]
```

In this command,

acl_entry_list Is the list of one or more ACL entries to add or modify on the file or directory. You can also add or modify default ACL entries on a directory. Table 58-4 and Table 58-5 show the valid ACL entries.

filename Is the file or directory on which to add or modify ACL entries.

Verification—Adding or Modifying ACL Entries on a File

To verify that the ACL entries were added or modified on the file, use the `getfacl` command.

```
$ getfacl filename
```

Examples—Adding or Modifying ACL Entries on a File

The following example adds read/write permissions for the user `george` on the `ch3.doc` file.

```
$ setfacl -m user:george:6 ch3.doc
```

The following example adds default ACL entries for the `book` directory, which already has a default entry specified for the owner of the directory, for the group owner of the directory, and for others. The users in the `staff` group are given read permissions and the required default mask is set to read/write.

```
$ setfacl -m default:group:staff:4,default:mask:6 book
```

How to Delete ACL Entries From a File

Delete ACL entries from a file by using the `setfacl` command.

```
$ setfacl -d acl_entry_list filename1 [filename2...]
```

In this command,

acl_entry_list Is 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 58-4 and Table 58-5 show the valid ACL entries.

filename Is the file or directory from which to delete the ACL entries.

Alternately, you can use the `-s` option of `setfacl` to delete all the ACL entries on a file and replace them with the new ACL entries specified.

Verification—Deleting ACL Entries From a File

To verify that the ACL entries were deleted from the file, use the `getfacl` command.

```
$ getfacl filename
```

Example—Deleting ACL Entries From a File

The following example deletes the ACL entry for the user `george` from the `ch3.doc` file.

```
$ setfacl -d user:george ch3.doc
```

▼ How to Display ACL Entries for a File

Display ACL entries for a file by using the `getfacl` command.

```
$ getfacl [-a | -d] filename1 [filename2...]
```

In this command,

`-a` Displays the file name, owner, group, and ACL entries for the specified file or directory.

`-d` Displays the file name, owner, group, and default ACL entries for the specified directory.

filename Is the file or directory for which to display the ACL entries.

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:rw-          #effective:rw-
group::r--               #effective:r--
mask:rw-
other:---
```

The following example shows the default ACL entries for the `book` directory.

```
$ getfacl -d book

# file: book
# owner: nathan
# group: sysadmin
default:user::rw-
default:user:george:r--
default:group::rw-
default:mask:rw-
default:other:r--
```

Securing Systems 59

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	*page 1224*
How to Display Users With No Passwords	*page 1225*
How to Save Failed Login Attempts	*page 1226*
How to Create a Dial-up Password	*page 1227*
How to Temporarily Disable Dial-up Logins	*page 1229*
How to Restrict Root Login to the Console	*page 1229*
How to Monitor Who Is Using the su Command	*page 1229*
How to Display Root Access Attempts to the Console	*page 1230*

For overview information about securing systems, see "System Security" on page 1165.

How to Display a User’s Login Status

1. **Become root.**
2. **Display a user’s login status by using the `logins` command.**

```
# logins -x -l username
```

In this command,

`-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 as a comma-separated list.

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 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
```

In this example,

`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
- the number of days required between changes
- the number of days allowed before a change is required
- the warning period

▼ How to Display Users With No Passwords

You should make sure that all users have a valid password.

1. **Become root.**
2. **Display users that have no passwords by using the** `logins` **command.**

```
# logins -p
```

In this command,

`-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 With No Passwords

The following example displays that the user `pmorph` does not have a password.

```
# logins -p
pmorph          501     other           1       Polly Morph
#
```

▼ How to Save Failed Login Attempts

1. **Become root.**
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
```

Verification—Saving Failed Login Attempts

To make sure the log works, attempt to log in to the a system five times with the wrong password after the `loginlog` file is created. Then display the `/var/adm/loginlog` file.

```
# more /var/adm/loginlog
pmorph:/dev/pts/0:Fri Jan 13 08:55:23 1995
pmorph:/dev/pts/0:Fri Jan 13 08:55:31 1995
pmorph:/dev/pts/0:Fri Jan 13 08:55:39 1995
pmorph:/dev/pts/0:Fri Jan 13 08:55:50 1995
pmorph:/dev/pts/0:Fri Jan 13 08:56:00 1995
#
```

▼ 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 root.**

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
   ```

   See steps 7 and 8 for information on how to obtain the encrypted passwords.

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 root.**
2. **Put the following entry by itself into the** `/etc/d_passwd` **file:**

```
/usr/bin/sh:*:
```

How to Restrict Root Login to the Console

1. **Become root.**
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 root.

Verification—Restricting Root Login to the Console

Attempt to log in remotely as root to this system, and verify that the operation fails.

How to Monitor Who Is Using the `su` Command

1. **Become root.**
2. **Edit the** `/etc/default/su` **file.**
3. **Uncomment the following line.**

```
SULOG=/var/adm/sulog
```

Verification—Monitoring Who Is Using the `su` *Command*

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 Root Access Attempts to the Console

1. **Become root.**
2. **Edit the** `/etc/default/su` **file.**
3. **Uncomment the following line.**

```
CONSOLE=/dev/console
```

Verification—Displaying Root Access Attempts to the Console

Use the `su` command to become root, and verify that a message is printed on the system console.

Running ASET 60

This chapter describes how to run the Automated Security Enhancement Tool (ASET) to monitor or restrict access to system files and directories.

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

How to Run ASET Interactively	*page 1232*
How to Run ASET Periodically	*page 1234*
How to Stop Running ASET Periodically	*page 1235*
How to Collect Reports on a Server	*page 1235*

For overview information about ASET, see “Automated Security Enhancement Tool (ASET)” on page 1177.

▼ How to Run ASET Interactively

1. **Become root.**
2. **Run ASET interactively by using the `aset` command.**

```
# /usr/aset/aset –l level  -d pathname
```

In this command,

level Specifies the level of security. Valid values are `low`, `medium`, or `high`. The default setting is `low`. See “ASET Security Levels” on page 1178 for detailed information about security levels.

pathname Specifies the working directory for ASET. The default is `/usr/aset`.

ASET starts running. The execution log message is displayed on the screen, telling you 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
    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 root.**

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 1194 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
   ```

   In this command,

   `-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.

Verification—Running ASET Periodically

The following command displays the `crontab` entry, which enables you to confirm the schedule of when ASET will run.

```
# crontab –l root
```

How to Stop Running ASET Periodically

1. **Become root.**
2. **Edit the** `crontab` **file.**

```
# crontab -e root
```

3. **Delete the ASET entry.**
4. **Save the changes and exit.**

How to Collect Reports on a Server

1. **Set up a directory on the server:**

   a. **Type** `cd /usr/aset` **and press Return.**

   b. **Type** `mkdir` *rptdir* **and press Return.**
   These two commands create a directory (*rptdir*) on the server for report collection.

   c. **Type** `cd` *rptdir* **and press Return.**

   d. **Type** `mkdir` *client_rpt* **and press Return.**
   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
```

2. **Share the client subdirectories.**
   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
```

3. **Type `shareall` and press Return.**
   This makes the resources in the `dfstab` file available to the clients.

4. **Type the following command on each client:**
   `mount server:/usr/aset/`*client_rpt* `/usr/aset/masters/reports`

   This mounts the client subdirectory (`/usr/aset/`*client_rpt*) from the server to the client, at the mount point, `/usr/aset/masters/reports`.

5. **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
```

ASET Troubleshooting 61

This appendix documents the error messages generated by ASET. The error messages are shown in `Courier font` and are listed in alphabetical order. For each message, the following is given:

Meaning: This section clarifies and expands the message.

Action: This section explains how to resolve or fix the problem that is causing the message to be issued.

ASET Error Messages

```
ASET failed: no mail program found.
```

Meaning: 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.
```

Meaning: 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 `ASETSECLEVEL` environment 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.
```

Meaning: 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.
```

Meaning: The ASET working (operating) directory is not defined, or it is defined incorrectly. This may 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.
```

Meaning: 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.
```

Meaning: 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.
```

Meaning: ASET cannot locate an `asetenv` file in its working directory.

Action: Ensure there is an `asetenv` file in ASET's working directory. See the `asetenv(4)` manual page for the details about this file.

```
filename doesn't exist or is not readable.
```

Meaning: 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.
```

Meaning: 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 the `asetenv(4)` manual page for the content of a good `asetenv` file.

```
ASET task list $TASKLIST missing.
ASET startup unsuccessful.
```

Meaning: 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 the asetenv(4) manual page for the content of a good asetenv file.

```
Schedule undefined for periodic invocation.
No tasks executed or scheduled. Check asetenv file.
```

Meaning: 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.
```

Meaning: 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 crontab(1) command interface, that the correct schedule is in effect. Ensure that no unnecessary crontab entries for ASET are in place.

Part 14 — Managing System Resources

This part provides instructions for managing system resources in the Solaris environment.

62

Overview of System Resource Management

Provides overview information about Solaris commands and utilities that help you manage system resources by using crash dumps, disk quotas, accounting programs, and `cron` and `at` commands.

63

Examining and Changing System Information

Provides step-by-step instructions for examining and changing common system information, including the Workstation Info menu.

64

Saving Crash Dumps

Provides step-by-step instructions for saving crash dumps, and viewing crash and system messages.

65

Managing Disk Use

Provides step-by-step instructions for optimizing disk space by locating unused files and large directories.

66

Managing Quotas

Provides step-by-step instructions for setting up and administering disk quotas.

67

Setting Up and Maintaining Accounting

Provides step-by-step instructions for setting up and maintaining accounting.

68

Scheduling System Events

Provides step-by-step instructions for scheduling routine or one-time system events using `crontab` and `at`.

Overview of System Resource Management 62

This chapter contains information about features offered by UNIX software and the Solaris operating environment to help you manage system resources by using crash dumps, disk quotas, accounting programs, and `crontab` and `at` commands that automatically run routine commands.

This is a list of the overview information in this chapter.

System Crashes	*page 1244*
Quotas	*page 1247*
Accounting Utilities	*page 1248*
Executing Routine Tasks Automatically	*page 1266*

For information about other UNIX and Solaris features that help regulate system performance, see "Managing System Performance" on page 1365.

For instructions on how to set up crash dumps, see Chapter 64, "Saving Crash Dumps."

For instructions on how to set up, maintain, and disable user quotas, see Chapter 66, "Managing Quotas."

For instructions on how to set up and maintain accounting utilities, see Chapter 67, "Setting Up and Maintaining Accounting."

For instructions on how to set up `crontab` files and `at` commands, see Chapter 68, "Scheduling System Events."

System Crashes

System crashes can occur due to hardware malfunctions, power failures, I/O problems, and software errors. If a software glitch, such as a fatal kernel error caused by an operating system bug, causes a system to crash, the system writes an image of its physical memory into a `core` file at the end of the swap slice of the disk. This file is a snapshot of the state of the kernel, including its program text, data, and control structures, captured at the time of the crash.

Crash Dump Files

The `core` file written when a UNIX system crashes can provide clues as to what caused the crash if it is examined by an experienced customer service representative who is familiar with the `crash` kernel debugger. However, when a UNIX system reboots after a crash, it generally overwrites any `core` file that may have been produced—unless you have enabled the system to save the `core` file in a crash dump file.

Because UNIX systems do not automatically save `core` files in crash dump files, if you want to save a system's crash dump files for later inspection, you must set up the system by creating a `crash` directory in `/var/crash`, reserving disk space, and editing `/etc/init.d/sysetup` to store crash dump files. Crash dump files can be very big, so do not retain them longer than necessary.

See Chapter 64, "Saving Crash Dumps," for instructions on how to enable a system to save crash dump files.

Message Files

After Solaris software installation, the error logging daemon, `syslogd`, automatically records various system warnings and errors in message files. By default, many of these system messages are stored in `/var/adm` (or `/usr/adm`) or displayed on the system console. You can direct where these messages are stored by setting up system logging. See "How to Customize System Logging" on page 1293 for more information. These messages can alert you to system problems, such as a device that is about to fail.

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 1311 and “Scheduling System Events” on page 1343 for more information on automating this task.

What to Do After a Crash

If a system crashes, making it run again may seem like your most pressing concern. However, before you reboot the system, examine the console screen for messages. These messages may provide some insight as to what caused the crash. Even if the system reboots automatically and the console messages have disappeared from the screen, you may 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 Crash and Boot Messages” on page 1288 for more information about viewing system error log files.

If you are having frequent crashes and are unable to determine their cause, before calling for help, 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.

What to Do If Rebooting Fails

After a crash, the system may reboot automatically. If the automatic reboot fails with a message such as:

```
reboot failed: help
```

then run `fsck` in single-user mode. For more information, see “Managing File Systems” in *System Administration Guide, Volume I*, and the `fsck(1M)` man page.

If the system does not reboot, or if it reboots and then crashes again, there may be a hardware problem with a disk or one of the boards.

Check your hardware connections:

- Make sure the equipment is plugged in.
- Make sure all the switches are in the proper settings and pushed all the way in.
- 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 you cannot find any obvious fault with the connections, and the system still doesn't respond, contact your local service provider.

What to Do If a System Hangs

Your system may freeze or hang rather than crash completely. If this is the case, use this checklist:

- Make sure the pointer is in the window where you are typing the commands.
- Press Control-q in case the user accidently pressed Control-s, which freezes the screen. Note that, in a windowing environment, Control-s freezes only the window, not the entire screen. If a window is frozen, try using another window.
- Press Control-\ to force a "quit" in the running program and (probably) write out a `core` file.
- Press Control-c to interrupt the program that may be running.
- If possible, log onto the system from another terminal or log in remotely from another system on the network. Type `ps -ef` and look for the hung process. If it looks like the window system is hung, find the process and kill it.
- Try becoming root and rebooting the system.
- If the system still does not respond, force a crash dump and reboot. See "Shutting Down and Booting a System" in *System Administration Guide, Volume I* for information on forcing a crash dump and booting.
- If the system still does not respond, turn the power off, wait a minute or so, then turn the power back on. This procedure is frequently called *power cycling*.

- If you cannot get the system to respond at all, contact your local service provider for help.

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 invovles several 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 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" on page 1314.

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 1323 for procedures that describe how to change quotas or the amount of time that quotas can be exceeded, 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 1321.

Accounting Utilities

The SunOS 5.x accounting utilities are programs that collect and record data about system usage and then provide reports. The accounting utilities can be used for:

- Monitoring system usage
- Troubleshooting
- Locating and correcting performance problems
- Maintaining system security

Once they have been set up, the system accounting facilities run mostly on their own. (For instructions on setting up accounting facilities, see Chapter 67, "Setting Up and Maintaining Accounting.")

The accounting utilities provide 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. Setting up automatic accounting involves putting the scripts into `crontab` files so that `cron` can invoke them automatically.

The following is an overview of how accounting works.

1. Between system startup and shutdown, raw data about system use (such as logins, processes run, 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.

Types of Daily Accounting

Daily accounting can help you do 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 file `/var/adm/wtmp`. Entries in the `wtmp` 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:

- The user and group IDs of those using the process
- The beginning and elapsed times of the process
- The CPU time for the process (user time and system time)
- The amount of memory used
- The commands run
- The `tty` controlling the process

Every time a process dies, the `exit` program collects this data and writes it to `/var/adm/pacct`.

The `pacct` file has a default maximum size of 500 blocks that is enforced by the accounting shell script, `ckpacct` (normally run as a `cron` job). If `ckpacct` finds that `/var/adm/pacct` is larger than 500 blocks, it moves the file to `/var/adm/pacct`*n*, where *n* is the next unused incremental number.

Disk Accounting

Disk accounting enables you to gather and format the following data about the files each user has on disks:

- The name and ID of the user
- The 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 `cron` command you add to the `/var/spool/cron/crontabs/root` file. See "How to Set Up Accounting" on page 1336 for more information about setting up `dodisk`.

In turn, `dodisk` invokes the commands `acctdusg` and `diskusg`, which gather information for each file in the system.

`acctdusg` 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` 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.

`diskusg` 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

If you provide special user services on a request basis, such as restoring files and remote printing, you may want to bill users by running a facility called `chargefee`. For example, `chargefee` can be used to bill a user for restoring a directory from tape. Each time a specified user logs in, the `chargefee` utility records an entry consisting of the login name, user ID, and the set fee in the file `/var/adm/fee`. Each time the `runacct` utility is executed, new entries are picked up and merged into the total accounting records. For instructions on setting up `chargefee` to bill users, see "How to Bill Users" on page 1338.

How Daily Accounting Works

Here is a step-by-step summary of how SunOS 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/wtmp. In this record, the system name is shown as the login name in the wtmp record. Table 62-1 presents a summary of how the raw accounting data is gathered and where it is stored.

Table 62-1 Raw Accounting Data

File in /var/adm	Information	Written By	Format
wtmp	Connect sessions	login, init	utmp.h
	Changes	date	
	Reboots	acctwtmp	
	Shutdowns	shutacct shell	
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 argument /var/adm/pacct.

4. The remove shell script “cleans up” the saved pacct and wtmp files left in the sum directory by runacct.

5. The login and init programs record connect sessions by writing records into /var/adm/wtmp. Any date changes (using date with an argument) are also written to /var/adm/wtmp. Reboots and shutdowns using acctwtmp are also rec‘orded in /var/adm/wtmp.

6. When a process ends, the kernel writes one record per process, in the form of acct.h, in the /var/adm/pacct file.

7. runacct is executed by cron each night. runacct processes the accounting files: /var/adm/pacct*n*, /var/adm/wtmp, /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.

Two programs track disk usage by login: acctdusg and diskusg. They are invoked by the shell script dodisk.

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.

If the system is shut down using shutdown, the shutacct program is executed automatically. The shutacct program writes a reason record into /var/adm/wtmp and turns off process accounting.

Daily Accounting Reports

The runacct 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 four basic reports generated are:

- *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 command summary.
- *Last Login* – *S*hows the last time each user logged in (arranged in chronological order).

The following sections describe the reports and the meaning of the data presented in each one.

Daily Report

This report gives information about each terminal line used. A sample daily report appears below.

```
Jun 26 09:53  1994 DAILY REPORT FOR sfxbs Page 1

from      Thu Jun 25 17:45:22 1994
to        Fri Jun 26 09:51:25 1994
1         runacct
1         acctcon

TOTAL DURATION IS 966 MINUTES
LINE      MINUTES     PERCENT # SESS # ON  # OFF
term/23   25          3       7      7     3
term/22   157         16      6      6     3
TOTALS    183         --      13     13    7
------------------------------------------------------
```

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/wtmp` by the `acctwtmp` program. For more, 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 Table 62-2.

Table 62-2 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.
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. It used to list 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 when 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/wtmp because it is the file from which the connect accounting is geared. If the wtmp file grows rapidly, execute acctcon –l *file* < /var/adm/wtmp to see which tty line is the noisiest. If the interrupting is occurring frequently, general system performance will be affected. Additionally, wtmp may become corrupted. To correct this, see “How to Fix a wtmp File” on page 1339.

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.

```
Jun 29  09:53  1994   DAILY USAGE REPORT FOR sfxbs 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    5     12     6     16     131     51       0    1114   13      0     0
0    root     2      8     1     11       0      0       0     519    0      0     0
3    sys      0      1     0      1       0      0       0      45    0      0     0
4    adm      0      2     0      1       0      0       0     213    0      0     0
5    uucp     0      0     0      0       0      0       0      53    0      0     0
999  rly      3      1     5      2     111     37       0     269    1      0     0
7987 jan      0      0     0      1      20     14       0      15    6      0     0
```

The data provided in the daily usage report is described in Table 62-3.

Table 62-3 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 file `/etc/acct/holidays`.
`KCORE-MINS`	A cumulative measure of the amount of memory in kilobyte 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.

Table 62-3 Daily Usage Report Data *(Continued)*

Column	Description
`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 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.

```
 Jun 29 09:52 1994 DAILY COMMAND SUMMARY Page 1

    TOTAL COMMAND SUMMARY
         PRIME          PRIME   PRIME
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  1114      2.44    16.69  136.33     0.15    0.01      0.12  4541666      1926

sh       227      1.01     2.45   54.99     0.41    0.01      0.04   111025       173
vi        12      0.35     0.62   44.23     0.55    0.05      0.01   151448        60
sed      143      0.09     0.82    1.48     0.10    0.01      0.55    14505        35
sadc      13      0.08     0.19    1.45     0.44    0.01      0.13   829088        19
more       3      0.04     0.07    2.17     0.59    0.02      0.03    30560         1
cut       14      0.03     0.09    0.28     0.37    0.01      0.33      154        13
uudemon.  76      0.03     0.66    2.30     0.05    0.01      0.29    43661        13
uuxqt     29      0.03     0.30    0.72     0.08    0.01      0.42    80765        35
mail       4      0.02     0.06    0.09     0.37    0.01      0.60     4540         9
ckstr     21      0.02     0.11    0.13     0.17    0.01      0.85        0         4
awk       13      0.02     0.12    0.21     0.15    0.01      0.54      444         2
ps         2      0.02     0.10    0.13     0.17    0.05      0.77     8060        21
find       9      0.02     3.35    5.73     0.00    0.37      0.58   355269       760
sar        1      0.01     0.19    0.24     0.08    0.19      0.80   564224         4
acctdisk   2      0.01     0.01    0.06     1.02    0.01      0.22        0         9
mv        24      0.01     0.14    0.17     0.10    0.01      0.81     3024        36
  .
  .
  .
```

The data provided, by column, in the daily command summary is described in Table 62-4.

Table 62-4 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 root privileges were used.
PRIME 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.
PRIME TOTAL CPU-MIN:	Total processing time this program has accumulated during prime time.
PRIME 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.

```
    TOTAL COMMAND SUMMARY

COMMAND  NUMBER  TOTAL      TOTAL      TOTAL     MEAN   MEAN    HOG     CHARS        BLOCKS
NAME       CMDS KCOREMIN  CPU-MIN   REAL-MIN  SIZE-K CPUMIN FACTOR   TRNSFD         READ

TOTALS 301314  300607.70 4301.59  703979.81   69.88   0.01   0.01 6967631360   10596385

troff     480   58171.37  616.15    1551.26   94.41   1.28   0.40  650669248     194926
rnews    5143   29845.12  312.20    1196.93   95.59   0.06   0.26 1722128384    2375741
uucico   2710   16625.01  212.95   52619.21   78.07   0.08   0.00  228750872     475343
nroff    1613   15463.20  206.54     986.06   74.87   0.13   0.21  377563304     277957
vi       3040   14641.63  157.77   14700.13   92.80   0.05   0.01  116621132     206025
expire     14   13424.81  104.90     265.67  127.98   7.49   0.39   76292096     145456
comp     3483   12140.64   60.22     423.54  201.62   0.02   0.14    9584838     372601
ad_d       71   10179.20   50.02    1158.31  203.52   0.70   0.04   11385054      19489
as       2312    9221.59   44.40     285.52  207.68   0.02   0.16   35988945     221113
gone      474    8723.46  219.93   12099.01   39.67   0.46   0.02   10657346      19397
i10       299    8372.60   44.45     454.21  188.34   0.15   0.10   60169932      78664
find      760    8310.97  196.91     728.39   42.21   0.26   0.27   58966910     710074
ld       2288    8232.84   61.19     425.57  134.55   0.03   0.14  228701168     279530
fgrep     832    7585.34   62.62     199.11  121.14   0.08   0.31   22119268      37196
sh      56314    7538.40  337.60  291655.70   22.33   0.01   0.00   93262128     612892
du        624    5049.58  126.32     217.59   39.97   0.20   0.58   16096269     215297
ls      12690    4765.60   75.71     541.53   62.95   0.01   0.14   65759473     207920
vnews      52    4235.71   28.11     959.74  150.70   0.54   0.03   28291679      28285
  .
  .
  .
```

See “Daily Command Summary” on page 1256 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.

```
Feb 13 04:40 1994 LAST LOGIN Page 1

00-00-00    **rje** 88-01-01    jlr    88-02-09   cec42   88-02-13     cec20
00-00-00    **rje** 88-01-13    crom   88-02-10   jgd     88-02-13     cec22
00-00-00    3bnet   88-01-14    usg    88-02-10   wbr     88-02-13     cec23
00-00-00    adm     88-01-17    cec11  88-02-11   cec30   88-02-13     cec24
00-00-00    daemon  88-01-17    cec38  88-02-11   cec41   88-02-13     cec25
00-00-00    notes   88-01-17    cec40  88-02-11   cec43   88-02-13     cec26
00-00-00    oas     88-01-18    cec60  88-02-11   cec53   88-02-13     cec27
00-00-00    pds     88-01-19    cec35  88-02-11   cec54   88-02-13     cec3
00-00-00    polaris 88-01-19    cec37  88-02-11   cec55   88-02-13     cec31
00-00-00    rje     88-01-22    dmk    88-02-11   cec56   88-02-13     cec32
00-00-00    shqer   88-01-26    ask    88-02-11   cec57   88-02-13     cec4
00-00-00    sys     88-01-26    cec39  88-02-11   cec58   88-02-13     cec6
00-00-00    trouble 88-01-27    sync   88-02-11   jwg     88-02-13     cec7
00-00-00    usors   88-02-02    pkl    88-02-11   skt     88-02-13     cec8
00-00-00    uucp    88-02-03    ibm    88-02-11   tfm     88-02-13     commlp
00-00-00    wna     88-02-03    slk    88-02-12   cec21   88-02-13     djs
87-07-06    lp      88-02-04    cec59  88-02-12   cec28   88-02-13     epic
87-07-30    dgn     88-02-05    cec33  88-02-12   cec29   88-02-13     jab
87-08-19    blg     88-02-05    cec34  88-02-12   csp     88-02-13     jcs
87-12-08    emna    88-02-05    cec36  88-02-12   drc     88-02-13     mak
88-01-14    s       88-02-05    cec51  88-02-12   emw     88-02-13     dn
88-01-09    rib     88-02-05    dfh    88-02-12   je      88-02-13     mlp
88-01-25    dmf     88-02-05    fsh    88-02-12   kab     88-02-13     nbh
88-01-25    emda    88-02-05    pkw    88-02-12   rap     88-02-13     rah
  .
  .
  .
```

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 (# sign if it was executed with root 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

Table 62-5 describes the acctcom options.

Table 62-5 acctcom Options

Option	Description
–a	Show some average statistics about the processes selected. (The statistics are printed after the output is recorded.)
–b	Read the files backward, showing latest commands first. (This has no effect if reading standard input.)
–f	Print the fork/exec flag and system exit status columns. (The output is an octal number.)
–h	Instead of mean memory size, show 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	Print columns containing the I/O counts in the output.
–k	Show total kcore minutes instead of memory size.
–m	Show mean core size (this is the default).

Table 62-5 `acctcom` Options *(Continued)*

Option	Description
`–q`	Don’t print output records, just print average statistics.
`–r`	Show CPU factor: *user_time* / (*system_time* + *user_time*).
`–t`	Show separate system and user CPU times.
`–v`	Exclude column headings from the output.
`–C` *sec*	Show only processes with total CPU time (system plus user) exceeding *sec* seconds.
`–e` *time*	Show processes existing at or before *time*, given in the format *hr[:min[:sec]]*.
`–E` *time*	Show processes starting at or before *time*, given in the format *hr[:min[:sec]]*. Using the same *time* for both `–S` and `–E` shows processes that existed at the time.
`–g` *group*	Show only processes belonging to *group*.
`–H` *factor*	Show only processes that exceed *factor*, where *factor* is the “hog factor” (see the `–h` option).
`–I` *chars*	Show 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*	Show only commands matching *pattern* (a regular expression except that “+” means one or more occurrences).
`–o` *ofile*	Instead of printing the records, copy them in `acct.h` format to *ofile*.
`–O` *sec*	Show only processes with CPU system time exceeding *sec* seconds.
`–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*	Show only processes belonging to *user*.

Accounting Files

The /var/adm directory structure contains the active data collection files and is owned by the adm login (currently user ID of 4).

Table 62-6 Files in /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
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 62-7 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

Table 62-7 Files in the `/var/adm/acct/nite` Directory *(Continued)*

File	Description
`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`
`log`	Diagnostic output from `acctcon`
`log.`*MMDD*	Same as `log` after `runacct` detects an error
`owtmp`	Previous day's `wtmp` file
`reboots`	Beginning and ending dates from `wtmp` and a listing of reboots
`statefile`	Used to record current state during execution of `runacct`
`tmpwtmp`	`wtmp` 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 `wtmp` 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 in Table 62-8.

Table 62-8 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

Table 62-8 Files in the `/var/adm/acct/sum` directory *(Continued)*

File	Description
`tacct`	Cumulative total accounting file for current fiscal period
`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 in Table 62-9.

Table 62-9 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 `rprtn` for fiscal period *n*
`tacct`*n*	Total accounting file for fiscal period *n*

Fixing Corrupted Files and `wtmp` *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 the backup. However, certain files must be fixed to maintain the integrity of the accounting system.

The `wtmp` 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/wtmp`. The `wtmpfix` 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 `wtmp` problems, see "How to Fix a wtmp File" on page 1339.

62

Executing Routine Tasks Automatically

Many routine system events can be set up to execute automatically. Some of these tasks need to occur repetitively, at regular intervals. Other tasks need to run only once, perhaps during off hours such as evenings or weekends.

This section contains information about two commands, `crontab` and `at`, that 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.

How the `cron` Daemon Handles Scheduling

The `cron` daemon handles the automatic scheduling of `crontab` commands. Its function is to check the `/usr/spool/cron/crontab` directory (or the `/var/spool/cron/crontab` directory, depending on your system configuration) 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 /usr/spool/cron/atjobs directory.

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:

```
0 20 * * 0,4 /etc/cron.d/logchecker
5 4 * * 6 /usr/lib/newsyslog
15 3 * * * /usr/lib/fs/nfs/nfsfind
```

The first command line instructs the system to run logchecker at 10 p.m. on Sundays and Thursdays. The second command line schedules the system to run newsyslog at 4:05 a.m. every Sunday. The third command line orders the system to execute nfsfind daily at 3:15 a.m.

For more information about the syntax of lines within a crontab file, see “Syntax of crontab File Entries” on page 1344.

The crontab files are stored in /usr/spool/cron/crontabs (or /var/spool/cron/crontabs). Several crontab files besides root are provided during SunOS software installation (see Table 62-10).

Table 62-10 Default crontab Files

crontab **File**	**Function**
adm	Accounting
lp	Printing
root	General system functions and file system cleanup
sys	Performance collection

Other crontab files are named after the user accounts in which they are created, such as bob, mary, smith, or jones.

Besides the default crontab file, users can create crontab files to schedule their own system events. To access crontab files belonging to root or other users, root privileges are required.

Procedures explaining how to create, edit, display, and remove crontab files are described in "Commands for Scheduling System Events" on page 1344.

crontab *Command Security*

For additional security over the scheduling of system events, you can set up files that control access to the crontab command, permitting only specified users to create, edit, display, or remove their crontab jobs. Two files control access to crontab: /etc/cron.d/cron.deny and /etc/cron.d/cron.allow. Each of these files consists of a list of user names, one per line. These files work together to control user access to crontab commands.

Whenever a user tries to access crontab, the crontab facility finds and reads the cron.allow file first, if one exists. crontab then reads the cron.deny file.

- If the cron.allow file 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 access crontab commands except for those listed in cron.deny.
- If neither file exists, only root can run crontab.

Root privileges are required to edit or create cron.deny and cron.allow.

By default, no cron.allow file is created during SunOS software installation. However, a cron.deny file is created, containing the following user names:

```
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 crontab.

If you create a cron.allow file, only these users can access crontab commands.

Scheduling a Single Job: at

at 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 that schedule the time your job will be executed. For more information about submitting at jobs, see “at Command Description” on page 1356.

The at command stores the command or script that you entered, along with a copy of your current environment variables, in either /usr/spool/cron/atjobs or /var/spool/cron/atjobs. 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.

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 root privileges, you can edit this file to add other user names whose at access you want to restrict.

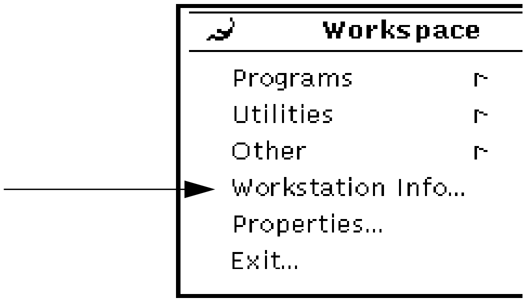
Examining and Changing System Information 63

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 Add "Workstation Info" to the Workspace Menu	*page 1272*
How to Display General System Information	*page 1274*
How to Display a System’s Host ID Number	*page 1275*
How to Display a System’s Installed Memory	*page 1275*
How to Display the Date and Time	*page 1276*
How to Synchronize Date and Time From Another System	*page 1278*
How to Set a System’s Date and Time Manually	*page 1279*
How to Set Up a Message of the Day	*page 1280*
How to Set the Number of Processes per User	*page 1281*
How to Increase the Number of Pseudo-ttys to 256	*page 1281*
How to Increase the Number of Lock Requests	*page 1282*
How to Increase Shared Memory Segments	*page 1282*

Using the Workstation Information Window to Display System Information

"Workstation Info . . ." in the OpenWindows Workspace menu . . .



. . . provides the following information about a system:

- Workstation name and type
- Host ID
- Internet address and network domain
- Physical memory (RAM) and virtual memory
- Operating system and window system versions

Some of the same information can be displayed with commands described later in this section, but this screen is a convenient way to display this particular collection of information.

If your OpenWindows Workspace menu does not include the Workstation Info command, see "How to Add "Workstation Info" to the Workspace Menu" on page 1272.

▼ How to Add "Workstation Info" to the Workspace Menu

1. **Use the editor of your choice to open** `.openwin-menu` **in your home directory.**

```
# vi .openwin-menu
```

2. **Add the following line to the** `.openwin-menu` **file.**

```
"Workstation Info ..."      exec $OPENWINHOME/bin/wsinfo
```

The placement of this line determines the placement of the command in the menu. In other words, if you place it above the Properties line, it will appear above the Properties command in the menu:

```
# openwin-menu - Openwindows x11/NeWS Server default root menu
.
.
.
"Utilities" MENU
     "Refresh"   DEFAULT  REFRESH
     "Reset Input"        POSTSCRIPT /resetinput ClassUI
     "Save Workspace"     SAVE_WORKSPACE
     "Lock Screen"        exec wcss -lock
     "Console"            exec $OPENWINHOME/bin/xview/cmdtool
"Utilities" END

"Workstation Info..."        exec $OPENWINHOME/bin/wsinfo

"Properties..."             PROPERTIES

"Exit..."                    EXIT
```

Add line here

3. **Exit the file, saving the changes.**

63

Using Commands to Display System Information

Table 63-1 shows man pages and descriptions for some commands that enable you to display general system information.

Table 63-1 Commands for Displaying System Information

Command	**Enables You to Display a System's …**
`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 Display General System Information

To display system information, use the `uname` command.

```
$ uname [-a]
```

In this command,

`uname` Displays only the name of the operating system.

`-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 sample output from the `uname` command on the system `jupiter`.

```
$ uname
SunOS
$ uname -a
SunOS jupiter 5.5 preview95 sunrm sparc
```

▼ 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
7725ac42
```

▼ 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]
```

In this command,

`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: 32 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 Apr 13 10:31:43 EST 1995
```

Using Commands to Change System Information

Table 63-2 shows man pages and descriptions for some commands that enable you to change general system information.

Table 63-2 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

By 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 pseudo-`tty`s to 256
- Increase the number of lock requests
- Increase shared memory segments

By default, the number of lock requests that may occur simultaneously is 512. As users log out, they lock files, including `utmp`. If more than 512 users are likely to log out simultaneously (within a few seconds), the number of file locks allowed must be increased.

▼ How to Synchronize Date and Time From Another System

1. **Become root.**
2. **To reset the date and time to synchronize with another other system, use the `rdate` command.**

```
# rdate other-system-name
```

In this command,

other-system-name Is the name of another system.

Verification—Synchronizing Date and Time From Another System

To verify that you have reset your system's date and time by using `rdate`, check 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 Neptune, which is running several hours behind, is reset to match the date and time of the server Pluto.

```
neptune$ date
Tue Mar 28 19:31:43 EST 1995
neptune$ rdate pluto
Tue Mar 28 22:00:00 EST 1995
neptune$ date
Tue Mar 28 22:00:00 EST 1995
```

▼ How to Set a System’s Date and Time Manually

1. **Become root.**
2. **Enter the new date and time.**

```
# date mmddHHMM[[cc]yy]
```

In this command,

mm	Is the month, using two digits.
dd	Is the day of the month, using two digits.
HH	Is the hour, using two digits and a 24-hour clock.
MM	Are the minutes, using two digits.
cc	Is the century, using two digits.
yy	Is the year, using two digits.

Verification—Setting a System’s Date and Time Manually

After you set the date and time manually, you can use `date` with no options to display your system’s new date and time to confirm that this information has changed.

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
Fri Dec 9 10:31:00 MST 1994
$ date 032318151995
Thu Mar 23 18:15:00 MST 1995
```

How to Set Up a Message of the Day

1. **Become root.**
2. **Open the** `/etc/motd` **file, using the editor of your choice.**
3. **Edit the text to include the message that will be displayed as part of the user login process, including spaces, Tabs, and Returns.**
4. **Exit the file, saving your changes.**

Verification—Setting Up a Message of the Day

To view the `/etc/motd` file, use the `cat` or `more` command.

```
$ 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:

```
Sun Microsystems Inc    SunOS 5.5      Generic       August 1995
```

The following example shows an edited `/etc/motd` file that provides information about system availabilty to each user who logs in.

```
The system will be down from 7:00 a.m to 2:00 p.m.on
Saturday, August 5, 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. **Open the** `/etc/system` **file, using the editor of your choice.**
2. **Add the following line to the file.**

```
set maxuprc=value
```

In this command,

value Is the number of processes a user can run at once.

3. **Exit the file, saving changes.**
4. **Reboot the system.**

Example—Setting the Number of Processes per User

The following example shows the line to add to the `/etc/system` file to allow users to run 10 processes each.

```
set maxuprc=10
```

How to Increase the Number of Pseudo-ttys to 256

1. **Open the** `/etc/system` **file, using the editor of your choice.**
2. **Add the following line to the file.**

```
set pt_cnt=256
```

3. **Exit the file, saving changes.**
4. **Instruct the system to reconfigure upon rebooting.**

```
$ touch /reconfigure
```

5. **Reboot the system.**

How to Increase the Number of Lock Requests

1. **Open the** `/etc/system` **file, using the editor of your choice.**
2. **Add the following line to the file to increase the number of lock requests (default is 512).**

```
set tune_t_flckrec=1024
```

3. **Exit the file, saving changes.**
4. **Reboot the system.**

How to Increase Shared Memory Segments

1. **Open the** `/etc/system` **file, using the editor of your choice.**
2. **Add the following lines to the file to 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 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
set semsys:seminfo_shmmin=200
set semsys:seminfo_shmmni=200
set semsys:seminfo_shmseg=200
```

3. **Exit the file, saving changes.**
4. **Reboot the system.**

Saving Crash Dumps 64

This section contains information about enabling and disabling crash dumps, and how to use the error messages generated when systems crash or boot. It also contains information about setting up system logging.

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

How to Create a Directory to Save Crash Dump Files	*page 1284*
How to Reserve Space for Crash Dump Files	*page 1285*
How to Enable Crash Dump Files	*page 1286*
How to Disable Crash Dump Files	*page 1287*
How to View Crash and Boot Messages	*page 1288*
How to Examine a Crash Dump	*page 1290*
How to Customize System Logging	*page 1293*

Enabling and Disabling Crash Dumps

During a system crash, the kernel dumps an image of its memory into a `core` file, which is overwritten during rebooting unless you set up the system to save it in a crash dump file. Enabling a system to save crash dumps involves:

1. Creating a crash dump directory.
2. Defining how much disk space to allow for a crash dump file.
3. Editing the `sysetup` file to activate the saving of crash dump files.

Disabling your system from saving crash dumps involves reversing these procedures. See “How to Disable Crash Dump Files” on page 1287 for more information.

▼ How to Create a Directory to Save Crash Dump Files

1. **Become root.**
2. **Create the `/var/crash` directory.**

```
# mkdir /var/crash
```

3. **Change to the `/var/crash` directory.**

```
# cd /var/crash
```

4. **Create a directory with the name of the system.**

```
# mkdir system-name
```

In this command,

system-name Is the system for which you want to save crash dump files.

Example—Creating a Directory to Save Crash Dump Files

The following example shows how to create a directory to save crash dump files for the system `saturn`.

```
# mkdir /var/crash
# cd /var/crash
# mkdir saturn
```

▼ How to Reserve Space for Crash Dump Files

1. **Become root.**
2. **Change to the `/var/crash/`*system-name* directory.**

   ```
   # cd /var/crash/system-name
   ```

   In this command,

   system-name Is the system for which you want to save crash dump files.

3. **Using the editor of your choice, create a file named `minfree` that contains a number specifying the minimum available free space (in kilobytes) that must remain available.**
4. **Exit the file, saving changes.**

Example—Reserving Space for Crash Dump Files

The following example shows the contents of a `minfree` file that reserves 500 Kbytes of available free space to contain crash dump files for the system `saturn`.

```
$ more /var/crash/saturn/minfree
500
```

▼ How to Enable Crash Dump Files

1. **Become root.**
2. **Using the editor of your choice, edit the** `/etc/init.d/sysetup` **file, activating the lines that enable the crash dumps by deleting the comment marks (#) from the beginning of those lines.**
3. **Exit the file, saving the changes.**

Example—Enabling Crash Dump Files

The following example shows the appropriate section of the `/etc/init.d/sysetup` file that has been edited to enable crash dumps.

```
##
## Default is to not do a savecore
##
If [ ! -d /var/crash/‘uname -n‘ ]
then mkdir -m 0700 -p /var/crash/‘uname -n‘
fi
     echo ’checking for crash dump...\c ’
savecore /var/crash/‘uname -n`
     echo ’’
```

▼ How to Disable Crash Dump Files

1. **Become root.**
2. **Edit the `/etc/init.d/sysetup` file, inserting a comment mark (#) at the beginning of each of the lines shown below.**

```
#if [ ! -d /var/crash/‘uname -n‘ ]
#then mkdir -p /var/crash/‘uname -n‘
#fi
#                echo ‘checking for crash dump...\c ‘
#savecore /var/crash/‘uname -n‘
#                echo ''
```

3. **Save the changes.**
4. **Remove the file set up for crash dumps from the `/var/crash` directory.**

```
# rm -rf /var/crash/system-name
```

In this command,

system-name Is the name of the system which will no longer save crash dump files.

Viewing System Information Generated by a Crash

When a system crashes, it displays a message like this:

```
panic: error message
```

where *error message* is one of the panic error messages described in the `crash(1M)` man page.

Less frequently, this message may be displayed instead of the panic message:

```
Watchdog reset !
```

System messages like these are automatically stored in `/var/adm/messages` (or `/usr/adm/messages`) throughout the session. These messages are saved whether or not crash dumps are enabled for a system.

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 file `messages.0` 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.

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 using the `crash` utility. Additionally, crash dumps saved by `crash` can be useful to send to a customer service representative for analysis. Using `crash` to its full potential requires a detailed knowledge of the kernel, and is beyond the scope of this manual. For more details on the operation of the `crash` utility, see the `crash(1M)` manual page.

▼ How to View Crash and Boot Messages

Display messages generated by a system crash or booting 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 the `dmesg(1M)` man page.

Example—Viewing Crash and Boot Messages

The following example shows output from the `dmesg` command.

```
$ dmesg

Mar 29 15:11
SunOS Release 5.5 Version A [UNIX(R) System V Release 4.0]
copyright (c) 1983-1995, Sun Microsystems, Inc.
DEBUG enabled
WARNING: cannot load psm xpcimach
mem = 32376K (0x1f9e000)
avail mem = 25247744
root nexus = i86pc
Unable to install/attach drive ‘isa’
eisa0 at root
NOTICE: eisa: DMA buffer-chaining not enabled
NOTICE: IN i8042_acquire
NOTICE: out i8042_acquire
NOTICE: IN i8042_release
NOTICE: about to enable keyboard
NOTICE: out i8042_release
.
.
.
```

▼ How to Examine a Crash Dump

To examine crash dumps, use the `crash` utility.

```
# /usr/sbin/crash [-d crashdump-file] [-n name-list] [-w output-file]
```

In this command,

`-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.

Example—Examining a Crash Dump

The following example shows sample output using the `crash` utility. Information about status, and buffer, process, and queue size is displayed.

```
# /usr/sbin/crash
dumpfile = /dev/mem, namelist = /dev/ksyms, outfile = stdout
> status
system name:    SunOS
release:        5.5
node name:      saturn
version:        test
machine name:   i86pc
time of crash:  Wed Mar 29 16:48:06 1995
age of system:  9 day, 2 hr., 52 min.
panicstr:
panic registers:
        eip: 0     esp: 0
> size buf proc queue
116
1580
88
```

Customizing System Logging

You can capture error messages that are generated by various system processes by editing the `/etc/syslog.conf` file to set up system logging.

The `/etc/syslog.conf` file has two columns separated by tabs. The first column specifies the source of the error condition and its priority. The second column specifies the place where the errors are logged. The following example shows sample lines from an `/etc/syslog.conf` file.

```
user.alert                                          /dev/console
kern.err                                       /var/adm/messages
```

The message sources in the first column are specified by two parts separated by a dot (`.`). The first part is the source or *facility*, which describes the part of the system generating the message. The second part is the priority of the message. The most common sources are shown in Table 64-1. The most common priorities are shown in Table 64-2 in order of severity.

Table 64-1 Sources for `syslog.conf` Messages

Source	**Description**
`kern`	The kernel
`auth`	Authentication
`daemon`	All daemons
`mail`	Mail system
`lp`	Spooling system
`user`	User processes

Note – A maximum of 24 `syslog` sources (or facilities) can be activated in the `/etc/syslog.conf` file.

Table 64-2 Priorities for `syslog.conf` Messages

Priority	**Description**
`emerg`	System emergencies
`alert`	Error requires immediate correction
`crit`	Critical errors
`err`	Other errors
`info`	Informational messages
`debug`	Output used for debugging
`none`	Setting that doesn’t log output

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 Crash and Boot Messages” on page 1288.

▼ How to Customize System Logging

1. **Become root.**

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 “Customizing System Logging” on page 1291.**

3. **Exit the file, saving the changes.**

Example—Customizing System Logging

The following sample lines from the `/etc/syslog.conf` file that is provided during Solaris installation show that, by default, user errors are printed to the console and also are logged to the file `/var/adm/messages`.

Mail debugging output is logged to the file `/var/log/syslog`.

```
user.err                                            /dev/console
user.err                                       /var/adm/messages
user.alert                                      ‘root, operator’
mail.debug                                       /var/log/syslog
```

Managing Disk Use 65

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	*page 1296*
How to Display the Size of Files	*page 1299*
How to Find Large Files	*page 1301*
How to Find Files That Exceed a Given Size Limit	*page 1302*
How to Display the Size of Directories, Subdirectories, and Files	*page 1303*
How to Display the User Allocation of Local UFS File System	*page 1304*
How to List the Newest Files	*page 1306*
How to Find and Remove Old or Inactive Files	*page 1308*
How to Clear Out Temporary Directories	*page 1309*
How to Find and Delete core Files	*page 1310*
How to Delete Crash Dump Files	*page 1311*

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 the `df(1M)` man page.

▼ 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]
```

In this command,

`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 — Is a 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.

Note – For remotely mounted file systems, `“-1 files”` is displayed instead of the number of files.

Examples—Displaying Information About Blocks, Files, and Disk Space

In the following example, the file systems root (`/`), `/usr`, `/proc`, and `/tmp` are on the local disk. The other file systems are mounted by NFS and do not use local disk resources.

```
$ df
/               (/dev/dsk/c0t0d0s0):   21338 blocks   9592 files
/usr            (/dev/dsk/c0t0d0s6):   46722 blocks  34103 files
/proc             (/proc          ):       0 blocks    112 files
/tmp              (swap           ):   66696 blocks   3177 files
/root             (saturn:(pid132)):       0 blocks     -1 files
/home             (saturn:(pid132)):       0 blocks     -1 files
/src              (saturn:(pid132)):       0 blocks     -1 files
/nse              (saturn:(pid132)):       0 blocks     -1 files
/net              (saturn:(pid132)):       0 blocks     -1 files
```

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  22199    11530    8459     58%       /
/dev/dsk/c0t0d0s6  73399    50038    16031    76%       /usr
/proc              0        0        0        0%        /proc
swap               33364    8        33356    0%        /tmp
saturn:(pid132)    0        0        0        0%        /root
saturn:(pid132)    0        0        0        0%        /home
saturn:(pid132)    0        0        0        0%        /src
saturn:(pid132)    0        0        0        0%        /nse
saturn:(pid132)    0        0        0        0%        /net
```

The following example shows information about the same system as the previous example, but only UFS file system information is displayed. Although /proc and /tmp are local file systems, they are not UFS file systems (/proc is a PROCFS file system, and /tmp is a TMPFS file system).

```
$ df -F ufs
/                (/dev/dsk/c0t0d0s0):   21338 blocks    9592 files
/usr             (/dev/dsk/c0t0d0s6):   46722 blocks   34103 files
```

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):    21338 blocks   9592 files
                            total:     44398 blocks   11264 files
/usr               (/dev/dsk/c0t0d0s6):     46722 blocks   34103
files
                            total:    146798 blocks   37888 files
/proc            (/proc          ):        0 blocks     112 files
                             total:        0 blocks     140 files
/tmp             (swap           ):    66712 blocks    3177 files
                            total:     66728 blocks    3179 files
/root            (saturn:(pid132)):        0 blocks      -1 files
                             total:        0 blocks      -1 files
/home            (saturn:(pid132)):        0 blocks      -1 files
                             total:        0 blocks      -1 files
/src             (saturn:(pid132)):        0 blocks      -1 files
                             total:        0 blocks      -1 files
/nse             (saturn:(pid132)):        0 blocks      -1 files
                             total:        0 blocks      -1 files
/net             (saturn:(pid132)):        0 blocks      -1 files
                             total:        0 blocks      -1 files
```

Checking the Size of Files

You can check the size of files and sort them by using the `ls` command, and you can find files that exceed a size limit by using the `find` command. For more information, see the `ls(1)` and `find(1)` man pages.

▼ 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]
```

In this command,

`-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`, `wtmp`, and `wtmpx` are substantially larger than the other files in the `/var/adm` directory.

```
venus% cd /var/adm
venus% ls -l
total 434
-r--r--r--   1 root     other     585872 Jan 28 14:53 lastlog
drwxrwxr-x   2 adm      adm          512 Dec  1 16:35 log
-rw-r--r--   1 root     other        408 Jan 28 14:15 messages
-rw-r--r--   1 root     other        177 Jan 24 16:56 messages.0
-rw-r--r--   1 root     other        177 Jan 17 16:13 messages.1
-rw-r--r--   1 root     other          0 Jan  4 04:05 messages.2
-rw-r--r--   1 root     other        562 Jan  2 13:13 messages.3
drwxrwxr-x   2 adm      adm          512 Dec  1 16:35 passwd
drwxrwxr-x   2 adm      sys          512 Jan 28 11:38 sa
-rw-rw-rw-   1 bin      bin            0 Nov 26 10:56 spellhist
-rw-------   1 root     root        1319 Jan 28 14:58 sulog
-rw-r--r--   1 root     bin          288 Jan 28 14:53 utmp
-rw-r--r--   1 root     bin         2976 Jan 28 14:53 utmpx
-rw-rw-r--   1 adm      adm        12168 Jan 28 14:53 wtmp
-rw-rw-r--   1 adm      adm       125736 Jan 28 14:53 wtmpx
```

The following example shows that `lpNet` uses eight blocks and `lpsched` and `lpsched-1` use two blocks each.

```
venus% cd /var/lp/logs
venus% ls -s
total 14            2 lpsched-1    0 lpsched-4      0 requests-2
   8 lpNet          2 lpsched-2    0 requests
   2 lpsched        0 lpsched-3    0 requests-1
venus%
```

▼ How to Find Large Files

1. **Change directory to where you want to search.**

2. **Display the size of files in blocks from largest to smallest.**

```
$ ls -s | sort -nr | more
```

In this command,

`sort -nr` Sorts the list of files by block size from smallest to largest.

Example—Finding Large Files

In the following example, `wtmpx` and `lastlog` are the largest files in the `/var/adm` directory.

```
$ cd /var/adm
$ ls -s | sort -nr | more
total 624
 320 wtmpx
 128 lastlog
  74 pacct
  56 messages
  30 wtmp
   6 utmpx
   2 utmp
   2 sulog
   2 sa
   2 passwd
   2 log
   0 spellhist
```

▼ 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
```

In this command,

directory Is the directory you want to search.

+*nnn* Is a number of 512-byte blocks. Files that exceed this size 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 the du(1M)and quot(1M) man pages.

▼ 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 ...]
```

In this command,

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 sizes of two directories and all the subdirectories they contain.

```
$ du /var/log /var/cron
4       /var/log
3250     /var/cron
```

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 -a /var/log /var/cron
0       /var/log/authlog
0       /var/log/syslog
2       /var/log/sysidconfig.log
4       /var/log
3248    /var/cron/log
3250    /var/cron
```

The following example displays the total sizes of two directories.

```
$ du -s /var/log /var/cron
4      /var/log
3250   /var/cron
```

▼ How to Display the User Allocation of Local UFS File System

3. **Become root.**

4. **Display users, directories, or file systems, and the number of 1024-byte blocks used.**

```
# quot [-a] [filesystem]
```

In this command,

`-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:
35400   bin
14312   smtp
  183   adm
   49   lp
   47   uucp
   37   bob
   28   sys
    2   mary
# quot -a
/dev/rdsk/c0t0d0s0 (/):
35400   bin
14312   smtp
  183   adm
   49   lp
   47   uucp
   37   bob
   28   sys
    2   mary
/dev/rdsk/c0t0d0s6 (/usr):
104276  smtp
56567   bin
 2000   lp
  698   uucp
    1   adm
/dev/rdsk/c0t0d0s7 (/export/home):
  617   smtp
```

65

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 the `ls(1)` and `find(1)` man pages.

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 “Crash Dump Files” and “Message Files” on page 1244.

▼ 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]
```

In this command,

directory Is the directory you want to search.

Verification—Listing the Newest Files

Verify that the first files displayed by `ls -t` are the files that have been created or changed most recently by using `ls -l` to list the date and time for all files in a directory.

Example—Listing the Newest Files

The following example shows how to use `ls -t` to locate the most recent files within the `/var/adm` directory. `sulog`, `messages`, `utmpx`, `wtmpx`, `utmp`, and `lastlog` were created or edited most recently. This is verified using output from `ls -l`, which shows that these three files were created or edited in March, while the other files in `/var/spool` were created or edited earlier.

```
$ ls -t /var/adm
sulog      wtmpx        wtmp        messages.1  vold.log   spellhist
messages   utmp         sa          messages.2  log        aculog
utmpx      lastlog      messages.0  messages.3  acct       passwd
$ ls -l /var/spool
total 686
drwxr-xr-x  5 adm       adm         512 Feb 13 16:20 acct
-rw-------  1 uucp      bin           0 Feb 13 16:04 aculog
-r--r--r--  1 root      other      8456 Mar 27 10:34 lastlog
drwxr-xr-x  2 adm       adm         512 Feb 13 16:36 log
-rw-r--r--  1 root      other    117376 Mar 27 13:11 messages
-rw-r--r--  1 root      other      4620 Jan 30 08:30 messages.0
-rw-r--r--  1 root      other     11176 Jan 23 04:30 messages.1
-rw-r--r--  1 root      other        60 Jan 13 09:45 messages.2
-rw-r--r--  1 root      other         0 Jan 31 04:05 messages.3
drwxr-xr-x  2 adm       adm         512 Feb 13 16:03 passwd
drwxr-xr-x  2 adm       sys         512 Mar 20 06:59 sa
-rw-rw-rw-  1 bin       bin           0 Feb 13 16:04 spellhist
-rw-------  1 root      root       1647 Mar 27 13:28 sulog
-rw-r--r--  1 root      bin         504 Mar 27 10:34 utmp
-rw-r--r--  1 root      bin        5208 Mar 27 10:34 utmpx
-rw-rw-rw-  1 root      root        500 Jan 11 14:40 vold.log
-rw-rw-r--  1 adm       adm       14724 Mar 27 10:34 wtmp
-rw-rw-r--  1 adm       adm      151404 Mar 27 10:34 wtmpx
```

▼ How to Find and Remove Old or Inactive Files

1. **Become root.**

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
```

In this command,

directory Is the 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 Is the file containing the list of inactive files.

3. **Remove the inactive files that you listed in the previous step.**

```
# rm ‘cat filename‘
```

In this command,

filename Is the file containing 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.

```
# 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/wtmp
/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 root.**

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 temporary or obsolete subdirectories and files (for example, `mail`, `lost+found`, or `quotas`), and delete them by repeating Step 3 above.**

Verification—Clearing Out Temporary Directories

To verify that you have removed all files from a temporary directory, use the `ls` command within that directory.

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 root.**

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 root.**

2. **Change to the directory where crash dump files are stored.**

```
# cd /var/crash/system
```

In this command,

system Is the 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 *
```

Verification—Deleting Crash Dump Files

To verify that you have removed crash dump files from their directory, use the `ls` command.

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
#
```

Managing Quotas 66

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	*page 1316*
How to Set Up Quotas for a User	*page 1317*
How to Use Prototype Quotas to Set Up Multiple Users	*page 1318*
How to Check Quota Consistency	*page 1319*
How to Turn Quotas On	*page 1320*
How to Change the Soft Time Limit Default	*page 1323*
How to Turn Quotas Off	*page 1328*
How to Change Quotas for a User	*page 1325*
How to Disable Quotas for a User	*page 1327*
How to Check for Exceeded Quotas	*page 1321*
How to Check Quotas on a File System	*page 1322*

66

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 contains procedures that describe how to configure file systems for quotas, and how to set up and activate quotas.

Table 66-1 describes the commands you use to set up disk quotas.

Table 66-1 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 reports on 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.

Table 66-2 Task Map: Setting Up Quotas

Activity	Description	For Instructions, Go To	
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	page 1316
Set Up Quotas for One User	Use `edquota` to create disk and inode quotas for a single user account.	▼ How to Set Up Quotas for a User	page 1317
Set Up Quotas for Multiple Users	Optional. Use `edquota` to apply prototype quotas to other user accounts.	▼ How to Use Prototype Quotas to Set Up Multiple Users	page 1318
Check for Consistency	Use `quotacheck` to compare quotas to current disk usage for consistency on one or more file systems.	▼ How to Check Quota Consistency	page 1319
Initiate Quotas	Use `quotaon` to initate quotas on one or more file systems.	▼ How to Turn Quotas On	page 1320

▼ How to Configure File Systems for Quotas

1. **Become root.**
2. **Edit the** `/etc/vfstab` **file by using the editor of your choice. Enter** `rq` **in the mount options field for each UFS file system that will have quotas.**
3. **Exit the file, saving the changes.**
4. **Change directory to the top of the file system that will have quotas.**
5. **Create a file named** `quotas`**.**

```
# touch quotas
```

6. **Change permissions to read/write for root only.**

```
# chmod 600 quotas
```

Example—Configuring File Systems for Quotas

The following example of a line from `/etc/vfstab` shows that the directory `/export/home` from the system `pluto` is mounted as an NFS file system on mount point `/usr/home`. `rq` was entered in the mount options field, therefore quotas are activated each time this file system is rebooted.

```
#device         device          mount           FS      fsck    mount     mount
#to mount       to fsck         point           type    pass    at boot   options
#
pluto:/export/home   -             /usr/home        nfs     -      yes       rq
```

▼ How to Set Up Quotas for a User

1. **Become root.**
2. **Use the quota editor to create a temporary file containing one line of quota information for each mounted file system that has a `quotas` file in its top-level directory.**

   ```
   # edquota username
   ```

   In this command,

username	Is the user for whom you wish 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. **Exit the editor, saving your changes.**

Verification—Setting Up Quotas for a User

To verify that you have set up a user's quota, use the `quota` command.

```
# quota [-v] [username]
```

In this command,

`-v`	Is the verbose option.
username	Is the user name whose quota you want to check.

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 Use Prototype Quotas to Set Up Multiple Users

1. **Become root.**
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 username1 [username2 username3 ...]
```

In this command,

prototype-user Is the user name of the account for which you have set up quotas.

username1, 2, 3 ... Specifies one or more user names of additional accounts.

Example—Using Prototype Quotas to Set Up 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 `checkquota` command. You could create a `cron` script to run `checkquota` during off hours. See `cron(1M)` for more information.

1. **Become root.**
2. **Run a consistency check and assign correct initial values to file systems.**

```
# quotacheck -v [-a | filesystem]
```

In this command,

`-v`	Is the verbose option.
`-a`	(Optional) Checks all file systems with an `rq` entry in the `/etc/vfstab` file.
filesystem	(Optional) Specifies a file system to check.

Example—Checking Quota Consistency

The following example checks quotas for the file system `/usr` on slice `/dev/rdsk/c0t0d0s6`. `/usr` is the only file system with an `rq` entry in `/etc/vfstab`.

```
# quotacheck -va
*** Checking quotas for /dev/rdsk/c0t0d0s6 (/usr)
```

▼ How to Turn Quotas On

1. **Become root.**
2. **Turn file system quotas on by using the** `quotaon` **command.**

```
# quotaon -v [-a | filesystem1 filesystem2 filesystem3 ...]
```

In this command,

`-v`	(Optional) Is the verbose option.
`-a`	(Optional) Turns quotas on for all file systems with an `rq` entry in the `/etc/vfstab` file.
filesystem1, 2, 3 ...	(Optional) Turns quotas on for one or more file systems that you specify.

Example—Turning Quotas On

The following example turns quotas on for `/files` on slice `/dev/dsk/c0t4d0s2` and `/snag` on slice `/dev/dsk/c0t3d0s2`.

```
# quotaon -v /dev/dsk/c0t4d0s2 /dev/dsk/c0t3d0s2
/dev/dsk/c0t4d0s2: quotas turned on
/dev/dsk/c0t3d0s2: 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.

Table 66-3 describes the commands you use to check quotas.

Table 66-3 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 root.**
2. **Display user quotas for mounted file systems where quotas are enabled.**

```
# quota -v [username]
```

In this command,

`-v`	Displays users' quotas on all mounted file systems that have quotas.
username	(Optional) Is the 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
/usr               0      1      2                  0      2      3
```

▼ 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 root.**
2. **Display all quotas for one or all file systems, even if there is no usage.**

```
# repquota [-v /dev/dsk/devicename | -a]
```

In this command,

`-v /dev/dsk/`*devicename*	(Optional) Reports on the specified file system.
`-a`	(Optional) Reports on all file systems.

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 (`/usr`).

```
# repquota -va
/dev/dsk/c0t0d0s6 (/usr):
                      Block limits                      File limits
User           used   soft   hard   timeleft     used   soft   hard    timeleft
#301     --       0      1      2                   0      2      3
#341     --       0      1      2                   0      2      3
```

Changing and Removing Quotas

You can change quotas to adjust the amount of disk space or number of inodes that users can consume. Or you can remove quotas for individual users or from entire file systems as needed.

Table 66-4 describes the commands you use to change or remove quotas.

Table 66-4 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 time period for each file system that its soft limits can be exceeded by any user.
`quotaoff`	Turn off quotas for specified file systems. See the `quotaon(1M)` man page for more information.

▼ 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 root.**
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`**.**
4. **Exit the file, saving your changes.**

Verification—Changing the Soft Time Limit Default

To verify that a user's soft time limit has been correctly changed, use the `quota` command.

```
# quota [-v] [username]
```

In this command,

`-v` Is the verbose option.

username Is the user name whose quota you want to check.

Examples—Changing the Soft Time Limit Default

The following example shows the contents of the temporary file opened by `edquota` on a system where `/usr` is the only mounted file system. The `0` (default) value means that the default time limit of one week is used.

```
fs /usr 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 one week, and the time limit for exceeding the number of files has been changed to ten days.

```
fs /usr blocks time limit = 1 week, files time limit = 10 days
```

▼ How to Change Quotas for a User

1. **Become root.**
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
   ```

   In this command,

   username Is the user name whose quota will be disabled.

Note – 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. **Exit the editor, saving your changes.**

Verification—Changing Quotas for a User

To verify that a user's quota has been correctly changed, use the `quota` command.

```
# quota [-v] [username]
```

In this command,

`-v` Is the verbose option.

username Is the 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 = 0) inodes (soft = 0, hard = 0)
```

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
```

How to Disable Quotas for a User

1. **Become root.**
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
```

In this command,

username Is the user name whose quota will be disabled.

Note – 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 (zero).**

Note – Be sure you change the values to zero. Do *not* delete the line from the text file.

---

4. **Exit the editor, saving your changes.**

Verification—Disabling Quotas for a User

To verify that you have disabled a user's quota, use the `quota` command.

```
# quota [-v] [username]
```

In this command,

`-v` Is the verbose option.

username Is the 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 = 10)
```

▼ How to Turn Quotas Off

1. **Become root.**

2. **Turn file system quotas off.**

```
# quotaoff [-v] [-a | filesystem1, filesystem2, filesystem3 ...]
```

In this command,

`-v` Is the verbose option.

`-a` (Optional) Turns quotas off for all file systems.

filesystem1, 2, 3 ... (Optional) Turns quotas off for one or more file systems you specify.

Example—Turning Quotas Off

The following example turns the quotas off for the /usr file system.

```
# quotaoff -v /usr
/usr: quotas turned off
```

Setting Up and Maintaining Accounting 67

This section contains some simple procedures for setting up and maintaining accounting. This is a list of the step-by-step instructions in this chapter.

How to Set Up Accounting	*page 1336*
How to Bill Users	*page 1338*
How to Fix a wtmp File	*page 1339*
How to Fix tacct Errors	*page 1340*
How to Restart runacct	*page 1341*

Setting Up Accounting

You can set up system accounting to run while the system is in multiuser mode (system state 2). Generally, this involves creating the `/etc/rc0.d/K22acct` and `/etc/rc2.d/S22acct` files, and modifying the `/var/spool/cron/crontabs/adm` and `/var/spool/cron/crontabs/root` files.

Most of the `cron` entries needed for accounting are put into the `/var/spool/cron/crontabs/adm` database file. The sample entries there run `ckpacct` periodically, `runacct` daily, and `monacct` 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.

Another feature offered by Solaris accounting facilities is the `chargefee` utility, which stores charges for special services provided to a user, such as file restoration, in the file `/var/adm/acct/fee`.

The `runacct` Program

The main daily accounting shell script, `runacct`, is normally invoked by `cron` outside of prime time 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 are removed, diagnostic files are saved, and execution is ended. For instructions on how to start runacct again, see "How to Restart runacct" on page 1341.

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 Table 67-1:

Table 67-1 runacct States

State	Description
SETUP	The command turnacct switch is executed to create a new pacct file. The process accounting files in /var/adm/pacctn (except for the pacct file) are moved to /var/adm/Spacctn.*MMDD*. The /var/adm/wtmp 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 wtmp 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 wtmp file if a record of a date change appears. It also deletes any corrupted entries from the wtmp 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 wtmp 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 MERGE program merges the process accounting records with the connect accounting records to form daytacct.

Table 67-1 `runacct` States *(Continued)*

State	Description
`FEES`	The `MERGE` program merges ASCII `tacct` records from the `fee` file into `daytacct`.
`DISK`	If the `dodisk` procedure has been run, producing the file `disktacct`, the `DISK` program merges the file into `daytacct` and move `disktacct` to `/tmp/disktacct.`*MMDD*.
`MERGETACCT`	The `MERGETACCT` 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 re-created if it is corrupted or lost.
`CMS`	The program `acctcms` 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 files `sum/daycms` and `sum/cms`, respectively. The program `lastlogin` is used to create the log file `/var/adm/acct/sum/loginlog`, 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.

Files Produced by runacct

The most useful files produced by runacct (found in /var/adm/acct) are shown in Table 67-2.

Table 67-2 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.

How to Set Up Accounting

1. **Become root.**

2. **If necessary, install the** `SUNWaccr` **and** `SUNWaccu` **packages on your system by using the** `pkgadd` **or** `swmtool` **command.**

3. **Install** `/etc/init.d/acct` **as the start script in Run Level 2**.

```
# ln /etc/init.d/acct /etc/rc2.d/S22acct
```

4. **Install** `/etc/init.d/acct` **as the stop script in Run Level 0**.

```
# ln /etc/init.d/acct /etc/rc0.d/K22acct
```

5. **Add entries for** `/usr/lib/acct/ckpacct`**,** `/usr/lib/acct/runacct`**, and** `/usr/lib/acct/monacct` **to the end of** `/var/spool/cron/crontabs/adm`**, by using the editor of your choice.**

6. **Add an entry for** `/usr/lib/acct/dodisk` **to the end of** `/var/spool/cron/crontabs/root` **by using the editor of your choice.**

7. **Edit** `/etc/acct/holidays` **to include national and local holidays, by using the editor of your choice.**

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       1.12   94/03/24 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.
#
0 2 * * 0,4 /etc/cron.d/logchecker
5 4 * * 6   /usr/lib/newsyslog
15 3 * * * /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 22 * * 4 /usr/lib/acct/dodisk
```

The following example shows a sample /etc/acct/holidays file.

```
* @(#)holidays  2.0 of 1/1/89
* Prime/Nonprime Table for UNIX Accounting System
*
* Curr  Prime   Non-Prime
* Year  Start   Start
*
  1989  0800    1800
*
* only the first column (month/day) is significiant.
*
* month/day     Company
*               Holiday
*
1/1             New Years Day
5/30            Memorial Day
7/4             Indep. Day
9/5             Labor Day
11/24           Thanksgiving
11/25           day after
12/25           Christmas
12/26           PDO after Christmas
```

How to Bill Users

1. **Become root.**

2. **Set up your system to charge for services provided to a user by using the** `chargefee` **command, which records charges in the file** `/var/adm/fee`**.**

```
# chargefee username amount
```

In this command,

username Is the user account you want to bill.

amount Is the number of units to bill the user.

Example—Billing Users

The following example charges 10 units each time a user logs in to the account `print_customer`.

```
# chargefee print_customer 10
```

Maintaining Accounting

▼ How to Fix a `wtmp` File

1. **Change to the** `/var/adm/acct/nite` **directory.**
2. **Convert the binary file** `wtmp.`*MMDD* **into the ASCII file** `xwtmp`**.**

```
$ fwtmp wtmp.MMDD xwtmp
```

In this command,

MMDD	Is a pair of two-digit numbers representing the month and day.

3. **Edit** `xwtmp`**. Delete the corrupted files, or delete all records from the beginning up to the date change.**
4. **Convert the ASCII file** `xwtmp` **to a binary file, overwriting the corrupted file.**

```
$ fwtmp -ic xwtmp wtmp.MMDD
```

How to Fix tacct Errors

1. **Change to the** `/var/adm/acct/sum` **directory.**
2. **Convert the contents of** `tacct.`*MMDD* **from binary to ASCII format.**

```
$ acctmerg -v tacct.MMDD xtacct
```

In this command,

MMDD Is the month and day specified by two-digit numbers.

3. **Edit the** `xtacct` **file, removing bad records and writing duplicate records to another file.**
4. **Convert the** `xtacct` **file from ASCII format to binary.**

```
$ acctmerg -i xtacct tacct.MMDD
```

In this command,

MMDD Is the month and day specified by two-digit numbers.

5. **Merge the files** `tacct.prv` **and** `tacct.`*MMDD* **into the file** `tacct`**.**

```
$ acctmerg tacctprv tacct.MMDD tacct
```

How to Restart runacct

The runacct procedure can fail for a variety of reasons, the most common being a system crash, /var running out of space, or a corrupted wtmp 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.

```
$ runacct [MMDD [state]]
```

In this command,

MMDD Is the month and day specified by two-digit numbers.

state Specifies a state, or starting point, where runacct processing should begin.

Scheduling System Events 68

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	*page 1346*
How to Display a crontab File	*page 1348*
How to Remove a crontab File	*page 1350*
How to Deny crontab Access	*page 1353*
How to Limit crontab Access to Specified Users	*page 1354*
How to Create an at Job	*page 1357*
How to Display the at Queue	*page 1358*
How to Display at Jobs	*page 1359*
How to Remove at Jobs	*page 1360*
How to Deny at Access	*page 1361*

Commands for Scheduling System Events

You can schedule system events to execute repetitively, at regular intervals, by using `crontab`. Or, you can schedule a single system event for execution at a specified time by using `at`. Table 68-1 summarizes `crontab` and `at`, as well as the files that enable you to control access to these commands.

Table 68-1 Command Summary: Scheduling System Events

Command	What It Schedules	Location of Files	Files That Control Access
`crontab`	Repetitive system events	`/usr/spool/cron/crontabs` or `/var/spool/cron/crontabs`	`/etc/cron.d/cron.allow` and `/etc/cron.d/cron.deny`
`at`	A single system event	`/usr/spool/cron/atjobs` or `/var/spool/cron/atjobs`	`/etc/cron.d/at.deny`

Scheduling a Repetitive System Event

The following sections describe how to create, edit, display, and remove `crontab` files, as well as how to control access to them.

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 Table 68-2, are separated by spaces. They indicate when the command will be executed.

Table 68-2 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.

```
16 0 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 the `crontab(1)`man page.

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.

Unless you have set up an editor for your account, the crontab facility defaults to 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: Undefined variable
$ setenv EDITOR /usr/bin/vi
```

When you create a crontab file, it is automatically placed in the /usr/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 root privileges.

Enter crontab command entries as described in “Syntax of crontab File Entries” on page 1344.

▼ How to Create or Edit a crontab File

1. **Be sure that you have access to the editor of your choice.**
2. **(Optional) To create or edit a crontab file belonging to root or another user, become root.**
3. **Create a new crontab file, or edit an existing one.**

```
$ crontab -e [username]
```

In this command,

username Is the name of another user’s account, and 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.

4. **Add command lines to the file, following the syntax described in “Syntax of crontab File Entries” on page 1344.**

5. **Exit the file, saving the changes.**
   The `crontab` file will be placed in `/usr/spool/cron/crontabs`.

Verification—Creating or Editing a `crontab` *File*

To verify that a `crontab` file exists for a user, use the `ls -l` command in the `/usr/spool/cron/crontabs` directory. For example, the following display shows that `crontab` files exist for users `smith` and `jones`.

```
$ ls -l /usr/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 1348.

Example—Creating or Editing a `crontab` *File*

The following example shows how to create a `crontab` file for another user.

```
$ su
Password:
# 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 a.m. every Sunday. 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 * * 6 rm /home/jones/*.log > /dev/null 2>&1
```

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 `/usr/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 root.

▼ How to Display a `crontab` File

1. **(Optional) To display a `crontab` file belonging to root or another user, become root.**
2. **Display the `crontab` file.**

```
$ crontab -l [username]
```

In this command, *username* is another user's account, and requires root privileges to display.

username Is the name of another user's account, and 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.

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 /usr/documents/*.book > /dev/null 2>&1
$ su
Password:
# crontab -l
#ident “@(#)root   1.12   94/03/24 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.
#
0 2 * * 0,4 /etc/cron.d/logchecker
5 4 * * 6   /usr/lib/newsyslog
15 3 * * * /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
0 1 * * * /usr/sbin/cfsadmin -s all
# 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 root to remove `crontab` files belonging to root or other users.

You do not have to change directories to `/usr/spool/cron/crontabs` (where `crontab` files are located) to use this command.

▼ How to Remove a `crontab` File

1. **(Optional) To remove a** `crontab` **file belonging to root or another user, become root.**

2. **Remove the** `crontab` **file.**

```
$ crontab -r [username]
```

In this command,

username   Is the name of another user's account, and requires root privilegs 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.

Verification—Removing a `crontab` *File*

You can verify that you have removed a `crontab` file by using the `ls` command within the `/usr/spool/cron/crontabs` directory to display the existing `crontab` files.

To verify that you have removed a `crontab` file, use the `ls` command to check the `crontab` directory.

```
# ls /usr/spool/cron/crontabs
adm    jones    lp     root    sys
# crontab -r jones
# ls /usr/spool/cron/crontabs
adm     lp     root    sys
```

Example—Removing a `crontab` *File*

The following example shows how to use `crontab -r` to remove the default user’s crontab file, as well as `crontab` files belonging to root and another user. `ls` verifies that the correct `crontab` files have been removed.

```
$ ls /usr/spool/cron/crontabs
adm     jones     lp     root    smith    sys
$ crontab -r
$ ls /usr/spool/cron/crontabs
adm     jones     lp     root    sys
$ su
Password:
# crontab -r
# ls /usr/spool/cron/crontabs
adm     jones     lp    sys
# crontab -r jones
# ls /usr/spool/cron/crontabs
adm     lp    sys
```

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, only root can run crontab.

During Solaris software installation, a default cron.deny file is provided:

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

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.

▼ How to Deny `crontab` Access

1. **Become root.**

2. **Using the editor of your choice, edit the `/etc/cron.d/cron.deny` file to 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. **Exit the file, saving the changes.**

▼ How to Limit crontab Access to Specified Users

1. **Become root.**
2. **Use the editor of your choice to create a file named** /etc/cron.d/cron.allow**.**
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, root access to crontab commands will be denied.

4. **Exit the file, saving the changes.**

Verification—Limiting crontab *Access to Specified Users*

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 is either 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`.

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
```

Scheduling a Single System Event

The following sections describe how to use `at` to schedule jobs, such as commands and scripts, for execution at a later time, as well as how to display and remove these jobs, and how to control access to `at`.

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 root 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. 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 belonging to Smith near midnight on the last day of March.

```
$ at 11:45pm mar 31
at> rm /home/smith/*core*
at> Press Control-d
job 793924770.a at Fri Mar 31 23:45:00 1995
```

▼ How to Create an at Job

1. **Enter the** at **facility, specifying the time you want your job executed, and press Return.**

```
$ at [-m] time [date]
```

In this command,

-m — Sends you mail after the job is completed.

time — Is the 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 — Is the 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.

Verification—Creating an at *Job*

To verify that you have created an at job, use the atq command (described in "How to Display at Jobs" on page 1359). atq confirms that at jobs belonging to jones have been submitted to the queue.

```
$ atq
Rank     Execution Date     Owner     Job         Queue   Job Name
 1st   Feb 28, 1995 14:30   jones    793920600.a     a     stdin
 2nd   Feb 29, 1995 08:10   jones    793962720.a     a     stdin
```

Examples—Creating an `at` *Job*

The following example shows the `at` job that user `jones` created to remove her backup files at 7:30 p.m. 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 543962000.a at Tue Feb 28 19:30:00 1995
```

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.

```
$ 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
```

Example—Displaying the `at` *Queue*

The following example shows output from the `atq` command on February 27, 1995. Execution times and dates for three jobs submitted by `jones` are displayed.

```
$ atq
Rank     Execution Date     Owner     Job         Queue   Job Name
 1st   Feb 28, 1995 14:30   jones    793920600.a     a     stdin
 2nd   Mar  1, 1995 08:10   jones    793962720.a     a     stdin
 3rd   Mar  1, 1995 12:00   jones    793991450.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]
```

In this command,

job-id Is the 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
793920600.a     Tue Feb 28 14:30:00 1995
793962720.a     Wed Mar 01 08:10:00 1995
793991450.a     Wed Mar 01 12:00:00 1995
```

The following example shows output displayed when a single job is specified with the `at -l` command.

```
$ at -l 793962720.a
793962720.a     Wed Mar 01 08:10:00 1995
```

▼ How to Remove `at` Jobs

1. **(Optional) To remove an `at` job belonging to root or another user, become root.**

2. **Remove the `at` job from the queue before it is executed.**

   ```
   $ at -r [job-id]
   ```

   In this command,

   job-id Is the identification number of the job you want to remove.

Verification—Removing `at` Jobs

To verify that you have removed an `at` job, use 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.

Example—Removing `at` *Jobs*

In the following example, a user wants to remove an `at` job that was scheduled to execute at noon on March 1. 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 displays the `at` queue again to confirm that this job has been removed.

```
$ at -l
793920600.a     Tue Feb 28 14:30:00 1995
793962720.a     Wed Mar 01 08:10:00 1995
793991450.a     Wed Mar 01 12:00:00 1995
$ at -r 793991450.a
$ at -l 793991450.a
at: 793991450.a does not exist
```

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 root.**

2. **Using the editor of your choice, open the** `/etc/cron.d/at.deny` **file.**

3. **Add or remove 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
.
.
.
```

4. **Exit the file, saving your changes.**

Verification—Denying `at` *Access*

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.

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 at.

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

The following example shows that if a user is not listed in at.deny, he can access at, display the at prompt, and receive output confirming that his at job has been accepted and assigned a number in the at queue. The command atq confirms this.

```
$ at 4:55pm
at> echo Testing > /dev/console
at> Press Control-d
warning: commands will be executed using /bin/csh
job 843720330.a at Thu Feb 23 16:55:00 1995
$ atq
Rank     Execution Date     Owner      Job         Queue   Job Name
 1st   Feb 23, 1995 16:55   smith    843720330.a     a     stdin
```

This means that smith is not listed in the at.deny file.

Part 15 —Managing System Performance

This part provides instructions for managing system performance.

69 **Overview of System Performance**
Provides overview information about performance topics.

70 **Managing Processes**
Provides step-by-step instructions for using process commands to enhance system performance.

71 **Monitoring Performance**
Provides step-by-step instructions for using `vmstat`, `sar`, and disk utilization commands to monitor performance.

72 **Monitoring Network Performance**
Provides step-by-step instructions for monitoring network performance.

73 **Tuning Kernel Parameters**
Provides step-by-step instructions for tuning selected kernel parameters.

74

The Scheduler

Provides overview information about the SunOS 5.x scheduler.

Overview of System Performance 69

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.

System Performance and System Resources	*page 1368*
Processes and System Performance	*page 1369*
Disk I/O and System Performance	*page 1375*
Memory and System Performance	*page 1375*
Kernel Parameters and System Performance	*page 1377*
About Monitoring Performance	*page 1378*

69

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 regular basis 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 include:

- *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 8, "Monitoring Performance," describes the tools that display statistics about the activity and the performance of the computer system.

Other Sources of Information

Performance is a broad subject that can't be adequately covered in these chapters. There are several books available that cover various aspects of improving performance and tuning your system or network. Three useful books are:

- *Sun Performance and Tuning: SPARC and Solaris*, by Adrian Cockcroft, SunSoft 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 Table 69-1.

Table 69-1 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 5.x environment. Various process structures are described in Table 69-2.

Table 69-2 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.

Figure 69-1 illustrates the relationship of these structures.

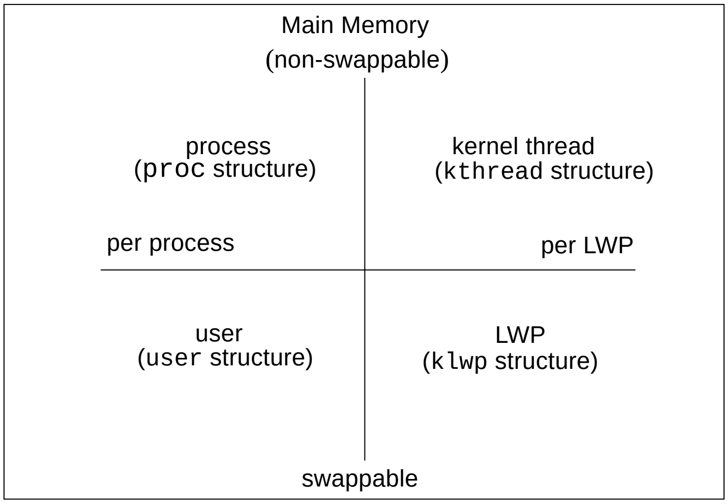


Figure 69-1 Process Structures

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.

Process Commands

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, and how to kill processes that have hung or become inactive. See Chapter 70, “Managing Processes,” for more information about using the `ps` command and its options.

In addition, process tools are available in `/usr/proc/bin` that display highly detailed information about the processes listed in `/proc`, 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.

The new /usr/proc/bin utilities are summarized in Table 69-3.

Table 69-3 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 70, “Managing Processes,” 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 the `proc(1)` man pages.

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 70, “Managing Processes,” for more information about stopping processes using the `kill` command.

Process Priority Levels

A process is allocated CPU time according to its scheduling class and its priority level. By default, the Solaris operating system has four process scheduling classes: *real-time*, *system*, *timesharing* and *interactive*.

- Real-time processes have the highest priority. This class includes processes that must respond to external events as they happen. For example, a process that collects data from a sensing device may need to process the data and respond immediately. In most cases, a real-time process requires a dedicated system. No other processes can be serviced while a real-time process has control of the CPU. By default, the range of priorities is 100-159.
- System processes have the middle priorities. This class is made up of those processes that are automatically run by the kernel, such as the swapper and the paging daemon. By default, the range of priorities is 60-99.
- Timesharing processes have the lowest priority. This class includes the standard UNIX processes. Normally, all user processes are timesharing processes. They are subject to a scheduling policy that attempts to distribute processing time fairly, giving interactive applications quick response time and maintaining good throughput for computations. By default, the range of priorities is 0-59.
- Interactive processes are introduced in the SunOS 5.4 environment. The priorities range from 0-59. All processes started under OpenWindows are placed in the interactive class and those processes with keyboard focus get higher priorities.

The scheduling priority determines the order in which processes will be run.

Real-time processes have fixed priorities. If a real-time process is ready to run, no system process or timesharing process can run.

System processes have fixed priorities that are established by the kernel when they are started. The processes in the system class are controlled by the kernel, and cannot be changed.

Timesharing and interactive processes are controlled by the scheduler, which dynamically assigns their priorities. You can manipulate the priorities of the processes within this class.

Changing the Scheduling Priority of Processes With `priocntl`

The scheduling priority of a process is the priority it is assigned by the process scheduler. These priorities are assigned according to the scheduling policies of the scheduler. The `dispadmin` command lists the default scheduling policies. See "Scheduler Configuration" on page 1461," for information on the `dispadmin` command.

The `priocntl` command can be used to assign processes to a priority class and to manage process priorities. See the section called "How to Designate Priority" on page 1401 for instructions on using the `priocntl` command to manage processes.

Changing the Priority of a Timesharing Process With `nice`

The `nice` 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 system administrator (or root) 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.

See “How to Change the Priority of a Process” on page 1404 for information about using the nice command.

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 root 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.

Disk I/O and System Performance

The disk is used to store data and instructions used by your computer system. You can examine how efficiently the system is accessing data on the disk by looking at the disk access activity and terminal activity. See “Monitoring Performance” on page 1405 for a discussion of the `iostat` and `sar` commands, which report statistics on disk activity. Managing and allocating disk space and dividing your disk into slices are discussed in "Managing Disks" in *System Administration Guide, Volume I.*

If the CPU spends much of its time waiting for I/O completions, there is a problem with disk slowdown. Some ways to prevent disk slowdowns are:

- Keep disk space with 10% free so file systems are not full. If a disk becomes full, back up and restore the file systems to prevent disk fragmentation. Consider purchasing products that resolve disk fragmentation.
- Organize the file system to minimize disk activity. If you have two disks, distribute the file system for a more balanced load. Using Sun’s Solstice DiskSuite™ product provides more efficient disk usage.
- Add more memory. Additional memory reduces swapping and paging traffic, and allows an expanded buffer pool (reducing the number of user-level reads and writes that need to go out to disk).
- Add a disk and balance the most active file systems across the disks.

Memory and System Performance

Performance suffers when the programs running on the system require more physical memory than is available. When this happens, the operating system begins paging and swapping, which is costly in both disk and CPU overhead.

Paging involves moving pages that have not been recently referenced to a free list of available memory pages. Most of the kernel resides in main memory and is not pageable.

Swapping occurs if the page daemon cannot keep up with the demand for memory. The swapper will attempt to swap out sleeping or stopped lightweight processes (LWPs). If there are no sleeping or stopped LWPs, the swapper will swap out a runnable process. The swapper will swap LWPs back in based on their priority. It will attempt to swap in processes that are runnable.

Swap Space

Swap areas are really file systems used for swapping. Swap areas should be sized based on the requirements of your applications. Check with your vendor to identify application requirements.

Table 69-4 describes the formula used to size default swap areas by the Solaris 2.x installation program. These default swap sizes are a good place to start if you are not sure how to size your swap areas.

Table 69-4 Default Swap Sizes

If Your Physical Memory Size Is ...	Your Default Swap Size Is ...
16–64 Mbytes	32 Mbytes
64-128 Mbytes	64 Mbytes
128-512 Mbytes	128 Mbytes
greater than 512 Mbytes	256 Mbytes

See the "Managing File Systems" section of *System Administration Guide, Volume I* for information about managing swap space.

Buffer Resources

The buffer cache for `read` and `write` system calls uses a range of virtual addresses in the kernel address space. A page of data is mapped into the kernel address space and the amount of data requested by the process is then physically copied to the process' address space. The page is then unmapped in the kernel. The physical page will remain in memory until the page is freed up by the page daemon.

This means a few I/O-intensive processes can monopolize or force other processes out of main memory. To prevent monopolization of main memory, balance the running of I/O-intensive processes serially in a script or with the `at(1)` command. Programmers can use `mmap(2)` and `madvise(3)` to ensure that their programs free memory when they are not using it.

Kernel Parameters and System Performance

Many basic parameters (or tables) within the kernel are calculated from the value of the maxusers parameter. Tables are allocated space dynamically. However, you can set maximums for these tables to ensure that applications won’t take up large amounts of memory.

By default, maxusers is approximately set to the number of Mbytes of physical memory on the system. However, the system will never set maxusers higher than 1024. The maximum value of maxusers is 2048, which can be set by modifying the /etc/system file.

See Chapter 73, “Tuning Kernel Parameters,” and the system(3S) man page for details on kernel parameters.

In addition to maxusers, a number of kernel parameters are allocated dynamically based on the amount of physical memory on the system, as shown in Table 69-5 below.

Table 69-5 Kernel Parameters

Kernel Parameter	**Description**
ufs_ninode	The maximum size of the inode table
ncsize	The size of the directory name lookup cache
max_nprocs	The maximum size of the process
ndquot	The number of disk quota structures
maxuprc	The maximum number of user processes per user-id

Table 69-6 lists the default settings for kernel parameters affected by the value assigned to maxusers.

Table 69-6 Default Settings for Kernel Parameters

Kernel Table	**Variable**	**Default Setting**
Inode	ufs_ninode	*max_nprocs* + 16 + *maxusers* + 64
Name cache	ncsize	*max_nprocs* + 16 + *maxusers* + 64
Process	max_nprocs	10 + 16 * *maxusers*
Quota table	ndquot	(*maxusers* * *NMOUNT*) / 4 + *max_nprocs*
User process	maxuprc	*max_nprocs* – 5

See Chapter 73, “Tuning Kernel Parameters,” for a description of the kernel parameters and how to change the default values.

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)

The following sections describe tools and commands that help you monitor performance.

The `sar` Command

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. “Automatic Collection of System Activity Data” on page 1384 describes these tools.

The vmstat *Command*

The vmstat command reports virtual memory statistics and shows CPU load, paging, number of context switches, device interrupts, and system calls.

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
```

The fields in the vmstat report have the following meanings:

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

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 used)

If sr does not equal zero, the page daemon has been running.

disk reports the number of disk operations per second. This field can show 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
`sy`	System time
`id`	Idle time

The `vmstat` command can also display statistics on swapping, cache flushing, and interrupts.

System Events

Run `vmstat -s` to show the total of various system events that have taken place since the system was last booted.

Swapping

Run `vmstat -S` to show swapping statistics in addition to paging statistics. The additional fields are:

`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.

Cache Flushing

Run `vmstat -c` to show cache flushing statistics for a virtual cache. It shows the total number of cache flushes since the last boot. The cache types are:

`usr`	User
`ctx`	Context
`rgn`	Region
`seg`	Segment
`pag`	Page
`par`	Partial-page

Interrupts

Run `vmstat -i` to show interrupts per device.

```
$ vmstat -i
interrupt         total     rate
---------------------------------
clock         104638405      100
esp0            2895003        2
fdc0                  0        0
---------------------------------
Total         107533408      102
```

The `iostat` *Command*

The `iostat` command reports statistics about disk input and output, and produces measures of throughput, utilization, queue lengths, transaction rates, and service time.

The following example shows disk statistics gathered every five seconds.

```
$ iostat 5
      tty          fd0           sd3        cpu
 tin tout bps tps serv  bps tps serv  us sy wt  id
   0    1   0   0    0    1   0 5640   0  1  0  98
   0   10   0   0    0    0   0    0   0  1  0  99
   0   10   0   0    0    0   0    0   0  1  0  99
   0   10   0   0    0   27   3  319   0  4  9  88
   0   10   0   0    0    2   0 5061   0  0  0  99
   0   10   0   0    0    0   0    0   0  0  0 100
   0   10   0   0    0    0   0    0   0  0  0 100
   0   10   0   0    0    0   0    0   0  0  0 100
   0   10   0   0    0    0   0    0   0  0  0 100
```

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`).

For each terminal, `iostat` displays:

- `tin` Number of characters in the terminal input queue
- `tout` Number of characters in the terminal output queue

For each disk, `iostat` displays the following information:

- `bps` Blocks per second
- `tps` Transactions per second
- `serv` Average service time, in milliseconds

For the CPU, `iostat` displays the CPU time spent in the following modes:

- `us` In user mode
- `sy` In system mode
- `wt` Waiting for I/O
- `id` Idle

Run `iostat –xtc` to get extended disk statistics.

```
$ iostat -xtc

disk   r/s w/s   Kr/s Kw/s   wait actv   svc_t    %w %b   tin tout us sy  wt  id
sd0    0.2 1.7    1.0  9.7   0.0  0.1    39.8     0  3     0   9   1  6  9   85
sd1     0.5 2.5    10.6 21.0   0.0  0.1     26.6     0  5
sd2     0.0 0.2     0.1  0.0   0.0  0.0    157.7     0  0
```

Each disk has a line of output:

- `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)
- `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

The df Command

The df command shows 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.

Use the df -k command to display file system information in Kbytes. The following information is given:

• 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

```
$  df -k
filesystem             kbytes     used     avail     capacity       mounted on
/dev/dsk/c0t3d0s0       17269    11099      4450          71%       /
/dev/dsk/c0t3d0s6      136045    79818     42627          65%       /usr
/proc                       0        0         0            0       /proc
swap                    40424        0     40416            0       /tmp
```

The profil Command

profil uses CPU statistics to show the amount of time that a program uses. You can analyze a program and identify the functions that consume a high percentage of CPU time. See the man page for profil(2) for more information.

Automatic Collection of System Activity Data

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`

There is another shell script, `sa2`, which 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 With `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.

Table 69-7 lists the sar options and their actions.

Table 69-7 sar Options

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
–v	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.

Monitoring Tools

The Solaris 2.x system software provides several tools to help you keep track of how your system is performing. These include:

- The `sar` and `sadc` utilities, which collect and report on many aspects of system activity. Chapter 71, "Monitoring Performance," describes these utilities and the information that they provide.
- The `ps` command, which provides information about the active processes. Chapter 70, "Managing Processes," describes the `ps` command.
- The performance meter, which provides a graphical representation of the status of your system and other hosts on the network. Chapter 71, "Monitoring Performance," describes the performance meter.
- The `vmstat` and `iostat` commands, which summarize system activity, providing information about virtual memory activity, disk usage, and CPU activity. Chapter 71, "Monitoring Performance," describes these tools.
- The `swap` command, which can be used to display information about available swap space on your system. See the "Managing File Systems" section in *System Administration Guide, Volume I* for information on using the `swap` command.
- The `netstat` and `nfsstat` commands, which display information about network performance. Chapter 72, "Monitoring Network Performance," describes these commands.

Managing Processes 70

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	*page 1391*
How to Display Information About Processes	*page 1394*
How to Control Processes	*page 1397*
How to Kill a Process	*page 1398*
How to Display Basic Information About Process Classes	*page 1399*
How to Display the Global Priority of a Process	*page 1400*
How to Designate Priority	*page 1401*
How to Change Scheduling Parameters of a Timeshare Process	*page 1402*
How to Change the Class of a Process	*page 1403*
How to Change the Priority of a Process	*page 1404*

70

Displaying Information About Processes (ps)

You can check the status of active processes on a system by using the ps command. 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

Table 70-1 describes some of the fields reported by the ps command. The fields displayed depend on which option you choose. See the ps(1) man page for a description of all available options.

Table 70-1 Summary of Fields in ps Reports

Field	Description
UID	The user ID of the process's owner.
PID	The process identification number.
PPID	The parent process's identification number.
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.
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).

Table 70-1 Summary of Fields in ps Reports *(Continued)*

Field	Description
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]
```

In this command,

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/automountd
    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
    root   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   199   191  0   May 05 ?        0:00 lpNet
    root   162     1  0   May 05 ?        0:07 /usr/sbin/syslogd
    root   181     1  0   May 05 ?        0:03 /usr/sbin/nscd -e passwd,no -e
group,no -e hosts,no -f /etc/nscd.conf
    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 -g -h -p
saturn console login:  -T AT386 -d /dev/console -l
    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
```

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`. Table 70-2 lists these process tools. For more detailed information, refer to the `proc(1)` man page.

Table 70-2 `/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** `ps` **command to obtain the identification number of the process you want to display more information about.**

```
$ ps -e | grep process
```

In this command,

process Is the 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
```

In this command,

pcommand Is the process tool command you want to run. Table 70-2 lists these commands.

PID Is the 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 `lpNet` process. First the `/usr/proc/bin path` is defined to avoid typing long process tool commands. Next, the identification number for `lpNet` is obtained. Finally, output from three process tool commands is shown.

```
❶  $ PATH=$PATH:/usr/proc/bin
   $ export PATH
❷  $ ps -e | grep lpNet
      191 ?        0:00 lpNet
❸  $ pwdx 191
   191:    /var/spool/lp
❹  #$ ptree 191
   183   /usr/lib/lpsched
     191   lpNet
❺  $ pfiles 191
   191:    lpNet
     Current rlimit: 1024 file descriptors
      0: S_IFCHR mode:0666 dev:102,0 ino:23278 uid:0 gid:3 rdev:13,2
         O_RDWR
      1: S_IFCHR mode:0666 dev:102,0 ino:23278 uid:0 gid:3 rdev:13,2
         O_RDWR
      2: S_IFCHR mode:0666 dev:102,0 ino:23278 uid:0 gid:3 rdev:13,2
         O_RDWR
      3: S_IFIFO mode:0000 dev:159,0 ino:65 uid:0 gid:0 size:0
         O_RDWR
      4: S_IFREG mode:0666 dev:102,0 ino:14900 uid:0 gid:0 size:105
         O_RDWR|O_APPEND
      5: S_IFREG mode:0664 dev:102,0 ino:17007 uid:71 gid:8 size:2141
         O_RDONLY
      6: S_IFIFO mode:0000 dev:159,0 ino:66 uid:0 gid:0 size:0
         O_RDWR
      7: S_IFIFO mode:0000 dev:159,0 ino:66 uid:0 gid:0 size:0
         O_RDWR
      8: S_IFIFO mode:0000 dev:159,0 ino:67 uid:0 gid:0 size:0
         O_RDWR
      9: S_IFIFO mode:0000 dev:159,0 ino:67 uid:0 gid:0 size:0
         O_RDWR
     10: ??? mode:0444 dev:165,0 ino:2 uid:0 gid:0 size:0
         O_RDONLY close-on-exec
```

❶ Adds the /usr/proc/bin directory to the PATH variable.

❷ Obtains the process identification number for lpNET.

❸ Displays the current working directory for lpNET.

❹ Displays the process tree containing lpNET.

❺ 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. Table 70-3 lists these process tools. For more detailed information, refer to the proc(1) man page.

Table 70-3 /usr/proc/bin Process Tools That Provide Control

Process Tool	What it Does
pstop	Stops a process
prun	Restarts a process

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.**

```
$ ps -e | grep process
```

In this command,

process Is the 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
```

In this command,

pcommand Is the process tool command you want to run. Table 70-3 lists these commands.

PID Is the identification number of a process.

Example—Controlling Processes

The following example shows how to use process tools to stop and restart Print Tool.

```
❶  $ PATH=$PATH:/usr/proc/bin
   $ export PATH
❷  $ ps -e | grep print*
   264 console 0:03 printtoo
❸  $ pstop 264
❹  $ prun 264
```

❶ Adds the /usr/proc/bin directory to the PATH variable.

❷ Obtains the process identification number for Print Tool.

❸ Stops the Print Tool process.

❹ Restarts the Print Tool process.

Killing a Process (`kill`*)*

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 root can kill any processes in the system except for those with process IDs `0`, `1`, `2`, `3`, and `4`.

For detailed information, refer to the `kill(1)` man page.

▼ How to Kill a Process

1. **(Optional) To kill a process belonging to another user, become root.**
2. **(Optional) Use output from the `ps` command to obtain the identification number of the process you want to display more information about.**

   ```
   $ ps -e | grep process
   ```

   In this command,

   process Is the 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 `kill` command to stop the process.**

   ```
   $ kill [-9] PID . . .
   ```

In this command,

`-9`	Ensures that the process terminates promptly.
PID . . .	Is the ID of the process or processes to stop.

Verification—Killing a Process

Use the `ps` command to be sure that the process has been stopped.

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 to and ensures a 0-59 range to create a *global* priority.

▼ 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: -20 through 20
```

▼ 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 Priority

1. **Become root.**

2. **Start a process with a designated priority.**

```
# priocntl -e -c class  -m  userlimit  -p pri command_name
```

In this command,

`-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.

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
```

▼ How to Change Scheduling Parameters of a Timeshare Process

1. **Become root.**

2. **Change the scheduling parameter of a running timeshare process.**

```
# priocntl -s -m userlimit [-p userpriority] -i idtype idlist
```

In this command,

`-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`.

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
```

▼ How to Change the Class of a Process

1. **(Optional) Become root.**

Note – You must be root 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
```

In this command,

`-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`.

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
```

Note – If, as root, you change a user process to the real-time class, the user cannot subsequently change the real-time scheduling parameters (using `priocntl -s`).

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 command increases the `nice` number by four units (the default); and the second 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.

Monitoring Performance *71*

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	*page 1406*
How to Display System Event Information	*page 1409*
How to Display Swapping Statistics	*page 1410*
How to Display Cache Flushing Statistics	*page 1410*
How to Display Interrupts Per Device	*page 1411*
How to Display Disk Utilization Information	*page 1412*
How to Display Extended Disk Statistics	*page 1414*
How to Display File System Information	*page 1415*
How to Check File Access	*page 1416*
How to Check Buffer Activity	*page 1417*
How to Check System Call Statistics	*page 1419*
How to Check Disk Activity	*page 1421*
How to Check Page-Out and Memory	*page 1423*
How to Check Kernel Memory Allocation	*page 1424*
How to Check Interprocess Communication	*page 1426*
How to Check Page-In Activity	*page 1427*
How to Check Queue Activity	*page 1429*
How to Check Unused Memory	*page 1431*

71

How to Check CPU Utilization	*page 1432*
How to Check System Table Status	*page 1433*
How to Check Swap Activity	*page 1435*
How to Check Terminal Activity	*page 1436*
How to Check Overall System Performance	*page 1437*
How to Set Up Automatic Data Collection	*page 1438*

Displaying Virtual Memory Statistics

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.

For a detailed description of this command, refer to the `vmstat(1M)` man page.

▼ How to Display Virtual Memory Statistics

Collect virtual memory statistics using the `vmstat` command with a time interval.

```
$ vmstat -n
```

In this command,

-n Is the interval in seconds between reports.

Table 71-1 describes the fields in the `vmstat` output.

Table 71-1 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

Table 71-1 Output From the `vmstat` Command *(Continued)*

Category	Field Name	Description
	`w`	Swapped out LWPs waiting for processing resources to finish
`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
	`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

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
  1329913 total address trans. faults taken
    25270 page ins
     3787 page outs
    38082 pages paged in
    13417 pages paged out
     3034 total reclaims
     3033 reclaims from free list
   335879 micro (hat) faults
   994034 minor (as) faults
    24210 major faults
   300634 copy-on-write faults
   141744 zero fill page faults
    34341 pages examined by the clock daemon
        5 revolutions of the clock hand
    28134 pages freed by the clock daemon
    11174 forks
     1259 vforks
     9086 execs
 11479519 cpu context switches
 95234544 device interrupts
  1426943 traps
  9100502 system calls
  1939346 total name lookups (cache hits 88%)
      496 toolong
   185566 user cpu
   977189 system cpu
 92045953 idle cpu
   130914 wait cpu
```

How to Display Swapping Statistics

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 s1 s3 -- in sy cs     us sy id
 0 0 0 6224  5536 0  0  0 0  0  0  0  0  0  0  0  2  9  12     0  1  99
```

The additional fields are described in Table 71-2.

Table 71-2 Output From the `vmstat -S` Command

Field	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

Run `vmstat -c` to show cache flushing statistics for a virtual cache.

```
$ vmstat -c
flush statistics: (totals)
    usr      ctx     rgn    seg     pag     par
  14512    20201       0   1811 1857286  815505
```

It shows the total number of cache flushes since the last boot. The cache types are described in Table 71-3.

Table 71-3 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

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         104638405      100
esp0            2895003        2
fdc0                  0        0
--------------------------------
Total         107533408      102
```

Displaying Disk Utilization Information

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 the `iostat(1M)` man page.

▼ How to Display Disk Utilization Information

You can display disk activity information by using the `iostat` command with a time interval.

```
$ iostat 5
      tty     fd0     sd1     sd3     cpu
tin tout Kps tps serv Kps tps serv Kps tps serv us sy wt id
  0    0   0   0    0   1   0   79   0   0   58  0  1  0 99
```

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`).

Table 71-4 describes the fields in the `iostat` command output.

Table 71-4 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

Table 71-4 Output From the `iostat -n` Command *(Continued)*

For Each ...	Field Name	Description
	`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          fd0           sd3       cpu
 tin tout bps tps serv  bps tps serv  us sy wt  id
   0    1   0   0    0    1   0 5640   0  1  0  98
   0   10   0   0    0    0   0    0   0  1  0  99
   0   10   0   0    0    0   0    0   0  1  0  99
   0   10   0   0    0   27   3  319   0  4  9  88
   0   10   0   0    0    2   0 5061   0  0  0  99
   0   10   0   0    0    0   0    0   0  0  0 100
   0   10   0   0    0    0   0    0   0  0  0 100
   0   10   0   0    0    0   0    0   0  0  0 100
   0   10   0   0    0    0   0    0   0  0  0 100
```

How to Display Extended Disk Statistics

Run `iostat –xtc` to get extended disk statistics.

```
$ iostat -xtc
                              extended disk statistics    tty       cpu
disk    r/s w/s   Kr/s Kw/s   wait actv  svc_t    %w %b   tin tout us sy  wt  id
sd0     0.2 1.7    1.0  9.7    0.0  0.1   39.8     0  3     0    9  1  6   9  85
sd1     0.5 2.5   10.6 21.0    0.0  0.1   26.6     0  5
sd2     0.0 0.2    0.1  0.0    0.0  0.0  157.7     0  0
```

This command displays a line of output for each disk. The output fields are described in Table 71-5.

Table 71-5 Output From the `iostat -xtc` Command

Field	**Description**
`r/s`	Reads per second
`w/r`	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)
`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

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

Use the `df -k` command to display file system information in Kbytes.

```
$ df -k
filesystem          kbytes    used     avail    capacity     mounted on
/dev/dsk/c0t3d0s0   17269     11099    4450     71%          /
```

Table 71-6 describes the `df -k` command output.

Table 71-6 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 from the `df -k` command.

```
$  df -k
filesystem            kbytes    used     avail    capacity     mounted on
/dev/dsk/c0t3d0s0      17269    11099    4450     71%          /
/dev/dsk/c0t3d0s6     136045    79818    42627    65%          /usr
/proc                      0    0        0        0            /proc
swap                   40424    0        40416    0            /tmp
```

Monitoring System Activities

For a detailed description of this command, refer to the `sar(1)` man page.

▼ How to Check File Access

Display file access operation statistics with the `sar -a` command.

```
$ sar -a
SunOS venus 5.4 prefcs3 sun4c 11/11/94


14:28:12    iget/s namei/s dirbk/s
14:29:12         0       2       1
14:30:12         0       4       1
14:31:12         0       3       1

Average          0       3       1
```

The operating system routines reported are described in Table 71-7.

Table 71-7 Output From the `sar -a` Command

Field	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

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

SunOS saturn 5.4 prefcs3 sun4c 01/24/95

00:00:03 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:02       0       0     100       0       0      57       0       0
```

The buffer activities displayed by the -b option are described in Table 71-8. 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 71-8 Output From the `sar -b` Command

Field	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 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 5.4 prefcs3 sun4c 11/11/94

14:28:12 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s
pwrit/s
14:29:12      0     14    100      6     17     67      0      0
14:30:12      0     12     99      6     16     65      0      0
14:31:12      0     12    100      6     16     65      0      0

Average       0     12    100      6     16     66      0      0
```

▼ How to Check System Call Statistics

Display system call statistics by using the `sar -c` command.

```
$ sar -c

SunOS saturn 5.4 prefcs3 sun4c 01/24/95

00:00:03 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:02       9       0       0   0.01   0.01      33       9
```

Table 71-9 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 71-9 Output From the `sar -c` Command

Field	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.
`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 example shows output from the `sar -c` command.

```
$ sar -c
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
14:29:12      17       2       2    0.28    0.28    2527    1542
14:30:12      25       2       1    0.50    0.47    1624     295
14:31:12      21       2       2    0.35    0.35    1812     703

Average       21       2       2    0.38    0.37    1987     847
```

▼ How to Check Disk Activity

Display disk activity statistics with the `sar -d` command.

```
$ sar -d
SunOS venus 5.4 prefcs3 sun4c 11/11/94

00:00:02 device %busy avque r+w/s blks/s avwait avserv

01:00:02 fd0        0   0.0     0      0    0.0    0.0
         sd1        0   0.0     0      0   19.6   35.4
         sd3        0   0.0     0      0   10.8   55.6
```

Table 71-10 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 71-10 Output From the `sar -d` Command

Field	**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

These two examples illustrate the `sar -d` output. The first example is from a computer with a non-SCSI (Small Computer System Interface, pronounced “scuzzy”) integral disk; that is, a disk that does not use a SCSI interface. This example illustrates data being transferred from a hard disk (`hdsk-0`) to the floppy disk (`fdsk-0`).

```
$ sar -d
Solaris mysys Solaris 2.5 sun4c    8/11/95
13:46:28  device %busy avque r+w/s blks/s  avwait  avserv
13:46:58  hdsk-0    6   1.6     3      5    13.8    23.7
          fdsk-0   93   2.1     2      4   467.8   444.0
13:47:28  hdsk-0   13   1.3     4      8    10.8    32.3
          fdsk-0  100   3.1     2      5   857.4   404.1
13:47:58  hdsk-0   17    .7     2     41      .6    48.1
          fdsk-0  100   4.4     2      6  1451.9   406.5
Average   hdsk-0   12   1.2     3     18     8.4    34.7
          fdsk-0   98   3.2     2      5   925.7   418.2
```

The following example is from a computer with SCSI integral disks; that is, disks that use a SCSI interface. The example illustrates data being transferred from one SCSI hard disk (`sd00-0`) to another SCSI integral disk (`sd00-1`).

```
$ sar -d
Solaris mysys Solaris 2.5 sun4c    8/11/95
14:16:24  device %busy avque r+w/s blks/s  avwait  avserv
14:16:52 sd00-0      2   1.0     1      3     0.0    17.9
         sd00-1      6   1.1     3      5     2.0    23.9
14:17:21 sd00-0      2   1.0     1      2     0.0    19.6
         sd00-1      6   1.1     3      5     0.2    24.3
14:17:48 sd00-0      3   1.0     1      3     0.3    18.3
         sd00-1      7   1.1     3      5     1.3    25.4
14:18:15 sd00-0      3   1.0     1      3     0.0    17.2
         sd00-1      5   1.0     2      5     0.0    21.6
Average  sd00-0      2   1.0     1      3     0.1    18.2
         sd00-1      6   1.0     3      5     0.9    23.0
```

▼ How to Check Page-Out and Memory

Use the `sar -g` option reports page-out and memory freeing activities (in averages).

```
$ sar -g

SunOS saturn 5.4 prefcs3 sun4c 01/24/95

00:00:03 pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:02    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.

`sar -g` also shows whether inodes are being recycled too quickly, causing a loss of reusable pages.

Output from the `-g` option is described in Table 71-11.

Table 71-11 Output From the `sar -g` Command

Field	**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.

Table 71-11 Output From the `sar -g` Command *(Continued)*

Field	Description
`%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 example shows output from the `sar -g` command.

```
$ sar -g
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
15:29:13     0.00     0.00     0.35     8.18   0.00
16:29:12     1.20     2.20     3.35     3.40   0.00
```

▼ How to Check Kernel Memory Allocation

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

SunOS saturn 5.4 prefcs3 sun4c 01/24/95

00:00:03 sml_mem   alloc   fail    lg_mem     alloc   fail  ovsz_alloc   fail
01:00:02 1245184  955332      0   3661824   2786336      0     2412544      0
```

Output from the -k option is described in Table 71-12.

Table 71-12 Output From the sar -k Command

Field	**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

The following is an example of `sar -k` output.

```
$ sar -k
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12  sml_mem  alloc  fail  lg_mem  alloc  fail ovsz_alloc fail
14:29:12   95232   73472     0  311296  198656    0     180224    0
14:30:12   95232   75120     0  311296  198656    0     180224    0
14:31:12   95232   73600     0  311296  197632    0     180224    0

Average    95232   74064     0  311296  198314    0     180224    0
```

▼ How to Check Interprocess Communication

Use the `sar -m` command to report interprocess communication activities.

```
$ sar -m
SunOS solaris 5.4 prefcs3 sun4c    01/24/95
00:00:03   msg/s  sema/s
01:00:02    0.00    0.05
```

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 Table 71-13.

Table 71-13 Output From the `sar -m` Command

Field	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 example shows output from the `sar -m` command.

```
$ sar -m
Solaris mysys 2.0 sun4c    08/22/95

14:28:12   msg/s  sema/s
14:29:12    0.00    0.00
14:30:12    0.00    0.00
14:31:12    0.00    0.00

Average     0.00    0.00
```

▼ How to Check Page-In Activity

Use the `sar -p` command to report page-in activity which includes protection and translation faults.

```
$ sar -p

SunOS saturn 5.4 prefcs3 sun4c 01/24/95

00:00:03 atch/s pgin/s ppgin/s pflt/s vflt/s slock/s
01:00:02   0.00   0.00    0.00   0.49   1.20    0.00
```

The reported statistics from the `-p` option are described in Table 71-14.

Table 71-14 Output From the `sar -p` Command

Field	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 example shows output from `sar -p`.

```
$ sar -p
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
14:29:12    1.17   12.87   12.87    5.67   11.28    1.15
14:30:12    1.67    7.08    7.08    9.12    6.33    0.67
14:31:12    1.37   12.48   12.48    6.83   10.78    1.03

Average     1.40   10.81   10.81    7.21    9.46    0.95
```

▼ How to Check Queue Activity

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

SunOS saturn 5.4  prefcs3   sun4c 01/24/95

00:00:03 runq-sz %runocc swpq-sz %swpocc
01:00:02     1.1       0
```

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 Table 71-15.

Table 71-15 Output From the sar -q Command

Field	**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.
swpq-sz	The average number of swapped out LWPs.
%swpocc	The percentage of time LWPs are swapped out.

Example—Checking Queue Activity

The following 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
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12 runq-sz %runocc swpq-sz %swpocc
14:29:12     1.2      53    1       100
14:30:12     1.3      38
14:31:12     1.1      37

Average      1.2      43
```

▼ How to Check Unused Memory

Use the `sar -r` command to report the number of memory pages and swap-file disk blocks that are currently unused.

```
$ sar -r

SunOS saturn 5.4 prefcs3 sun4c 01/24/95

00:00:03 freemem freeswap
01:00:02     983   187590
```

Output from the `-r` option is described in Table 71-16.

Table 71-16 Output From the `sar -r` Command

Field	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
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12 freemem freeswap
14:29:12     268    3034
14:30:12     351    3009
14:31:12     297    3033

Average      306    3025
```

How to Check CPU Utilization

Display CPU utilization with the `sar -u` command.

```
$ sar -u

SunOS saturn 5.4 prefcs3 sun4c  01/24/95

00:00:03    %usr    %sys    %wio %idle
01:00:02 0 1 0 99
```

(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 Table 71-17.

Table 71-17 Output From the `sar -u` Command

Field	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
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12    %usr    %sys    %wio   %idle
14:29:12      22      27      18      32
14:30:12       6      24      13      57
14:31:12       8      28      19      45

Average       12      27      17      45
```

▼ How to Check System Table Status

Use the `sar -v` command to report the status of the process table, inode table, file table, and shared memory record table.

```
$ sar -v

SunOS saturn 5.4  prefcs3 sun4c   01/24/95

00:00:03 proc-sz      ov   inod-sz   ov  file-sz ov  lock-sz
01:00:02  56/426       0  1311/1311   0  372/372  0   0/0
```

Output from the `-v` option is described in Table 71-18.

Table 71-18 Output From the `sar -v` Command

Field	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.

Table 71-18 Output From the sar -v Command *(Continued)*

Field	Description
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 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
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12 proc-sz ov inod-sz ov file-sz ov lock-sz
14:29:12  28/200  0 297/300  0  63/0    0  6/0
14:30:12  30/200  0 297/300  0  65/0    0  6/0
14:31:12  28/200  0 296/300  0  63/0    0  6/0
```

▼ How to Check Swap Activity

Use the `sar -w` command to report swapping and switching activity.

```
$ sar -w

SunOS saturn 5.4 prefcs3 sun4c    01/24/95

00:00:03  swpin/s  bswin/s  swpot/s  bswot/s  pswch/s
01:00:02     0.00      0.0     0.00      0.0       12
```

Target values and observations are described in Table 71-19.

Table 71-19 Output From the `sar -w` Command

Field	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
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12 swpin/s pswin/s swpot/s pswot/s pswch/s
14:29:12    0.00     0.0    0.00     0.0      22
14:30:12    0.00     0.0    0.00     0.0      12
14:31:12    0.00     0.0    0.00     0.0      18

Average     0.00     0.0    0.00     0.0      18
```

▼ How to Check Terminal Activity

Use the `sar -y` command to monitor terminal device activities.

```
$ sar -y

SunOS saturn 5.4 prefcs3 sun4c   01/24/95

00:00:03 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:02       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 Table 71-20.

Table 71-20 Output From the `sar -y` Command

Field	**Description**
`rawch/s`	Input characters (raw queue), per second.
`canch/s`	Input characters processed by canon (canonical queue) per second.
`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 example shows output from the `sar -y` command.

```
$ sar -y
Solaris mysys Solaris 2.5 sun4c    08/22/95

14:28:12 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
14:29:12       0       1     157       1       3       0
14:30:12       0       2      34       2       2       0
14:31:12       0       1      11       1       2       0

Average        0       1      67       1       2       0
```

How to Check Overall System Performance

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.

▼ How to Set Up Automatic Data Collection

1. **Become root.**
2. **Using the editor of your choice, open the** `/etc/init.d/perf` **file, which contains the** `sadc` **start-up instructions. Verify that the following lines are uncommented:**

   ```
   MATCH=`who -r|grep -c “[234][ ]*0[ ]*[S1]”`
   if [ ${MATCH} -eq 1 ]
   then
   su sys -c “/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`”
   fi
   ```

   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. **Using the editor of your choice, open the** `/var/spool/cron/crontabs/sys` **file (the system** `crontab` **file). Uncomment the following lines:**

   ```
   # 0 * * * 0-6 /usr/lib/sa/sa1
   # 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
   ```

   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.

Monitoring Network Performance 72

This chapter describes the how to monitor network performance. This is a list of the step-by-step instructions in this chapter.

How to Check the Response of Hosts on the Network	*page 1440*
How to Send Packets to Hosts on the Network	*page 1441*
How to Capture Packets From the Network	*page 1442*
How to Check the Network Status	*page 1443*
How to Display NFS Server and Client Statistics	*page 1446*

72

Monitoring Network Performance

Table 72-1 describes the commands available for monitoring network performance.

Table 72-1 Network Monitoring Commands

Command	Use This Command To ...
`ping`	Look at the response of hosts on the network.
`spray`	Test the reliability of your packet sizes. It can tell you whether packets are being delayed or dropped.
`snoop`	Capture packets from the network and trace the calls from each client to each server.
`netstat`	Display network status, including state of the interfaces used for TCP/IP traffic, the IP routing table, and the per-protocol statistics for `UDP`, `TCP`, `ICMP`, and `IGMP`.
`nfsstat`	Display a summary of server and client statistics that can be used to identify NFS problems.

▼ How to Check the Response of Hosts on the Network

Check the response of hosts on the network with the `ping` command.

```
$ ping hostname
```

If you suspect a physical problem, you can use `ping` to find the response time of several hosts on the network. If the response from one host is not what you would expect, you can investigate that host. Physical problems could be caused by:

- Loose cables or connectors
- Improper grounding
- Missing termination
- Signal reflection

For more information about this command, see the `ping(1M)` man pages.

Examples—Checking the Response of Hosts on the Network

The simplest version of `ping` sends a single packet to a host on the network. If it receives the correct response, it prints the message *host* `is alive`.

```
$ ping elvis
elvis is alive.
```

With the `–s` option, `ping` sends one datagram per second to a host. It then prints each response and the time it took for the round trip. For example:

```
$ ping –s pluto
64 bytes from pluto (123.456.78.90): icmp_seq=0. time=10. ms
64 bytes from pluto (123.456.78.90): icmp_seq=5. time=0. ms
64 bytes from pluto (123.456.78.90): icmp_seq=6. time=0. ms
^C
----pluto PING Statistics----
8 packets transmitted, 8 packets received, 0% packet loss
round-trip (ms) min/avg/max = 0/2/10
```

▼ How to Send Packets to Hosts on the Network

Test the reliability of your packet sizes with the `spray` command.

```
$ spray [ -c count -d interval -l packet_size ] hostname
```

In this command,

`-c` *count* Is the number of packets to send.

`-d` *interval* Is the number of microseconds to pause between sending packets. If you don’t use a delay, you may run out of buffers.

`-l` *packet_size* Is the packet size.

hostname Is the system to send packets.

For more information about this command, see the `spray(1M)` man pages.

Example—Sending Packets to Hosts on the Network

The following example sends 100 packets to a host (`-c 100`) with each packet having a size of 2048 bytes (`–l 2048`). The packets are sent with a delay time of 20 microseconds between each burst (`–d 20`).

```
$ spray -c 100 -d 20 –l 2048 pluto
sending 100 packets of length 2048 to pluto ...
no packets dropped by pluto
279 packets/sec, 573043 bytes/sec
```

▼ How to Capture Packets From the Network

To capture packets from the network and trace the calls from each client to each server, use `snoop`. This command provides accurate time stamps that allow some network performance problems to be isolated quickly. For more information, see `snoop(1M)`.

```
# snoop
```

Dropped packets could be caused by insufficient buffer space, or an overloaded CPU.

▼ How to Check the Network Status

Display network status information, such as statistics about the state of network interfaces, routing tables, and various protocols, with the `netstat` command.

```
$ netstat [-i] [-r] [-s]
```

In this command,

`-i` Displays the state of the TCP/IP interfaces.

`-r` Displays the IP routing table.

`-s` Displays statistics for the UDP, TCP, ICMP, and IGMP protocols.

For more information, see the `netstat(1M)` man pages.

Examples—Checking the Network Status

The following example shows output from the `netstat –i` command, which displays the state of the interfaces used for TCP/IP traffic.

```
$ netstat -i
Name  Mtu  Net/Dest      Address        Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 software       localhost      1280   0     1280   0     0      0
le0   1500 loopback       venus          1628480 0     347070 16    39354  0
```

This display shows how many packets a machine has transmitted and received on each interface. A machine with active network traffic should show both `Ipkts` and `Opkts` continually increasing.

Calculate the network collisions rate by dividing the number of collision counts (`Collis`) by the number of out packets (`Opkts`). In the above example, the collision rate is 3.5 percent. A network-wide collision rate greater than 5 to 10 percent can indicate a problem.

Calculate the input packet error rate by dividing the number of input errors by the total number of input packets (Ierrs/Ipkts). The output packet error rate is the number of output errors divided by the total number of output packets (Oerrs/Opkts). If the input error rate is high (over 0.25 percent), the host may be dropping packets.

The following example shows output from the netstat -s command, which displays the per-protocol statistics for the UDP, TCP, ICMP, and IGMP protocols.

```
UDP                                                             0
       udpInDatagrams      =   61321    udpInErrors
       udpOutDatagrams     =   6783

       tcpRtoAlgorithm     =    4       ttcpRtoMin      =    50
       tcpRtoMax           =    60000   tcpMaxConn      =    -1
       .
       .
       .

IP     ipForwarding        =   1        ipDefaultTT     =   255
       ipInReceives        =   13429    ipInHdrError    =     0
                               8        s
       .
       .
       .

ICMP   icmpInMsgs          =   116      icmpInErrors    =  0
       icmpInCksumErrs     =    0       icmpInUnknow
                                        n



IGMP:
       0 messages received
       0 messages received with too few bytes





       0 membership reports sent
```

The following example shows output from the `netstat -r` command, which displays the IP routing table.

```
Routing Table:
Destination       Gateway          Flags    Ref     Use   Interface
---------------   ------------     -----    ---     ---   --------
--
localhost         localhost        UGHD     0         0   lo0
earth-bb          sleepy           U        3         1
software          pluto            U        3       147   lo0
224.0.0.0         pluto            UG       3         0   lo0
default           mars             UG       0        18
default           earth            UG       0        30
default           venus            UG       0        18
default           neptune          UG       0        26
default           saturn           UG       0         3
```

The fields in the `netstat –r` report are described in Table 72-2.

Table 72-2 Output From the `netstat -r` Command

Field		**Description**
`Flags`	`U`	The route is up
	`G`	The route is through a gateway
	`H`	The route is to a host
	`D`	The route was dynamically created using a redirect
`Ref`		Shows the current number of routes sharing the same link layer
`Use`		Indicates the number of packets sent out
`Interface`		Lists the network interface used for the route

▼ How to Display NFS Server and Client Statistics

The NFS distributed file service uses a remote procedure call (RPC) facility which translates local commands into requests for the remote host. The remote procedure calls are synchronous. That is, the client application is blocked or suspended until the server has completed the call and returned the results. One of the major factors affecting NFS performance is the retransmission rate.

If the file server cannot respond to a client's request, the client retransmits the request a specified number of times before it quits. Each retransmission imposes system overhead, and increases network traffic. Excessive retransmissions can cause network performance problems. If the retransmission rate is high, you could look for:

- Overloaded servers that take too long to complete requests
- An Ethernet interface dropping packets
- Network congestion which slows the packet transmission

Use `nfsstat -c` to show client statistics, and `nfsstat -s` to show server statistics. Use `netstat –m` to display network statistics for each file system. For more information, see the `nfsstat(1M)` man pages.

Examples—Displaying NFS Server and Client Statistics

The following example displays RPC and NFS data for the client, `pluto`.

```
$ nfsstat -c

Client rpc:
calls  badcalls  retrans badxid  timeout  wait    newcred timers
6888   123       10      51      101      0        0       138

Client nfs:
calls  badcalls  nclget  nclcreate
6765   0         6765    0

null   getattr   setattr root   lookup    readlink  read
0  0%  1364 20%   4 0%   0 0%   1643 24%   928 13%  1622%

wrcache write   create   remove  rename   link      symlink
0 0%    14 0%    11 0%    1 0%     0 0%     0 0%     0 0%

mkdir   rmdir   readdir  fsstat
1 0%    0 0%    2535 37% 10 21%
```

The output of the `nfsstat -c` command is described in Table 72-3.

Table 72-3 Output of the `nfsstat -c` Command

Field	Description
`calls`	Shows the total number of calls sent.
`badcalls`	The total number of calls rejected by RPC.
`retrans`	The total number of retransmissions. For this client, the number of retransmissions is less than 1 percent (10 time-outs out of 6888 calls). These may be caused by temporary failures. Higher rates may indicate a problem.
`badxid`	The number of times that a duplicate acknowledgment was received for a single NFS request.
`timeout`	The number of calls that timed out.

Table 72-3 Output of the `nfsstat -c` Command

Field	Description
`wait`	The number of times a call had to wait because no client handle was available.
`newcred`	The number of times the authentication information had to be refreshed.
`timers`	The number of times the time-out value was greater than or equal to the specified time-out value for a call.
`readlink`	The number of times a `read` was made to a symbolic link. If this number is high (over 10 percent), it could mean that there are too many symbolic links.

The following example shows output from the `nfsstat -m` command.

```
pluto$ nfsstat -m
/usr/man from pluto:/export/svr4/man
 Flags:   hard,intr,dynamic read size=8192, write size=8192,  retrans = 5
 Lookups: srtt=14 (35ms), dev=4 (20ms), cur=3 (60ms)
 Reads:   srtt=17 (42ms), dev=6 (30ms), cur=5 (100ms)
 All:     srtt=15 (37ms), dev=7 (35ms), cur=5 (100ms)
```

This output of the `nfsstat -m` command, which is displayed in milliseconds, is described in Table 72-4:

Table 72-4 Output of the `nfsstat -m` Command

Field	Description
`srtt`	The smoothed average of the round-trip times
`dev`	The average deviations
`cur`	The current “expected” response time

- `srtt` is the smoothed average of the round-trip times
- `dev` is the average deviations
- `cur` is the current “expected” response time

If you suspect that the hardware components of your network are creating problems, you need to look carefully at the cabling and connectors.

Tuning Kernel Parameters 73

This chapter describes the procedures for tuning kernel parameters. This is a list of the step-by-step instructions in this chapter.

How to List the Kernel Parameters	*page 1450*
How to Change the Value of a Kernel Parameter	*page 1451*
How to Set the Value of a Kernel Module Variable	*page 1451*
How to Tune the Interprocess Communication Parameters	*page 1454*
How to Tune Memory Management Parameters	*page 1455*
How to Tune Miscellaneous Parameters	*page 1456*

▼ How to List the Kernel Parameters

Display the current kernel parameters values by using the sysdef command.

```
# sysdef -i
* Hostid
  53001b80
*
* sun4c Configuration
* Devices
packages (driver not attached)
    disk-label (driver not attached)
    deblocker (driver not attached)
    obp-tftp (driver not attached)
              .
              .
              .
* Loadable Objects
drv/arp
    hard link:  strmod/arp
drv/bpp
              .
              .
              .
* System Configuration
 swap files
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t1d0s1   32,9       8 205352 205352
*
* Tunable Parameters
  561152maximum memory allowed in buffer cache (bufhwm)
     426maximum number of processes (v.v_proc)
      99maximum global priority in sys class (MAXCLSYSPRI)
               .
               .
               .
* Utsname Tunables
*    5.4  release (REL)
  minnie  node name (NODE)
   SunOS  system name (SYS)
 prefcs3  version (VER)
               .
               .
               .
```

▼ How to Change the Value of a Kernel Parameter

1. **Become root.**
2. **Add a line to the** `/etc/system` **file in the form:**

   ```
   set parameter=value
   ```

3. **Reboot the system.**
   The kernel parses the `/etc/system` file during autoconfiguration and overrides the default value for the parameters specified in this file.

Example—Changing the Value of a Kernel Parameter

The following line in the `/etc/system` file sets the value of the `max_nprocs` to `500` parameter.

```
set max_nprocs=500
```

▼ How to Set the Value of a Kernel Module Variable

1. **Become root.**
2. **Add a line to the** `/etc/system` **file in the form:**

   ```
   set module_name:variable=value
   ```

3. **Reboot the system.**
   The kernel parses the `/etc/system` file during autoconfiguration and overrides the default value for the parameters specified in this file.

Example—Setting the Value of a Kernel Module Variable

The following line in the `/etc/system` file sets the value of the `msginfo_msgmap` parameter in the `msgsys` module to `150`.

```
set msgsys:msginfo_msgmap=150
```

Buffer Cache Parameters

The bufhwm parameter specifies the maximum size for buffer cache memory usage expressed in units of 1K bytes. The default is 2% of physical memory. Use sar(1M) to measure the buffer cache statistics.

UFS File System Parameters

Table 73-1 describes the tunable UFS parameters.

Table 73-1 UFS File System Parameters

Parameter	Description
ufs_ninode	Maximum size of the inode table (default = max_nprocs + 16 + maxusers+ 64)
ncsize	Number of dnlc entries (default = max_nprocs+16+maxusers + 64); dnlc is the directory-name lookup cache.

STREAMS Parameters

Table 73-2 describes the tunable STREAMS parameters.

Table 73-2 STREAMS Parameters

Parameter	Default	Description
nstrpush	9	The maximum number of STREAMS pushes allowed.
strmsgsz	0	The maximum size for the STREAMS message that a user can create. A value of 0 indicates no upper bound. This parameter may disappear entirely in a future release.
strctlsz	1024	The maximum size of the ctl part of a message.
strthresh	0	The maximum amount of dynamic memory that the STREAMS subsystem can consume, in bytes. Once this threshold is passed, any pushes, opens, and writes on a STREAMS devices will fail for non-root processes. A value of 0 means no limit.
sadcnt	16	Number of sad devices.

Interprocess Communication (IPC™) Parameters

Table 73-3 describes the tunable interprocess communication parameters.

Table 73-3 Interprocess Communication Parameters

Parameter Type	Parameter	Default	Description
Message Queue	msginfo_msgmap	100	Number of entries in the message map
	msginfo_msgmax	2048	Maximum message size
	msginfo_msgmnb	4096	Maximum bytes on queue
	msginfo_msgmni	50	Number of message queue identifiers
	msginfo_msgssz	8	Segment size of a message (should be a multiple of the word size)
	msginfo_msgtql	40	Number of system message headers
	msginfo_msgseg	1024	Number of message segments (must be < 32768)
Semaphore Facility			
	seminfo_semmap	10	Number of entries in the semaphore map
	seminfo_semmni	10	Number of semaphore identifiers
	seminfo_semmns	60	Number of semaphores in the system
	seminfo_semmnu	30	Number of undo structures in the system
	seminfo_semmsl	25	Maximum number of semaphores, per id
	seminfo_semopm	10	Maximum number of operations, per semaphore call
	seminfo_semume	10	Maximum number of undo entries, per process
	seminfo_semvmx	32767	Semaphore maximum value
	seminfo_semaem	16384	Maximum value for adjustment on exit
Shared Memory			
	shminfo_shmmax	1048576	Maximum shared memory segment size
	shminfo_shmmin	1	Minimum shared memory segment size
	shminfo_shmmni	100	Number of shared memory identifiers
	shminfo_shmseg	6	Segments, per process

▼ How to Tune the Interprocess Communication Parameters

1. **Become root.**
2. **Add a line to the `/etc/system` file using the syntax described in Table 73-4.**

Table 73-4 Tuning Interprocess Communication Parameters

Parameter Type	Parameter	Tuning Syntax
`Message Queue`	`msgsys`	`set msgsys:msginfo_`*variable*=*value*
`Semaphore Facility`	`semsys`	`set semsys:seminfo_`*variable*=*value*
`Shared Memory`	`shmsys`	`set shmsys:shminfo_`*variable*=*value*

3. **Reboot the system.**
   The kernel parses the `/etc/system` file during autoconfiguration and overrides the default value for the parameters specified in this file.

Memory Management Parameters

Table 73-5 describes the tunable memory management parameters.

Table 73-5 Memory Management Parameters

Parameter	Default	Description
`lotsfree`	scaled based on physical memory	If `freemem` drops below `lotsfree`, the system starts to steal pages from processes.
`tune_t_fsflushr`	`30`	Rate at which `fsflush` is run, in seconds
`tune_t_minarmem`	`25`	The minimum available resident (not swappable) memory needed to avoid deadlock, in pages
`tune_t_minasmem`	`25`	The minimum available swappable memory needed to avoid deadlock, in pages
`tune_t_flckrec`	`512`	The maximum number of active `frlocks`

Note – Since the Solaris 2.4 release, the tune_t_gpgslo parameter has been replaced by a more complicated criteria for swapping based on the number of runnable threads. More information on this new criteria will be available for the Solaris 2.5 FCS release.

The freemem parameter is defined in pages. Utilities like vmstat translates freemem into bytes from pages.

▼ How to Tune Memory Management Parameters

1. **Become root.**
2. **Add a line to the /etc/system file using the following syntax.**

   ```
   set tune:variable=value
   ```

3. **Reboot the system.**
   The kernel parses the /etc/system file during autoconfiguration and overrides the default value for the parameters specified in this file.

Miscellaneous Parameters

Table 73-6 describes tunable miscellaneous parameters.

Table 73-6 Miscellaneous Parameter

Parameter	Default	Description
lwp_default_stksize	8192	Size of the kernel stack for lwps. Do not adjust this value unless there is a kernel overflow. The value is expressed in bytes and must be a multiple of PAGESIZE bytes.
npty	48	Total number of 4.x pseudo-ttys configured
pt_cnt	48	Total number of 5.x pseudo-ttys configured

▼ How to Tune Miscellaneous Parameters

1. **Become root.**
2. **Add a line to the** `/etc/system` **file using the following syntax.**

   ```
   set parameter=value
   ```

3. **Reboot the system.**
   If you changed device related kernel parameters, you need to use the `-r` option when booting the system. When the system boots, the kernel parses the `/etc/system` file during autoconfiguration and overrides the default value for the parameters specified in this file.

Example—Tuning Miscellaneous Parameters

The following line in the `/etc/system` file sets the value of the `pt_cnt` parameter to `200`.

```
set pt_cnt=200
```

The Scheduler 74

This chapter contains reference information for the SunOS 5.x scheduler. This is a list of the overview information in this chapter.

About the Scheduler	*page 1457*
Scheduler Class Policies	*page 1459*
Scheduler Configuration	*page 1461*

About the Scheduler

The *scheduler* (or dispatcher) is the portion of the kernel that controls the allocation of the CPU to processes. It determines when processes run and for how long, depending on their assigned priorities. Priorities are based on scheduling class and process behavior. Four scheduling classes are supported by default: timesharing, system, real-time and interactive.

The scheduler has an overriding effect on the performance of a system.

Note – The fundamental scheduling entity is the kernel thread. For single-threaded processes, scheduling the kernel thread is synonymous with process scheduling.

The SunOS 5.x scheduler controls the order in which processes run and the amount of CPU time each process may use before another process can run.

The scheduler allocates CPU time to processes according to the scheduling policies defined for each scheduling class. Associated with each scheduling class is a set of priority levels or queues. Ready-to-run processes are moved among these queues. Within a class, you can view these queues as a contiguous set of priority levels. These priority levels are mapped into a set of global scheduling priorities.

The global priority of a process determines when it runs—the scheduler runs the process with the highest global priority that is ready to run. Processes with numerically higher priorities run first, and processes with the same priority run using a round robin scheduling policy.

Once the scheduler assigns a process to a CPU, the process runs until one of the following events occur:

- The process uses up its time slice.
- The process blocks waiting for an event (for example, I/O) or a suspended lock.
- The process is preempted by a higher-priority process.

By default, all real-time processes have higher priorities than any system process, and all system processes have higher priorities than any timesharing process.

A process inherits its scheduler parameters from its parent process, including its scheduler class and its priority within that class. A process changes class only from a user request (with the `priocntl` command or system call). The system manages the priority of a process based on user requests and the policy associated with the scheduling class of the process.

Scheduler Class Policies

The following sections describe the scheduling policies of the three default classes: timesharing, system, and real-time.

Timesharing Class Policies

In the default configuration, the initialization process (`init`) belongs to the timesharing class. Because processes inherit their scheduler parameters, all user login shells—and consequently the processes run from those shells—begin as timesharing processes.

The goal of the timesharing policy is to provide good response time for interactive processes and good throughput for processes that use a lot of CPU time. The scheduler tries to divide the CPU's time fairly between processes, subject to the priorities associated with the processes. Those with higher priorities get more attention than those with lower priorities. However, to prevent any one job (process) from hogging the CPU, the scheduler can move jobs from high priorities to low priorities and vice versa.

The scheduler switches CPU allocation frequently enough to provide good response time, but not so frequently that it spends too much time doing the switching. Time slices are typically on the order of a few hundredths of a second.

The timesharing policy changes priorities dynamically and assigns time slices of different lengths. Once a process has started, its timesharing priority varies according to how much CPU time it's getting, how much time it's spending in queues, and other factors. The scheduler raises the priority of a process that "sleeps." (A process sleeps, for example, when it starts an I/O operation such as a terminal read or a disk read.) Entering sleep states frequently is characteristic of interactive tasks such as editing and running simple shell commands. On the other hand, the timesharing policy lowers the priority of a process that uses the CPU for long periods without sleeping.

The default timesharing policy gives larger time slices to processes with lower priorities. A process with a low priority is likely to be stuck in the CPU. Other processes get the CPU first, but when a lower-priority process finally gets the CPU, it gets a bigger chunk of time. If a higher-priority process becomes ready to run during a time slice, however, it preempts the running process.

The scheduler manages timesharing processes using parameters in the timesharing parameter table `ts_dptbl`. This table contains information specific to the timesharing class. It is automatically loaded into core memory from the `TS_DPTBL` loadable module located in the `/kernel/sched` directory.

System Class Policies

The system class uses a fixed-priority policy to run kernel processes such as servers, and housekeeping processes such as the page daemon. Their priorities are not dynamically adjusted like timesharing processes. The system class is reserved for use by the kernel, and users may neither add nor remove a process from the system class. Priorities for system-class processes are set up in the kernel code for the kernel processes, and, once established, these priorities do not change. (User processes running in kernel mode are not in the system class.)

Real-Time Class Policies

The SunOS 5.x operating system uses a real-time scheduling policy as well as a timesharing policy. Real-time scheduling allows users to set fixed priorities on a per-process basis, so that critical processes can run in predetermined order. The real-time scheduler never moves jobs between priorities. Real-time priorities change only when a user requests a change (using the `priocntl` command). Contrast this fixed-priority policy with the timesharing policy, in which the system changes priorities to provide good interactive response time.

The user process with the highest real-time priority always gets the CPU as soon as it can be run, even if other processes are ready to run. An application can be written so that its real-time processes have a guaranteed response time from the operating system.

Note – As long as there is a real-time process ready to run, no process and no timesharing process runs. Other real-time processes can run only if they have a higher priority. Real-time processes managed carelessly can have a dramatic negative effect on the performance of timesharing processes.

The real-time policy gives higher-priority processes smaller time slices, by default. The higher priorities are allocated to real-time processes that are driven by external events. The operating system must be able to respond instantly to I/O. The lower-priority real-time processes are those that need more computation time. If a process with the highest priority uses up its time slice, it runs again because there is no process with a higher priority to pre-empt it.

The scheduler manages real-time processes by using parameters in the real-time parameter table `rt_dptbl`. This table contains information specific to the real-time class. It is automatically loaded into core from the `RT_DPTBL` loadable module located in the `/kernel/sched` directory.

Scheduler Configuration

This section describes the parameters and tables that control the scheduler configuration. A basic assumption is that your work load is reasonable for your system resources, such as CPU, memory, and I/O. If your resources are inadequate to meet the demands, reconfiguring the scheduler won't help.

You can display or change (fine tune) the scheduler parameters in a running system for both the timesharing and real-time classes by using the `dispadmin` command. Changes made by the `dispadmin` command do not survive a reboot. To make permanent changes in scheduler configuration, you must change the scheduler parameter tables in the appropriate loadable module: `TS_DPTBL` or `RT_DPTBL` provided in the `/kernel/sched` directory. See `ts_dptbl(4)` and `rt_dptbl(4)` for instructions on replacing these modules.

The primary user command for controlling process scheduling is `priocntl(1)`. With this command, a user can start a process at a specified priority or manipulate the priorities of running processes. You can find out what classes are configured on your system with the `priocntl -l` command. The primary function call for controlling process scheduling is `priocntl(2)`.

See "Managing Processes" on page 1389," for examples of using the `priocntl` command. See *System Interfaces Guide* for a detailed descriptions of real-time programming, and the `dispadmin(1M)` and `priocntl(1)` commands.

Default Global Priorities

The following table shows the scheduling order and ranges of global priorities for each scheduler class.

Table 74-1 Scheduling Order and Global Priorities

Scheduling Order	**Global Priority**	**Scheduler Class**
First	159	Real-Time
	.	
	.	
	.	
	100	
	99	System
	.	
	.	
	.	
	60	
	59	Timesharing
	.	
	.	
	.	
Last	0	

How Global Priorities Are Constructed

When your operating system is built, it constructs the global priorities from the tunable parameters and scheduler parameter tables described in the following sections. There isn't any command that will show you this complete global priority table. However, the `dispadmin` command displays the priorities (from 0 to *n*) specific to the real-time and timesharing classes. You can display the global priority of an active process with the `ps -cl` command.

Initial Global Priorities of Processes

A timesharing process inherits its scheduling class and priority from its parent process. The `init` process is the first process to entire the timesharing class.

System processes initially run with a priority that depends on the process's importance (which is programmed into the kernel). The most important system processes start with a priority at or near the top of the system class range.

Tunable Parameters

This section describes the tunable parameters that control scheduler configuration. To change any of these kernel parameters, enter a line in the `/etc/system` file with the format:

`set` *parameter=value*

See `system(4)` for more information.

The parameters described in this section control aspects of process scheduling, timesharing policy, and real-time policy.

The initial priority of a real-time process is determined when the process is put into the real-time scheduling class.

The `-p` option of the `priocntl` command is used to specify the relative priority within the real-time class.

This is added to the base priority of the real-time class, which by default is 100. For example:

```
priocntl -e -c RT -p 20 command
```

would put the command into execution at a real-time priority of 120.

Process Scheduling Parameters

The following kernel parameters control aspects of process scheduling:

- `maxclsyspri`

maxclsyspri is the maximum global priority of processes in the system class. When the kernel starts system processes, it assigns their priorities using the value of maxclsyspri as a reference point. maxclsyspri must have a value of 39 or greater, because the kernel assumes that the total range of system class priorities is at least 40.

If you change this parameter, you must rebuild the scheduling class tables with values that correspond to the maximum priorities that you assign.

- sys_name

  sys_name is the character string name of the system scheduler class. The default value of sys_name is SYS.

Timesharing Policy

The following parameter is specified in the TS loadable module, which controls the timesharing policy:

- ts_maxupri

  ts_maxupri specifies the range within which users may adjust the priority of a timesharing process, using the priocntl(l) command or the priocntl(2) system call. The valid range for the user-supplied priority in the timesharing class is from +ts_maxupri to –ts_maxupri. The default value of ts_maxupri is 20 (which sets the range between +20 and –20, emulating the behavior of the older, less general scheduler interfaces, nice and setpriority.)

  The value of ts_maxupri is independent of the configured number of global timesharing priorities. In the default configuration, there are 0-59 timesharing priorities, but users may adjust their priorities only within a range of –20 to +20, relative to the system-calculated priority of the process. See “How to Designate Priority” on page 1401 for more information.

  To change the value of this parameter, enter a line in /etc/system with the format:

```
set TS:ts_maxupri=value
```

Real-Time Policy

The following parameter is specified in the RT loadable module, which controls the real-time policy:

- `rt_maxpri`

  `rt_maxpri` specifies the maximum priority to assign to real-time processes. The default value of `rt_maxpri` is 159.

  If you change this parameter, you must rebuild the scheduling class tables with values that correspond to the maximum priorities that you assign.

  To change the value of this parameter, enter a line in the `/etc/system` file with the format:

  ```
  set RT:rt_maxupri=value
  ```

Scheduler Parameter Tables

The scheduler tables are described in Table 74-2.

Table 74-2 Scheduler Parameters

Table	Used to Manage ...
rt_dptbl	Real-time processes
ts_dptbl	Timesharing processes
ts_kmdpris	Sleeping timesharing processes that own critical resources

These tables define scheduling policy by setting the scheduling parameters to use for real-time and timesharing processes. The parameters specify how much CPU time processes get at different priority levels.

Default time slices for the priority levels are specified in the `ts_dptbl` and `rt_dptbl` configuration tables, which are defined in the `TS_DPTBL` and `RT_DPTBL` loadable modules. These modules are automatically loaded from the `/kernel/sched` directory into the kernel as needed.

The time slices are specified in units (quanta) with a resolution defined by a "resolution" line. The default resolution is 1000, which means the time quantum values are interpreted as milliseconds. This is derived from the reciprocal of the specified resolution in seconds. The quanta are rounded up to the next integral multiple of the system clock's resolution in clock ticks. (The system clock ticks `HZ` times per second, where `HZ` is a hardware-dependent constant defined in the `param.h` header file.) For example, if the clock tick is 10 milliseconds, 42 quanta is rounded up to 50 milliseconds.

Timesharing Parameter Table

A default version of the `ts_dptb`, is delivered with the system in `/kernel/sched/TS_DPTBL`. The default configuration has 60 timesharing priorities.

The `dispadmin -c TS -g` command displays a sample `ts_dptbl` table.

```
# ts_quantum ts_tqexp ts_slpret ts_maxwait ts_lwait PRIORITY
                                                       LEVEL
    200        0        59        0         50        #  0
    200        0        59        0         50        #  1
    200        0        59        0         50        #  2
    200        0        59        0         50        #  3
    200        0        59        0         50        #  4
    200        0        59        0         50        #  5
    200        0        59        0         50        #  6
    200        0        59        0         50        #  7
    200        0        59        0         50        #  8
    200        0        59        0         50        #  9
    160        0        59        0         51        #  10
    160        1        59        0         51        #  11
    160        2        59        0         51        #  12
    160        3        59        0         51        #  13
    160        4        59        0         51        #  14
    160        5        59        0         51        #  15
    160        6        59        0         51        #  16
    160        7        59        0         51        #  17
    160        8        59        0         51        #  18
    160        9        59        0         51        #  19
    120        10       59        0         52        #  20
    120        11       59        0         52        #  21
    120        12       59        0         52        #  22
    120        13       59        0         52        #  23
    120        14       59        0         52        #  24
    120        15       59        0         52        #  25
    120        16       59        0         52        #  26
    120        17       59        0         52        #  27
    120        18       59        0         52        #  28
    120        19       59        0         52        #  29
    80         20       59        0         53        #  30

# ts_quantum ts_tqexp ts_slpret ts_maxwait ts_lwait PRIORITY
                                                       LEVEL
    80         21       59        0         53        #  31
    80         22       59        0         53        #  32
    80         23       59        0         53        #  33
    80         24       59        0         53        #  34
    80         25       59        0         54        #  35
    80         26       59        0         54        #  36
    80         27       59        0         54        #  37
    80         28       59        0         54        #  38
    80         29       59        0         54        #  39
    40         30       59        0         55        #  40
    40         31       59        0         55        #  41
    40         32       59        0         55        #  42
    40         33       59        0         55        #  43
    40         34       59        0         55        #  44
    40         35       59        0         56        #  45
    40         36       59        0         57        #  46
    40         37       59        0         58        #  47
    40         38       59        0         58        #  48
    40         39       59        0         58        #  49
    40         40       59        0         59        #  50
    40         41       59        0         59        #  51
    40         42       59        0         59        #  52
    40         43       59        0         59        #  53
    40         44       59        0         59        #  54
    40         45       59        0         59        #  55
    40         46       59        0         59        #  56
    40         47       59        0         59        #  57
    40         48       59        0         59        #  58
    40         49       59        0         59        #  59
```

Figure 74-1 Sample `ts_dptbl` Table

Table 74-3 describes the fields in the `ts_dptbl` table.

Table 74-3 Fields in the `ts_dptbl` Table

Field Name	Description
`ts_quantum` (runtime)	Contains the time slice (in milliseconds by default) that a process at a given priority is allowed to run before the scheduler reevaluates its priority. If the process uses up its entire time slice, it is put on the expired-level (`ts_tqexp`) queue. Time slices run from 40 milliseconds for the highest priority (`59`) to 200 milliseconds (`0`) for the lowest priority.

Table 74-3 Fields in the ts_dptbl Table *(Continued)*

Field Name	Description
ts_tqexp (expired level)	Determines the new process priority for a process whose time slice has expired. If a process uses its whole time slice without sleeping, the scheduler changes its priority to the level indicated in the ts_tqexp column. The expired level is lower than the prior level. For example, a process with a priority of 30 that used up its time slice (80 milliseconds) will get a new priority of 20.
ts_slpret (sleep level)	Determines the priority assigned to a process when it returns from sleep. A process may sleep during certain system calls or when waiting for I/O (for example, servicing a page fault or waiting for a lock). When a process returns from sleep, it is always a given a priority of 59.
ts_maxwait (wait time)	Specifies the number of seconds a process will be left on a dispatch queue without its time slice expiring. If it does not use its time slice (in ts_maxwait seconds), its new priority will be set to ts_lwait. This is used to prevent a low-priority process from being starved of CPU time.
ts_lwait (wait level)	Contains the new priority for a ready-to-run process that has exceeded the maximum wait time (ts_maxwait) without getting its full time slice.
PRIORITY LEVEL	Contains global priorities. Processes put in queues at the higher priority levels run first. The global priorities run from a high of 59 to a low of 0. This is the only column in the table that is not tunable.

Real-Time Parameter Table

A default version of rt_dptbl is delivered with the system in the /kernel/sched/RT_DPTBL loadable module.

The dispadmin -c RT -g command displays rt_dptbl information similar to the following.

```
# TIME QUANTUM                          PRIORITY
# (rt_quantum)                            LEVEL
      1000                     #            0
      1000                     #            1
      1000                     #            2
      1000                     #            3
      1000                     #            4
      1000                     #            5
      1000                     #            6
      1000                     #            7
      1000                     #            8
      1000                     #            9
       800                     #           10
       800                     #           11
       800                     #           12
       800                     #           13
       800                     #           14
       800                     #           15
       800                     #           16
       800                     #           17
       800                     #           18
       800                     #           19
       600                     #           20
       600                     #           21
      600                     #           22
       600                     #           23
       600                     #           24
       600                     #           25
       600                     #           26
       600                     #           27
       600                     #           28
       600                     #           29
       400                     #           30
```

```
# TIME QUANTUM                          PRIORITY
# (rt_quantum)                            LEVEL
       400                     #           31
       400                     #           32
       400                     #           33
       400                     #           34
       400                     #           35
       400                     #           36
       400                     #           37
       400                     #           38
       400                     #           39
       200                     #           40
       200                     #           41
       200                     #           42
       200                     #           43
       200                     #           44
       200                     #           45
       200                     #           46
       200                     #           47
       200                     #           48
       200                     #           49
       100                     #           50
       100                     #           51
       100                     #           52
       100                     #           53
       100                     #           54
       100                     #           55
       100                     #           56
       100                     #           57
       100                     #           58
       100                     #           59
```

Figure 74-2 Sample rt_dptbl Table

Table 74-4 describes the fields in the real-time parameter table.

Table 74-4 Fields in the `rt_dptbl` Table

Field Name	Description
`rt_glbpri`	Contains global priorities. Processes put in queues at the higher priority levels run first. Note that the `dispadmin` command, which you can use to display the table, shows only the relative priorities within the class, and not the global priorities. This column cannot be changed with `dispadmin`.
`rt_qntm`	Describes the default time slice (in milliseconds) a process with this priority (`rt_glbpri`) may run before the scheduler gives another process a chance. The time slice for a real-time process can be specified with the `-t` option of the `priocntl` command.

Kernel-Mode Parameter Table

The scheduler uses the kernel-mode parameter table, `ts_kmdpris`, to manage sleeping timesharing processes. A default version of `ts_kmdpris` is delivered with the system, in the `/kernel/sched/TS_DPTBL` loadable module, and is automatically built into the kernel as part of system configuration. See the `ts_dptbl(4)` man page for more information.

Note – The kernel assumes that it has at least 40 priorities in `ts_kmdpris`. It panics if it does not.

The kernel-mode parameter table is a one-dimensional array of global priorities from 60 through 99. If a process owns a critical resource, it is assigned a kernel priority so that it can release the resource as soon as possible. Critical resources are:

- An exclusive lock on a page
- A read lock on a readers/writer lock

Prior to SunOS 5.3, processes were assigned kernel priorities while they were asleep. This ensured that the resources they were waiting for were not paged out before they had a chance to execute again.

In order to do this after SunOS 5.3, processes return from sleep with the highest time-share priority (59).

Index

Symbols

- (minus sign), 1075
  - file permissions symbol, 1160
  - file type symbol, 1161

# DISK SAMPLES column (daily usage report), 1256

# in crontab file, 1345

# OF PROCS column (daily usage report), 1256

# OF SESS column (daily usage report), 1256

# OFF column (daily report), 1254

# ON column (daily report), 1254

# OF PROCS column (daily usage report), 1256

# SESS column (daily report), 1254

# OF SESS column (daily usage report), 1256

* (asterisk)
  - in crontab file, 1345
  - wildcard character, 1196

+ (plus sign)
  - /etc/hosts.equiv file syntax, 1075

, in crontab file, 1345

. (dot)
  - path variable entry, 1158
  - rcp command syntax, 1096 to 1100

/etc/inittab file, 1132

/etc/saf/_sactab file, 1132

/etc/utmp file, 1134

/usr/aset/masters/uid_aliases file, 1186

/var/adm/sulog file, 1172

: (colon) CKLISTPATH_*level* variable, 1196

= (equals sign) CKLISTPATH_*level* variable, 1196

? (question mark) in ASET tune files, 1196

~ (tilde)
  - abbreviated pathnames, 1094 to 1095
  - rcp command syntax, 1096 to 1100

Numerics

4.x systems (running with 5.x systems), 867

A

.a file extension, 1270, 1356

a.out program name, 1258

absolute mode
  - changing file permissions, 1204, 1206
  - described, 1204
  - setting special permissions, 1208

ACB4000 disk controllers
See disk controllers
accept command, 886
accepting print requests, 886, 951, 984
access
See also ACLs (access control lists); permissions; security
root access
displaying attempts on console, 1172, 1230
monitoring su command use, 1172, 1229
restricting, 1172, 1176, 1229
security, 1156 to 1159
ACLs, 1164 to 1165, 1212 to 1221
file access restriction, 1157
firewall setup, 1158 to 1159
login access restrictions, 1165 to 1166
login control, 1156
monitoring system usage, 1157
network control, 1157
path variable setting, 1157 to 1158
physical site security, 1156
reporting problems, 1159
root access restrictions, 1172
root login tracking, 1158
setuid programs, 1158
sharing files, 1176
system logins, 1166 to 1167
to forms
limiting for printers, 991
limiting for users, 990
to printers
adding with commands, 928
deleting, 930
to remote printers
adding, 915
adding with Admintool, 919
adding with commands, 924
access control lists, *See* ACLs (access control lists)
accounting, 1248 to 1265, 1331 to 1341
See also monacct command; prdaily command
See also runacct command
automatic, 1248, 1332
billing users, 1250, 1251, 1256, 1332, 1338
connect, 1249, 1251, 1254, 1255, 1263, 1333
daily, 1251 to 1335
See also prdaily command; runacct command
reports, 1252 to 1262
step-by-step summary of, 1251 to 1252
disk, 1250, 1251, 1252, 1256
files for, 1263 to 1265
See also specific files
fixing corrupted files
tacct file, 1340
wtmp file, 1265, 1333, 1339
installation-dependent local programs, 1334
maintaining, 1339 to 1341
overview, 1248
process, 1249, 1251, 1255 to 1256
raw data, 1251
reports, 1252 to 1262
daily command summary, 1253, 1256 to 1258, 1263, 1264, 1334, 1335
daily report (tty line utilization), 1253 to 1254
daily usage report, 1252, 1255 to 1256
last login report, 1253, 1260
overview, 1252
total command summary (monthly), 1253, 1259, 1264, 1265
setting up, 1332 to 1338
types of, 1248 to 1250
user fee calculation, 1250, 1251, 1256, 1332, 1338
accounts, 1084
/acct directory

/var/adm/acct directory
See /usr/adm/acct directory; /usr/lib/acct directory;

acct file, 1336

acct.h format files, 1260 to 1262

acctcms command, 1334, 1335

acctcom command, 1258, 1260 to 1262

acctcon command, 1264, 1265, 1333

acctdusg command, 1250, 1252, 1256, 1263

acctprc command, 1333

acctwtmp command, 1249, 1251, 1253

ACLs (access control lists), 1212 to 1221
- adding entries, 1218
- changing entries, 1218
- checking entries, 1217
- commands, 1165
- default entries for directories, 1213 to 1214
- deleting entries, 1165, 1219
- described, 1164 to 1165
- directory entries, 1213 to 1214
- displaying entries, 1165, 1220 to 1221
- format of entries, 1212
- setting entries, 1215 to 1216
- valid file entries, 1213

active file, 1263, 1332, 1341

active.*MMDD* file, 1263, 1341

actv field (iostat report), 1382, 1414

Adaptec disk controllers *See* disk controllers

adapter board (serial port), 1105

Add Access To Printer window, 920

Add Local Printer window, 917 to 918

adding
- access to remote printers, 915, 919
  - by using Admintool, 919
  - by using commands, 928
- forms, 981
- local printers, 917
- printer description, 939
- swap space, 1376
- terminfo entry, 1002
- Workstation Info command to Workspace Menu, 1272

ADDR field (ps report), 1390

address space map, 1371, 1393

address translation page faults, 1428

adjusting printer port characteristics, 1000

/adm directory, *See* /usr/adm directory; /var/adm directory

adm file, 1268, 1332, 1336

adm login account, 1167

administering
- character sets, 888, 962 to 972
- fonts, 888, 992 to 996
- forms, 888, 978 to 992
- print filters, 888, 973 to 978
- printers, 929 to 960

Admintool
- ability to define printer with, 870
- Add Access To Printer window, 920
- Add Local Printer window, 917 to 918
- as method to install printers, 913
- Browse menu, 916
- main window, 915 to 916
- prerequisites for using, 915
- Printers window, 916, 917, 919
- starting, 915
- terminals and modems, 1106

alert message priority, 1292

alerts
- for mounting character sets, 968
- for mounting font cartridges, 964
- for mounting forms, 979, 986
- for mounting print wheels, 964
- for printer faults, 914, 945
- printer-related, 903

alias for selectable character set, 963, 970 to 972

aliases file (ASET)
- described, 1186
- example, 1198
- format, 1198
- specification, 1190

alignment pattern
  defining, 1027
  printing, 984
  protection of, 980
alloc field (sar command), 1425
allow list
  for printer access to forms, 991
  for user access to forms, 990
  for user access to printers, 914, 938
anonymous ftp accounts, 1084, 1086
application threads, 1369 to 1370
ASCII file
  as configuration file type, 890
  file content type of, 876
aset command
  See also Automated Security Enhancement Tool (ASET)
  initiating ASET sessions, 1177
  -d option, 1232
  -l option, 1232
  -n option, 1182
  -p option, 1234
  running ASET interactively, 1232 to 1233
  running ASET periodically, 1234
  stop running periodically, 1235
ASET environment file asetenv not found message, 1239
ASET failed message, 1237
ASET startup unsuccessful messages, 1238
ASET task list TASKLIST undefined message, 1239
ASET working directory $ASETDIR missing message, 1238
ASET working directory undefined message, 1238
ASET, *See* Automated Security Enhancement Tool (ASET)
aset.restore utility, 1191
aset: invalid/undefined security level message, 1239
ASETDIR variable, 1193
asetenv file
  described, 1187
  error messages, 1239
  modifying, 1188
  running ASET periodically, 1234
ASETSECLEVEL variable
  described, 1193
  setting security levels, 1194
asterisk (*)
  in crontab file, 1345
  wildcard character, 1196
at command, 1269 to 1270, 1356 to 1363
  automatic scheduling of, 1267
  controlling access to, 1270, 1344, 1361 to 1363
  error messages, 1362
  -l option (list), 1359 to 1360
  -m option (mail), 1357, 1358
  overview, 1269 to 1270, 1344, 1356, 1376
  quitting, 1269
at job files, 1356 to 1361
  creating, 1357 to 1358
  deleting, 1360 to 1361
  described, 1269, 1270
  displaying, 1359 to 1360
  displaying queue of, 1358
  location of, 1270
  submitting, 1356
  verifying, 1357
at.deny file, 1270, 1344, 1361 to 1363
atch/s field (sar command), 1428
/atjobs directory, 1267, 1270, 1344
atq command, 1357, 1358
atrun command, 1270
audio devices, 1156
auth facility, 1292
authentication
  defined, 1173, 1175
  network security, 1173, 1175 to 1176
  remote logins using ftp, 1084, 1086 to 1087
  remote logins using rlogin, 1073 to 1076, 1082

direct vs. indirect logins, 1078
/etc/hosts.equiv file, 1074 to 1075
network vs. remote system authentication, 1073 to 1075, 1078
.rhosts files, 1075 to 1076
types, 1175
authorization
defined, 1173, 1175
network security, 1173, 1175 to 1176
types, 1175
Automated Security Enhancement Tool (ASET), 1177 to 1198, 1231 to 1240
automatic accounting, 1248, 1332
automatic quota turn on, 1247
automatic system activity data collection, 1384, 1438
automatic system activity reporting, 1384 to 1385
automatic system event execution
repetitive events, 1266 to 1269, 1344 to 1355
single events, 1269 to 1270, 1344, 1356 to 1363
avail field (df report), 1383, 1415
avque field (sar command), 1421
avserv field (sar command), 1421
avwait field (sar command), 1421

B

%b field (iostat report), 1414, 1382
b field (vmstat report), 1379, 1406
b file type, 1161
background processing, 1167
backing up .rhosts files, 1076
backup files, 1266, 1358
badcalls field (nfsstat report), 1447
badxid field (nfsstat report), 1447
banner option, 941
banner pages
making optional, 914, 941
nothing else prints, 1061
reasons to turn off, 879
setting
with Admintool, 870, 879
with lpadmin command, 879, 941
with nobanner variable, 942
troubleshooting incorrect output, 1033
turning off, 914, 942
baud settings, 1034
bidirectional modem service, 1104, 1134
billing users, 1250, 1251, 1256, 1332, 1338
/bin directory, 1370, 1393 to 1398
bin login account, 1167
blks/s field (sar command), 1421
block devices
See also specific block devices
checking activity, 1386
block special file type, 1161
BLOCKS READ column (daily command summary), 1258
booting
displaying messages generated during, 1288 to 1289
running sadc command when, 1384
Bourne shell
ASET security level specification, 1194
ASET working directory specification, 1193
bps field (iostat report), 1382, 1412
bread/s field (sar command), 1418
Browse menu (Admintool), 916
bswin/s field (sar command), 1435
bswot/s field (sar command), 1435
buffer cache
checking activity, 1386, 1417 to 1419
overview, 1376
setting maximum memory usage for, 1452
bufhwm parameter, 1452
bus-oriented disk controllers

See also disk controllers
%busy field (sar command), 1421
bwrit/s field (sar command), 1418
bye command, 1085, 1087

C

C field (ps report), 1390
c file type, 1161
C shell
ASET security level specification, 1194
ASET working directory specification, 1193
cable pin configuration, 1038
caches
buffer cache, 1376, 1386, 1417 to 1419, 1452
directory name lookup cache, 1377, 1417, 1452
calls field (nfsstat report), 1447
cancel command, 884
canceling
print requests, 884, 954
by disabling printer, 953
for specific user, 956
remote logins, 1072
canch/s field (sar command), 1436
Cannot decide current and previous security levels message, 1237 to 1238
Cannot expand $ASETDIR to full pathname message, 1238
capacity field (df report), 1383, 1415
cartridges, *See* font cartridges
catman database, 1266
central processing unit, *See* CPU (central processing unit)
centralized
print administration, 869, 913
print configuration, 866
CERT/CC, *See* Computer Emergency Response Team/Coordination Center (CERT/CC)
changing
crontab files, 1346 to 1348
date, 1277, 1279
/etc/system file, 1277, 1281 to 1282
forms paper, 983
kernel parameter values, 1451
message of the day, 1280
number of lock requests, 1277, 1282
number of processes per user, 1277, 1281
number of pseudo-ttys, 1277, 1281
number of shared memory segments, 1277, 1282
priority, 1373 to 1374, 1401 to 1402, 1404
real-time processes, 1460, 1463
timesharing processes, 1373 to 1374, 1402
priority of print requests, 949, 959
quotas for individual users, 1325 to 1326
scheduling classes, 1403, 1458
soft limit time, 1324 to 1325
time, 1277, 1279
character sets
hardware, 962, 963 to 964
managing, 962 to 972
number, 963
selectable, 962 to 963
software, 962
character special file type, 1161
characters transferred
read/write system calls, 1258, 1262, 1419 to 1420
terminal I/O, 1436
chargefee command, 1250, 1251, 1256, 1332, 1338
CHARS TRNSFD column (daily command summary), 1258
chgrp(1) command
described, 1163
syntax, 1203

chmod(1) command
  changing special permissions, 1209 to 1210
  described, 1163
  syntax, 1209
chown(1) command
  described, 1163
  syntax, 1202
cklist.rpt file
  described, 1180, 1185
  format, 1185
CKLISTPATH_*level* variable
  described, 1187, 1189, 1193
  specifying directories, 1189, 1196
ckpacct command, 1252, 1332, 1336
class (printer), 880
  checking status for, 950
  defining with lpadmin command, 943
  not valid for enabling / disabling printer, 953
classes, *See* scheduling classes
CLEANUP state (runacct command), 1334
clients
  displaying information about, 1440, 1446 to 1448
  tracing calls to servers, 1440, 1442
close command, 1085
closewtmp command, 1333
closing remote system connections, 1087
CLS field (ps report), 1374, 1390
CMD field (ps report), 1391
cms file, *See* /var/adm/acct/nite/cms file; /var/adm/acct/sum/cms file
CMS state (runacct command), 1334
cms*n* file, 1265
cmsprev file, 1264
Collis field (netstat report), 1443
collision rate (network), 1443
COMMAND NAME column (daily command summary), 1258
commands
  See also specific commands
  disable, 953
  lpadmin, *See* lpadmin command
  lpfilter, 973
  monitoring usage of, 1263, 1335
comment lines in crontab file, 1345
Computer Emergency Response Team / Coordination Center (CERT / CC), 1159
configuration scheduler, 1461 to 1470
configuring
  ASET, 1187 to 1190
  port monitors, 925, 926
  printer ports
    for PowerPC systems, 872
    for x86 systems, 872
    with Admintool, 917 to 918
connect accounting, 1249, 1251, 1254, 1255, 1263, 1333
  See also accounting
CONNECT state (runacct command), 1333
CONNECT-MINS column (daily usage report), 1255
consistency checking, 1319
console
  displaying su command use on, 1172, 1230
  root access restriction to, 1172, 1229
context switches
  See also switching
  displaying information on, 1379, 1380, 1407
controllers, *See* disk controllers
controlling
  access to at command, 1270, 1344, 1361 to 1363
  access to crontab command, 1268 to 1269, 1344, 1352 to 1355
  client access to printers, 919
  printer access to forms, 991
  processes, 1371, 1396 to 1398
  user access to forms, 990

- copying (remote)
  - using `ftp`, 1085 to 1093
  - using `rcp`, 1093 to 1100
- “copy-on-writes”, 1428
- `core` files
  - automatically deleting, 1356
  - finding and deleting, 1306, 1310 to 1311
  - generation of, 1244, 1284
- `core` program name, 1258
- corrupted files, *See* files, fixing corrupted
- CPU (central processing unit)
  - *See also* scheduler
  - analyzing usage of, 1383
  - defined, 1368
  - displaying information on
    - context switch rate, 1379, 1380, 1407
    - daily command summary, 1258
    - dead processes, 1261, 1262
    - overview, 1387
    - queue activity and, 1430
    - time usage, 1255, 1374, 1379, 1380, 1382, 1383, 1386, 1391, 1407, 1412, 1432 to 1433
  - high-usage processes, 1374
  - when to add capacity, 1430
- `cpu` fields
  - `iostat` report, 1381, 1382, 1412
  - `vmstat` report, 1380, 1407
- `CPU-MINS` column (daily usage report), 1255
- `/crash` directory, 1244, 1284 to 1285, 1311
- `crash` utility, 1288, 1290 to 1291
- crashes, 1244 to 1247, 1283 to 1292
  - causes, 1244
  - creating directory for crash dump files, 1284 to 1285
  - customer service and, 1245, 1288
  - deleting crash dump files, 1311
  - disabling crash dumps, 1284, 1287
  - displaying system information generated by, 1288 to 1291
  - enabling crash dumps, 1284 to 1286
  - examining crash dumps, 1290 to 1291
  - fatal kernel errors, 1244
  - hung system, 1246
  - procedure following, 1245 to 1247
  - rebooting fails after, 1245 to 1246
  - reserving space for crash dump files, 1285
  - saving crash dump information, 1244, 1284 to 1286
  - saving other system information, 1244 to 1245
- creating
  - `at` jobs, 1357 to 1358
  - crash dump information directory, 1284 to 1285
  - `crontab` files, 1346 to 1348
  - directories on remote system, 1085
  - form definitions, 1027
  - forms, 1025
  - print filters, 1011, 1022
- credentials, 1371, 1393
- `crit` message priority, 1292
- critical resources, 1470
- `cron` daemon, 1266, 1270
- `cron.allow` file, 1268 to 1269, 1344, 1352, 1354 to 1355
- `cron.deny` file, 1268, 1344, 1352 to 1353
- `crontab` command, 1266 to 1269
  - accounting commands run by, 1332 to 1337
  - automatic accounting and, 1248
  - controlling access to, 1268 to 1269, 1352 to 1355
    - denying access, 1268, 1344, 1352 to 1353
    - limiting access to specific users, 1268 to 1269, 1344, 1352, 1354 to 1355
    - overview, 1268 to 1269, 1344, 1352
  - `cron` daemon and, 1266
  - daily tasks, 1266

-e option (edit), 1346 to 1347
error messages, 1354
files used by, 1266 to 1268
-l option (list), 1348 to 1349
overview, 1266, 1344
quitting without saving changes, 1347
-r option (remove), 1350 to 1351
scheduling of, 1266
/var/adm maintenance and, 1245

/crontab directory, 1266, 1268, 1344, 1346

crontab files
creating, 1346 to 1348
deleting, 1350 to 1351
described, 1266 to 1268, 1344
displaying, 1348 to 1349
editing, 1346 to 1348
location of, 1268
running ASET periodically, 1178, 1234
stop running ASET periodically, 1235
syntax, 1344 to 1345
verifying existence of, 1347

crypt command, 1164

cs field (vmstat report), 1380, 1407

csh program, 1170 to 1171

.cshrc file, 1158

ctacct.*MMDD* file, 1263, 1333

ctmp file, 1263

ctx cache name, 1411

cur field (nfsstat report), 1448

current user, 1095

customer service, 1245, 1288

customizing
exit codes, printer, 1008
LP print service, 912, 999 to 1027
printer interface program, 1006, 1009, 1010
stty modes, 1007
system logging, 1291 to 1292

D

d file type, 1161

d_passwd file
creating, 1227
described, 1169, 1170 to 1171
disabling dial-up logins temporarily, 1229
/etc/passwd file and, 1171

daemon facility, 1292
See also specific daemons

daemon login account, 1167

daemons
lpd, 1050
lpNet, 926
lpsched, 899, 902, 910
print, 892

daily accounting, 1251 to 1335
See also accounting; prdaily command; runacct command
reports, 1252 to 1262
daily command summary, 1253, 1256 to 1258, 1263, 1264, 1334, 1335
daily report (tty line utilization), 1253 to 1254
daily usage report, 1252, 1255 to 1256
last login report, 1253, 1260
overview, 1252
step-by-step summary of, 1251 to 1252

daily tasks (scheduling with crontab), 1266

date
changing, 1277, 1279
displaying, 1274, 1276
synchronizing with another system, 1277 to 1278

date command
accounting data and, 1249, 1251
described, 1274, 1276, 1277

Day of month field (crontab file), 1344

Day of week field (crontab file), 1344

daycms file, *See*
    /var/adm/acct/nite/daycms file;
    /var/adm/acct/sum/daycms file
daytacct file, 1256, 1263, 1333, 1334, 1335
de field (vmstat report), 1379, 1407
deadfiles file, 1309
deadlock avoidance, 1454
debug message priority, 1292
default printer
  ability to set with Admintool, 870
  setting with Admintool, 878
  setting with lpadmin command, 878, 940
defaults
  See also maximums; minimums
  ACL entries for directories, 1213 to 1214
  at.deny file, 1361
  buffer cache memory usage, 1452
  cron.deny file, 1352
  crontab files, 1268
  lock requests, 1277
  maxusers parameter, 1377
  message of the day, 1280
  nice number, 1374
  pacct file maximum size, 1249
  priority levels, 1372
  real-time parameter table, 1469
  scheduling classes, 1401, 1459
  soft limit time, 1323 to 1325
  sys_name parameter, 1464
  time slices, 1466
  timesharing parameter table, 1466
defective sectors, *See* disk drives, defects
defining
  font cartridges, 965
  print servers, 870
  print wheels, 965
  printer characteristics, 937
delete command, 1085
deleting
  access to printers, 930
  ACL entries, 1165, 1219
  at jobs, 1360 to 1361
  backup files, 1266, 1358
  core files, 1310 to 1311
  crash dump files, 1311
  crontab files, 1350 to 1351
  directories on remote system, 1085
  directory permissions and, 1161
  file permissions for, 1160
  files from remote directories, 1085
  forms, 982
  log files, 1348
  old/inactive files, 1266, 1306 to 1312, 1348
  print filters, 976
  .rhosts files, 1076
  temporary files, 1309 to 1310
deny list
  for printer access to forms, 991
  for user access to forms, 990
  for user access to printers, 914, 938
destination printer, 878
dev field (nfsstat report), 1448
/dev/term/a, 872
/dev/term/b, 872
device drivers, 1424 to 1425
device field (sar command), 1421
device interrupts, 1379, 1380, 1381, 1407, 1411, 1436, 1437
devices
  See also disk drives; tape drives
  system device access control, 1156, 1172
df command, 1296 to 1298, 1383, 1415 to 1416
  examples, 1297 to 1298, 1383, 1415 to 1416
  -F option (unmounted file systems), 1296, 1298
  -g option (statvfs structure), 1296
  -k option (kilobytes), 1296, 1297, 1383, 1415 to 1416
  overview, 1296, 1383, 1415

-t option (total blocks), 1296, 1298
dfstab file, 1176
dial-in modem service, 1104
dial-out modem service, 1104
dial-up passwords, 1168 to 1171, 1227
basic sequence, 1170
creating, 1227
disabling dial-up logins temporarily, 1229
/etc/d_passwd file, 1169, 1170 to 1171
/etc/dialups file, 1169
dialups file
creating, 1227
described, 1169
diff utility (ASET), 1186
dirbk/s field (sar command), 1417
direct disk controllers
See also disk controllers
direct printing, 875, 877
direct remote logins
indirect logins vs.
rlogin, 1076 to 1078
using rlogin, 1082 to 1083
directories
See also specific directories
abbreviated pathnames, 1094 to 1095
ACL entries, 1213 to 1214
ASET files, 1177
checklist task (CKLISTPATH) setting, 1189, 1196
master files, 1186
reports, 1183 to 1185
working directory, 1193, 1232, 1238
crash dump information directory, 1284 to 1285
current working directory for processes, 1371, 1393, 1395, 1396
displaying files and related information, 1163, 1200 to 1201
displaying information about, 1299 to 1300, 1303 to 1305
file type, 1161
ftp commands, 1085
permissions
defaults, 1163
described, 1161
public directories, 1163
remote copying, 1096
setgid permissions, 1163
size of, 1303 to 1305
sticky bit permissions, 1163
temporary, clearing out, 1306, 1309 to 1310
working directory, 1095
directory name lookup cache, 1377, 1417, 1452
disable command, 887, 953
disabling
See also stopping
See also turning off
crash dumps, 1284, 1287
dial-up logins temporarily, 1229
printers, 931, 953
quotas for individual users, 1327 to 1328
disk accounting, 1250, 1251, 1252, 1256
DISK BLOCKS column (daily usage report), 1256
DISK command (acounting program), 1334
disk drives
See also SCSI disk drives
described, 1375
displaying information about, 1381 to 1383, 1412 to 1416, 1421
basic activity statistics, 1379, 1381 to 1383, 1407, 1412 to 1416, 1421
df command, 1296 to 1298, 1383, 1415 to 1416
directory sizes, 1303 to 1305
extended activity statistics, 1382, 1414

file sizes, 1299 to 1300, 1303 to 1305
free disk space, 1296 to 1298, 1383, 1415 to 1416
iostat command, 1375, 1381 to 1382, 1387, 1412 to 1416
mount point, 1297, 1383, 1415
overview, 1375, 1381, 1387
sar command, 1421
UFS file system user allocation, 1304 to 1305
vmstat command, 1379, 1407
file system usage, 1296 to 1298, 1375, 1383, 1415 to 1416
finding and deleting old/inactive files, 1306 to 1312, 1348, 1358
finding files exceeding a size limit, 1302
finding large files, 1301
fragmentation of, 1375
optimizing, 1295 to 1312
overview, 1375
partitions, *See* disk slices; fdisk partitions
quotas, *See* quotas
slices, *See* disk slices; fdisk partitions
slowdowns
avoiding, 1375
checking for, 1432
defined, 1375
space
amount of free, 1296 to 1298, 1383, 1415 to 1416
optimizing usage, 1295 to 1312
disk field (vmstat report), 1379, 1407
disk labels, *See* disk drives, labels
# DISK SAMPLES column (daily usage report), 1256
disk space for print queue, 868
DISK state (runacct command), 1334
disks, *See* disk drives
DiskSuite, 1375
disktacct file, 1250, 1251, 1263, 1334
disktacct.*MMDD* file, 1334
diskusg command, 1250, 1252
dispadmin command
overview, 1373
real-time parameter table display, 1469, 1470
rebooting and, 1461
timesharing parameter table display, 1467
dispatch activity, 1386, 1427 to 1429
dispatcher, *See* scheduler
displaying
acct.h format files, 1260 to 1262
ACL entries, 1165, 1220 to 1221
ASET task status, 1179, 1183
at jobs, 1359 to 1360
at queue, 1358
booting messages, 1288 to 1289
cache flushing information, 1380, 1410 to 1411
context switch information, 1379, 1380, 1407
CPU information, *See* CPU (central processing unit), displaying information on
crash information, 1288 to 1291
crontab files, 1348 to 1349
date, 1274, 1276
directory information, 1299 to 1300, 1303 to 1305
disk information, *See* disk drives, displaying information about
file information, 1299 to 1300, 1303 to 1305, 1306 to 1307
file system information, 1296 to 1298, 1375, 1383, 1415 to 1416
files and related information, 1163, 1200 to 1201
host ID, 1274, 1275
interrupt information, 1379, 1380, 1381, 1407, 1411, 1436, 1437
kernel parameters, 1450
linked libraries, 1371, 1393
LWP information, 1371, 1379, 1380, 1410

memory information, *See* memory, displaying information on
network information, 1387, 1440, 1443 to 1448
operating system information, 1274 to 1275
pacct*n* file, 1260 to 1262
paging information, *See* paging, displaying information on
priority information, 1390, 1400, 1462
process information, *See* processes, displaying information on
queue information, 1386, 1414, 1421, 1429 to 1430
quota information, 1314, 1321, 1322
remote directory contents, 1085
remote directory name, 1085
root access attempts on console, 1172, 1230
scheduling class information, 1390, 1399 to 1400
status of forms, 989
su command use on console, 1172, 1230
swap area information, 1379, 1387, 1407, 1431
swapping information, 1380, 1386, 1410, 1430 to 1431, 1435 to 1436
switching information, 1386, 1435 to 1436
system activity information, 1378, 1385 to 1386
system call information, 1379, 1380, 1386, 1407, 1419 to 1420
system event information, 1380, 1409
system information
  commands for, 1274 to 1276
  WorkStation Information window for, 1272 to 1273
terminal I/O information, 1381, 1386, 1412, 1436 to 1437
time, 1274, 1276
trap rate information, 1380, 1407
tty line I/O information, 1381, 1386, 1412, 1436 to 1437
unmounted file system list, 1296
user's login status, 1224 to 1225
Workstation Information window for, 1272 to 1273

dmesg command, 1288 to 1289

dnlc, *See* directory name lookup cache

dodisk command, 1250
  caution, 1250
  crontab entry that runs, 1336, 1337
  DISK command and, 1334
  files created by, 1250, 1251, 1263, 1334
  overview, 1250, 1251, 1252

domains (remote logins), 1072

dot (.)
  path variable entry, 1158
  rcp command syntax, 1096 to 1100

download filter, 993, 994

downloaded PostScript fonts, 996

downloading
  fonts, 992, 1062
  host-resident fonts, 994

drivers, *See* device drivers

dropped packets, 1442

dtmp file, 1263

du command, 1303 to 1305

dumps, *See* crashes

Duplicate ASET execution scheduled message, 1240

E

editing
  See also changing
  crontab files, 1346 to 1348

edquota command
  changing quotas for individual users, 1325
  disabling quotas for individual users, 1327 to 1328
  overview, 1314, 1315, 1323
  -p option (prototype), 1318
  setting up user quotas, 1317 to 1318
  -t option (time limit), 1325

eeprom (password protection), 1167

eeprom.rpt file, 1181, 1185
emerg message priority, 1292
Emulex disk controllers *See* disk controllers
enable command, 887
enabling
  See also turning on
  crash dumps, 1284 to 1286
  printers, 953
encrypting
  capturing encrypted passwords, 1228
  files, 1164
env.rpt file, 1181, 1185
environment file (ASET)
  described, 1187
  error messages, 1239, 1240
  modifying, 1188
  running ASET periodically, 1234
environment variables
  ASET, 1193 to 1196
    ASETDIR, 1193
    ASETSECLEVEL, 1194
    CKLISTPATH_*level*, 1187, 1189, 1196
    PERIODIC_SCHEDULE, 1189, 1194 to 1195, 1234, 1240
    summary table, 1193
    TASKS, 1188, 1195
    UID_ALIASES, 1187, 1190, 1195
    YPCHECK, 1190, 1195
  LPDEST, 940
  PRINTER, 940
equals sign (=)
  CKLISTPATH_*level* variable, 1196
  file permissions symbol, 1205
err message priority, 1292
error messages
  at command, 1362
  crash messages, 1288 to 1289
  crontab command, 1354
  customizing logging of, 1291 to 1292
  log file for, 1244, 1245, 1288
  priorities for, 1292
  runacct command, 1341
  sources of, 1291 to 1292
  specifying storage location for, 1244, 1291, 1292
error protection, 1332 to 1333
/etc/acct/holidays file, 1255, 1336, 1337
/etc/cron.d/at.deny file, 1270, 1344, 1361 to 1363
/etc/cron.d/cron.allow file, 1268 to 1269, 1344, 1352, 1354 to 1355
/etc/cron.d/cron.deny file, 1268, 1344, 1352 to 1353
/etc/d_passwd file
  creating, 1227
  described, 1169, 1170 to 1171
  disabling dial-up logins temporarily, 1229
  /etc/passwd file and, 1171
/etc/default/login file
  restricting root access to console, 1229
  restricting root access to devices, 1172
/etc/default/su file
  displaying su command use on console, 1172, 1230
  monitoring su command, 1229
/etc/dfs/dfstab file, 1176
/etc/dialups file
  creating, 1227
  described, 1169
/etc/format.dat file, *See* format.dat file
/etc/hosts.equiv file, 1074 to 1075
/etc/hosts.lpd file, 1051
/etc/init.d/acct file, 1336
/etc/init.d/perf file, 1384, 1438
/etc/init.d/sysetup file, 1244, 1286
/etc/logindevperm file, 1156
/etc/lp directory, 869, 890
/etc/lp/classes/*printer-class* file, 943
/etc/lp/default file, 940
/etc/lp/fd directory, 901, 974
/etc/lp/filter.table file, 973

filter added in, 975
filter deleted from, 976
/etc/lp/forms directory, 980
/etc/lp/forms/*form-name* file, 982
/etc/lp/forms/*form-name*/alert.sh file, 986
/etc/lp/forms/*form-name*/allow file, 990
/etc/lp/forms/*form-name*/deny file, 990
/etc/lp/forms/*form-name*/describe file, 981
/etc/lp/interfaces directory, 942
/etc/lp/printers directory, 891
of print client
printer added in, 919
printer deleted from, 930
of print server
printer added in, 917
printer deleted from, 931
/etc/lp/printers/*printer-name*/alert.sh file, 945
/etc/lp/printers/*printer-name*/comment file, 939
/etc/lp/printers/*printer-name*/configuration file, 941, 944
banner page setting entered in, 942
font cartridge added in, 965, 967
font cartridge deleted from, 966
form added in, 984
form deleted from, 983
print wheel added in, 965, 967
print wheel deleted from, 966
/etc/lp/printers/*printer-name*/form.allow file, 991
/etc/lp/printers/*printer-name*/form.deny file, 991
/etc/lp/printers/*printer-name*/users.allow file, 947
/etc/lp/printers/*printer-name*/users.deny file, 947
/etc/lp/pwheels/*charset-name*/alert.sh file, 968
/etc/lp/Systems file, 930
/etc/motd file, 1277, 1280
/etc/nsswitch.conf file, 1074, 1165
/etc/passwd file
ASET checks, 1180
/etc/d_passwd file and, 1171
/etc/password file, 1084
/etc/printcap file, 891, 1055
/etc/rc0.d/K22acct file, 1332
/etc/rc2.d/S22acct file, 1332
/etc/syslog.conf file, 1291 to 1292
/etc/system file
changing, 1277, 1281 to 1282
kernel module variables, 1451
kernel parameter values, 1451, 1454 to 1456
max_nprocs parameter, 1451
maxusers parameter, 1377
number of lock requests, 1277, 1282
number of processes per user, 1277, 1281
number of pseudo-ttys, 1277, 1281
number of shared memory segments, 1277, 1282
scheduler configuration tuning and, 1463 to 1465
/etc/vfstab file, 1316, 1319
events, *See* system events
exec/d field (sar command), 1420
execute permissions
directories, 1161
files, 1160
symbolic mode, 1205
executing, *See* starting
execution log (ASET), 1182 to 1183
exit codes (printer interface), 1008
standard, 1007
table of, 1008
exit command, 1083 to 1084
exiting, *See* quitting; stopping
export command, 1193, 1194

F

fail field (sar command), 1425
failed login attempts, 1166, 1226
fast print filters, 994
fault alerts (printer), 914
fault notification (printer)
- ability to set with Admintool, 870
- setting with Admintool, 878, 917
- setting with lpadmin command, 878, 945
- values for alerts, 937

fault recovery (printer), 871, 882, 914, 937
faults
- *See also* interrupts; traps
- page, 1379, 1407, 1427 to 1429

faults fields (vmstat report), 1380, 1407
fcntl information, 1371, 1393, 1395, 1396
fd fields (iostat report), 1381, 1382, 1412
fd2log file, 1263, 1332, 1341
FEE column (daily usage report), 1256
fee file, 1251, 1263, 1334, 1338
fees (user), 1250, 1251, 1256, 1332, 1338
FEES state (runacct command), 1334
file content type, 875
- ability to set with Admintool, 870
- converted by print filters, 973, 1012
- for common printers, 876
- menu in Admintool, 876
- non-PostScript printers, 877
- PostScript, 876
- simple, 876
- troubleshooting incorrect output, 1033

file systems
- *See also* files; UFS file systems; *specific file systems*
- disk space usage, 1296 to 1298, 1383, 1415 to 1416
- displaying information about, 1296 to 1298, 1375, 1383, 1415 to 1416
- mount point, 1297, 1383, 1415
- network statistics for, 1447 to 1448
- quotas, *See* quotas
- restoring, 1250, 1256, 1332, 1338
- statvfs structure for mounted, 1296
- for swapping, 1376
- unmounted, 1296, 1297

file table, 1433 to 1434
File Transfer Protocol, *See* ftp command; ftp sessions
file transfers, 1167
filename doesn’t exist or is not readable message, 1239
files
- *See also* file systems; *specific files*
- accounting, 1263 to 1265
- backup, 1266
- checking access operations, 1386, 1416 to 1417
- deleting old / inactive, 1266, 1306 to 1312, 1348
- displaying information about
  - listing, 1299 to 1300
  - listing newest, 1306 to 1307
  - size, 1299 to 1300, 1303 to 1305
- finding and deleting old / inactive, 1306 to 1312, 1348, 1358
- finding files exceeding a size limit, 1302
- fixing corrupted
  - tacct file, 1340
  - wtmp file, 1265, 1333, 1339
- fstat and fcntl information display, 1371, 1393, 1395, 1396
- lock requests, 1277, 1282
- quotas, *See* quotas
- sa filename prefix, 1384, 1385, 1438
- size of, 1299 to 1300, 1303 to 1305
- usage monitoring, 1250, 1252, 1256
- used by LP print service, 892

files and file systems
- abbreviated pathnames, 1094 to 1095
- ACL entries

adding or modifying, 1218
checking, 1217
deleting, 1165, 1219
displaying, 1165, 1220 to 1221
setting, 1215 to 1216
valid entries, 1213
administration commands, 1163
ASET checks, 1179 to 1180
ownership
changing, 1163
setgid permission and, 1162
setuid permission and, 1162
permissions
absolute mode, 1204, 1206
changing, 1163, 1204 to 1208
defaults, 1163
described, 1160
setgid, 1162 to 1163
setuid, 1162
sticky bit, 1163
symbolic mode, 1204, 1205, 1207 to 1208
umask setting, 1163 to 1164
security, 1160 to 1165, 1199 to 1221
access restriction, 1157
ACLs (access control lists), 1164 to 1165, 1212 to 1221
changing ownership, 1202 to 1203
changing permissions, 1204 to 1208
directory permissions, 1161
displaying file information, 1163, 1200 to 1201
encryption, 1164
file permissions, 1160
file types, 1161
overview, 1155
special file permissions, 1162 to 1163, 1208 to 1211
umask default, 1163 to 1164
user classes, 1160
sharing files, 1176
system logins, 1167

files entry in nsswitch.conf file, 1165
file-sz field (sar command), 1433
FILTER variable, 922
filtering
printing without, 875
See also print filters
filters
See also print filters
download, 993, 994
find command, 1302, 1306 to 1308, 1310 to 1311
finding files with setuid permissions, 1210 to 1211
searching .rhosts files, 1080
finding
files exceeding a size limit, 1302
large files, 1301
old / inactive files, 1306 to 1312, 1348, 1358
finding and deleting, old / inactive files, 1358
firewall systems
ASET setup, 1174, 1181
described, 1158, 1173, 1174
packet smashing, 1174 to 1175
trusted hosts, 1174
firewall.rpt file, 1182, 1185
/fiscal directory, 1263, 1265
fiscrpt*n* file, 1265
flags, 1393
Flags field (netstat report), 1445
font cartridges, 963
alerts for mounting, 964, 968
defining, 965
mounting, 966
naming, 964
unmounting, 966
fonts
downloaded PostScript, 996
downloading, 992, 995, 1062
host-resident, 993, 994, 995
installing, 996

managing, 992
permanently downloaded, 993
PostScript, 992
printer-resident, 993
styles, 962

forcing programs to quit, 1246

fork/exec flag, 1261

fork/s field (sar command), 1420

forms
adding, 978, 981
alerts for mounting, 979
allowing user access, 990
changing, 978
controlling access to, 980
creating, 1025
default values for, 1026
definition, creating, 1027
deleting, 978, 982
denying user access, 990
displaying attributes of, 980
limiting printer access to, 991
limiting user access to, 990
managing, 978
mounting, 979, 983
paper (loading and removing), 983
printer access required for, 981
setting alerts for mounting, 986
tracking forms mounted, 902
unmounting, 983
viewing status of, 989

fr field (vmstat report), 1379, 1407

fragmentation, 1375

frame buffers, 1156

free field (vmstat report), 1379, 1407

free list, 1379, 1407

freeing memory, 1376, 1386, 1423

freemem field (sar command), 1431

freemem parameter, 1455

freeswap field (sar command), 1431

frlocks, 1454

frozen system, 1246
See also crashes

fsck command, 1266

fsflush command, 1454

fstat information, 1371, 1393, 1395, 1396

ftp command
authenticating remote logins, 1084
authentication, 1175
described, 1084, 1085
interrupting logins, 1072
opening remote system connections, 1086 to 1087
remote logins compared to rlogin and rcp, 1084

ftp command interpreter, 1085

ftp sessions
anonymous ftp accounts, 1084, 1086
authenticating remote logins, 1084
closing remote system connections, 1087
commands, 1085
copying files
from remote system, 1087 to 1090
to remote system, 1090 to 1093
opening remote system connections, 1086 to 1087
remote logins compared to rlogin and rcp, 1084

G

get command
copying from remote systems, 1087 to 1089
described, 1085
example, 1089

getfacl command
described, 1165
displaying ACL entries, 1220 to 1221
examples, 1220 to 1221
verifying ACLs set on files, 1216

getty, 1106

global priorities
construction of, 1462
defined, 1399
displaying, 1400, 1462
initial, 1463
maximum, 1464

real-time processes, 1462, 1470
scheduler and, 1458, 1462 to 1463
system processes, 1462
timesharing processes, 1462, 1468

group 14, 915

group ACL entries
default entries for directories, 1214
described, 1213
setting, 1215 to 1216

group identifier numbers (GIDs), 1167

groups
ASET check, 1180
changing file ownership, 1203

H

hard disk
recommended for print server, 869

held signals, 1371, 1393

help command, 1085

hex+symbolic stack trace, 1371, 1393

high ASET security level, 1178

history log (print requests), 894

hog factor
See also free hog slice
reporting on, 1258, 1261

HOG FACTOR column (daily command summary), 1258

holidays file, 1255, 1336, 1337

hostid command, 1274, 1275

host-resident fonts
downloading, 994
PostScript, 996

hosts
checking response of, 1440
in /etc/hosts.equiv file, 1074 to 1075
sending packets to, 1440, 1441
trusted hosts, 1174

hosts.equiv file, 1074 to 1075

Hour field (crontab file), 1344

hung system, 1246
See also crashes

hyphen (-) in crontab file, 1345

I

I/O (input/output) devices
See also disk drives; terminals
described, 1368

I/O-intensive processes, 1376

IA processes, *See* interactive processes

ICMP protocol, 1444

id field
iostat report, 1382, 1413
vmstat report, 1380, 1407

IDE controllers *See* disk controllers

identification number, 1372

identifying *See* names

%idle field (sar command), 1432

idle mode (CPU), 1380, 1382, 1407, 1413, 1432

Ierrs field (netstat report), 1444

iget/s field (sar command), 1417

igets command
number of requests for inodes not in directory name lookup table, 1417
with page flushes, 1424

IGMP protocol, 1444

in field (vmstat report), 1380, 1407

increasing, *See* changing

indirect remote logins, 1076 to 1078

info message priority, 1292

init process, 1249, 1251, 1459, 1463

init program, 1132

initializing print servers, 925

initializing quotas, 1315, 1319

initiating, *See* starting

inode table
size of, 1377, 1452
status of, 1433 to 1434

inodes, 1417, 1423, 1424

inod-sz field (sar command), 1433

input/output devices, *See* I/O devices

installing
  local printers, 917
  PostScript fonts, 996
interactive processes, 1372, 1373
  See also processes
interactively running ASET, 1232 to 1233
Interface field (netstat report), 1445
interface program (printer)
  customizing, 1006, 1009, 1010
  standard, 1010
interface script
  standard, 922
  SVR4, 921
Internet firewall setup, 1158
Internet Protocol address, 926
interprocess communication
  checking, 1386, 1426
  increasing shared memory, 1282
  parameters, 1453
  tuning parameters, 1454
interrupting programs, 1246
interrupting remote logins, 1072
interrupts, 1379, 1380, 1381, 1407, 1411, 1436, 1437
invalid/undefined security level message, 1239
iostat command, 1381 to 1382, 1412 to 1416
  basic information display, 1381 to 1382, 1412 to 1413
  overview, 1375, 1381 to 1382, 1387, 1412
  -xtc option (extended), 1382, 1414
IP address, *See* Internet Protocol address
IP routing table, 1445
IPC, *See* interprocess communication
IPI disk controllers
  See also disk controllers
Ipkts field (netstat report), 1444
ISP-80 disk controllers, *See* disk controllers

K

K22acct file, 1332
kbytes field (df report), 1383, 1415
KCORE-MINS column (daily usage report), 1255
kern facility, 1292
kernel
  See also scheduler
  accounting data and, 1251
  buffer cache and, 1376
  fatal kernel errors, 1244
  memory allocation for, 1386, 1424 to 1426
  module variable setting, 1451
  paging and, 1375
  parameters
    buffer cache, 1452
    changing value of, 1451
    IPC, 1453
    kernel-mode parameter table, 1470
    memory management, 1454 to 1455
    miscellaneous, 1455 to 1456
    overview, 1377 to 1378
    scheduler configuration, 1463 to 1465
    STREAMS, 1452
    tuning, 1454 to 1456, 1463 to 1465
    UFS, 1377, 1452
  system process priority and, 1373
Kernel Memory Allocator, 1386, 1424 to 1426
kernel thread
  displaying information on, 1379, 1430, 1435
  scheduling and, 1390, 1457
  structures, 1369, 1390
/kernel/sched directory, 1460, 1461, 1466
kernel-mode parameter table, 1470
keyboards, 1156
kill command, 1372, 1398

killing processes, 1370, 1372, 1398
kilobytes
  file system disk usage in, 1383, 1415 to 1416
  read / write statistics in, 1382, 1414
klwp structure, 1369
KMA, *See* Kernel Memory Allocator
Korn shell
  ASET security level specification, 1194
  ASET working directory specification, 1193
Kr/s field (iostat report), 1382, 1414
ksh program, 1170 to 1171
kthread structure, 1369
Kw/s field (iostat report), 1382, 1414

L

l file type, 1161
large files, 1301
last login report, 1253, 1260
lastdate file, 1264, 1332
lastlogin command, 1334
latest directory (ASET), 1185
lcd command, 1085
leaks, memory, 1425
lg_mem field (sar command), 1425
libraries (linked), 1371, 1393
lightweight processes, *See* LWPs
limiting
  printer access to forms, 991
  user access to forms, 990
limits, *See* defaults; maximums; minimums
LINE column (daily report), 1254
line discipline, 1133
line usage monitoring, 1249, 1253 to 1254, 1264, 1333, 1335
lineuse file, 1264, 1333, 1335
linked libraries, 1371, 1393
linking remote logins, 1076
listen port monitor
  adding, 1144
  disabling, 1146
  enabling, 1146
  killing, 1145
  listing, 1145
  removing, 1147
  starting, 1146
listen service
  See also port monitor
  adding, 1147
  checking setup of, 1048, 1058
  disabling, 1149
  enabling, 1149
  listing, 1148
  registering, 926
  types of, 926
listenBSD service, 926
listenS5 service, 926
listing
  See also displaying
  files and directories, 1299 to 1300, 1306 to 1307
  kernel parameters, 1450
  processes being executed, 1391 to 1392
listing, *See* displaying
loading form paper, 983
local printer
  adding by using Admintool, 917
  adding by using commands, 924, 927
  defined, 867
  example of setting up, 925
  task map for setting up, 914
local printing, 904
lock file, 1264, 1332, 1341
lockl file, 1332
locks
  as critical resources, 1470
  increasing number of requests for, 1277, 1282
  software, 1428
lock-sz field (sar command), 1434
log file, 1264

log files
ASET execution log, 1182 to 1183
cleaning out, 903
codes in request log, 895
deleting automatically, 1348
for LP print service, 893, 1032
lpnet, 903
lpsched, 903
monitoring su command, 1172, 1229
print queue, 893
print request history log, 894
requests, 903
log.*MMDD* file, 1264
logging in
displaying user's login status, 1224 to 1225
remote logins, 1072 to 1087
authentication (ftp), 1084
authentication (rlogin), 1073 to 1076
direct vs. indirect (rlogin), 1076 to 1078
finding who is logged in, 1081 to 1082
ftp commands, 1085
interrupting, 1072
linking logins, 1076
opening ftp connection, 1086 to 1087
using rlogin, 1082 to 1083
root login
access restrictions, 1172
account, 1167, 1172
restricting to console, 1172, 1229
tracking, 1158
security
access restrictions, 1165 to 1166
saving failed attempts, 1166, 1226
system access control, 1156
system device access control, 1156
tracking root login, 1158
system logins, 1166 to 1167
logging out (remote systems), 1083 to 1084
login command, 1249, 1251
.login file, 1158
login file
restricting root access to console, 1229
restricting root access to devices, 1172
login monitoring
last login, 1253, 1260, 1334, 1335
number of logins, 1256
time usage, 1249, 1251, 1255, 1333
LOGIN NAME column (daily usage report), 1255
logindevperm file, 1156
loginlog file, 1264, 1334, 1335
overview, 1166
saving failed login attempts, 1166, 1226
logins command
displaying user's login status, 1224 to 1225
displaying users with no passwords, 1225
syntax, 1224, 1225
lotsfree parameter, 1454
low ASET security level, 1178
LP commands, 1065
lp facility, 1292
lp file, 1268
lp login account, 1167
LP print service
checking basic functions of, 1040, 1044
configuration files in, 890
customizing, 912, 999 to 1027
daemons, 892
defined, 888
defining printer characteristics to, 867
directories in, 889
files used by, 892
hung LP commands, 1065

interface program, 902
log files, 893, 1032
overview of, 888 to 912
scheduler, *See* print scheduler
structure of, 889
tracking forms, 902
tracking print wheels, 903
troubleshooting, 1031
See also printer(s), troubleshooting

LP print service commands
as alternative method to install printers, 913
setting up printers by using, 923 to 928
troubleshooting, 1035

LP print spooler, 921

lpadmin command
adding printer description with, 939
adjusting printer port characteristics with, 1001
defining font cartridges with, 965
defining print wheels with, 965
defining printer class with, 943
limiting access to printers with, 947
limiting printer access to forms with, 991
making banner pages optional with, 941
mounting font cartridge with, 967
mounting forms with, 983, 984
mounting print wheel with, 967
setting alerts to mount forms with, 986
setting alerts to mount print wheels with, 968
setting default printer with, 940
setting printer fault alerts with, 945
setting printer fault recovery with, 944
unmounting forms with, 983

lpd daemon, 1050

LPDEST environment variable, 940

lpfilter command, 973

lpNet daemon, 926

lpNet log file, 903, 1032

lpsched daemon, 899, 902, 910

lpsched log file, 903

lread/s field (sar command), 1418

ls command, 1299 to 1300, 1306 to 1307
-l option (size in bytes), 1299 to 1300
-s option (size in blocks), 1299, 1300, 1301
-t option (newest files), 1306

lwp_default_stksize parameter, 1455

LWPs (lightweight processes)
defined, 1369
displaying information on, 1371, 1379, 1380, 1410, 1429, 1430, 1435
kernel stack size for, 1455
processes and, 1369
structures for, 1369
swapping, 1375, 1429, 1430, 1435

lwrite/s field (sar command), 1418

M

madvise command, 1376

mail facility, 1292

managing
fonts, 992
forms, 978
print filters, 974
print requests, 884 to 888, 949
print scheduler, 930
printer-resident fonts, 993

mask ACL entries
default entries for directories, 1214
described, 1213
setting, 1215 to 1216

master files (ASET), 1179, 1186 to 1187

max_nprocs parameter, 1377, 1451

maxclsyspri parameter, 1464

maximums
See also defaults; minimums
acctdusg users processed, 1250
buffer cache memory, 1452

finding files exceeding maximum size, 1302
frlocks, 1454
global priority, 1464
inode table size, 1377
nice number, 1374
pacct file size, 1249
priority, 1401, 1464, 1465
process size, 1377
real-time process priority, 1465
user processes per user-id, 1377
users, 1377
maxuprc parameter, 1281, 1377
maxusers parameter, 1377
MD21 disk controllers *See* disk controllers
mdelete command, 1085
mdmin/s field (sar command), 1436, 1437
MEAN CPU-MIN column (daily command summary), 1258
MEAN SIZE-K column (daily command summary), 1258
medium ASET security level, 1178
memory
See also paging; swapping
buffers, *See* buffer cache
deadlock avoidance, 1454
described, 1368
disk slowdowns and, 1375
displaying information on
amount installed, 1274, 1275 to 1276
daily command summary, 1258
daily usage report, 1255
dead processes, 1261
free space, 1379, 1407
freeing activities, 1386, 1423
kernel allocation, 1424 to 1426
swap activity, 1435 to 1436
swap space, 1379, 1387, 1407, 1431
swapping activities, 1380, 1386, 1410, 1430 to 1431
unused memory, 1431
virtual memory statistics, 1274, 1275 to 1276, 1379, 1387, 1406 to 1408
free list, 1379, 1407
freeing, 1376, 1386, 1423
kernel, *See* kernel
leaks in, 1424 to 1425
parameters, 1454 to 1455
preventing monopolization of, 1376
process structures and, 1369
required for print server, 869
shared
increasing number of segments, 1277, 1282
interprocess communication parameters, 1453, 1454
process virtual memory, 1370
record table, 1433 to 1434
swap areas, *See* swap areas
unused, 1386, 1431
virtual
displaying information on, 1274, 1275 to 1276, 1379, 1387, 1406 to 1408
process, 1370
when to add more, 1423
memory fields (vmstat report), 1379, 1407
MERGE command, 1333
MERGE state (runacct command), 1333
MERGETACCT command, 1334
MERGETACCT state (runacct command), 1334
message of the day (MOTD) facility, 1277, 1280
messages file, 1245, 1288, 1292
messages, *See* error messages; interprocess communication
messages.*n* file, 1288
metadata, 1417
mf field (vmstat report), 1379, 1407
mget command
copying from remote systems, 1087 to 1090
described, 1085

example, 1090
minfree file, 1285
minimums
See also defaults; maximums
memory to avoid deadlocks, 1454
nice number, 1374
priority, 1401
minus sign (-)
/etc/hosts.equiv file syntax, 1075
file permissions symbol, 1160, 1205
file type symbol, 1161
Minute field (crontab file), 1344
MINUTES column (daily report), 1254
mmap command, 1376
modem interrupts, 1436, 1437
modems
bidirectional service, 1104, 1134
dial-in service, 1104
dial-out service, 1104
different ways to use, 1104
menu items in Serial Port Manager, 1113
overview of Serial Port Manager, 1110
setting up, 1119
setting up for use with UUCP, 1121
tools for managing, 1106
troubleshooting tips for, 1127
monacct command
crontab entry that runs, 1336
files used / produced by, 1265, 1335
monthly command summary and, 1256, 1259
overview, 1248
runacct command and, 1252, 1332
scheduling running of, 1332
monitoring
su command use, 1172, 1229
system usage, 1157
Month field (crontab file), 1344
monthly command summary, 1253, 1259
monthly tasks, 1266
MOTD (message of the day) facility, 1277, 1280
motd file, 1277, 1280
mount point, 1297, 1383, 1415
mounted on field (df report), 1383, 1415
mounting
font cartridges, 966
forms, 979, 983
print wheels, 966
mouse (system device access control), 1156
moving print requests, 885, 957
mput command
copying to remote systems, 1090 to 1093
described, 1085
example, 1093
msg/s field (sar command), 1426
msginfo_*xxxxxx* parameters, 1451, 1453, 1454
multiple files (ftp), 1087 to 1093
multiuser state, 1254

N

name service
administration guide, 1031
advangtages of, 913
advantages of, 869
named pipe (FIFO) file type, 1161
namei/s field (sar command), 1417
names
.a file extension, 1270, 1356
directory name lookup cache, 1377
sa filename prefix, 1384, 1385, 1438
ncsize parameter, 1377, 1452
ndquot parameter, 1377
netstat command, 1443 to 1445
-i option (interfaces), 1443 to 1444
overview, 1387, 1440, 1443
-r option (IP routing table), 1445
-s option (per protocol), 1444
network

distributing printers on, 866
planning printers on, 866
network authentication for remote logins, 1073 to 1075, 1078
network listen service, See listen service
network printer
adding, 920 to 922
defined, 920
task map for setting up, 914
network security, 1173 to 1176
authentication, 1173, 1175 to 1176
authorization, 1173, 1175 to 1176
firewall systems, 1173, 1174 to 1175
described, 1158, 1173, 1174
need for, 1158 to 1159
packet smashing, 1174 to 1175
trusted hosts, 1174
issues, 1157
overview, 1155, 1173
networks
commands for monitoring performance, 1440
displaying performance information, 1387, 1440, 1443 to 1448
client statistics, 1446 to 1448
collision rate, 1443
host response, 1440
interface statistics, 1443 to 1445
IP routing table, 1445
server statistics, 1446 to 1448
packets
capturing from network, 1440, 1442
dropped, 1442
error rates, 1444
number transmitted, 1443
reliability testing, 1440, 1441
sending to hosts, 1440, 1441
tracing client calls to servers, 1440, 1442
troubleshooting
hardware components, 1448
high retransmission rate, 1446
newcred field (nfsstat report), 1448
NFS systems (ASET), 1191 to 1192
nfsstat command, 1446 to 1448
-c option (clients), 1446, 1447
-m option (per file system), 1446 to 1448
overview, 1387, 1440, 1446
-s option (servers), 1446
NI field (ps report), 1374, 1390
nice command, 1373 to 1374, 1404
nice number, 1373 to 1374, 1390, 1404
NIS, 869, 913
nis entry in nsswitch.conf file, 1165
NIS+
advantages of, 869, 913
ASET checks, 1190
authentication, 1175
authorization, 1175
nisplus entry in nsswitch.conf file, 1165
/nite directory, *See* /var/adm/acct/nite directory
nlsadmin command, 1136
nobanner option, 879, 941, 942
nobody user, 1176
none message priority, 1292
non-PostScript printers, 877, 962 to 964
npty parameter, 1455
nsswitch.conf file, 1074, 1165
nstrpush parameter, 1452
nuucp login account, 1167

O

Oerrs field (netstat report), 1444
# OF PROCS column (daily usage report), 1256
# OF SESS column (daily usage report), 1256
# OFF column (daily report), 1254
# ON column (daily report), 1254
Online: DiskSuite, 1375

open command, 1085
opening remote system connections, 1085, 1086 to 1087
opening, *See* starting
OpenWindows
  adding Workstation Info command to Workspace Menu, 1272
  displaying system information using Workstation Information window, 1272 to 1273
OpenWindows software, 915
operating system, 1274 to 1275
Opkts field (netstat report), 1444
optimizing disks, 1295 to 1312
options, *See specific options*
other ACL entries
  default entries for directories, 1214
  described, 1213
  setting, 1215 to 1216
outch/s field (sar command), 1436
ov field (sar command), 1434
ovsz_alloc field (sar command), 1425
ownership of files
  ACLs and, 1164 to 1165
  changing, 1163, 1202
  changing group ownership, 1203
owtmp file, 1264

P

p file type, 1161
pacct file, *See* pacct*n* file
pacct*n* file
  displaying, 1260 to 1262
  monitoring size of, 1249, 1252, 1332
  overview, 1249, 1251, 1263, 1333
packet transfers
  firewall security, 1159
  packet smashing, 1174 to 1175
packets, *See* networks, packets
pag cache name, 1411
page daemons
  cycles used by, 1423
  scheduling policies and, 1460
page fields (vmstat report), 1379, 1407
paging
  See also swapping
  defined, 1375
  displaying information on
    general paging activity, 1379, 1407
    page faults, 1379, 1407, 1427 to 1429
    page-ins, 1379, 1386, 1407, 1427 to 1429
    page-outs, 1379, 1386, 1407, 1423
    unused pages, 1431
  priority of, 1372
panic: messages, 1288
par cache name, 1411
parallel printer, 872
parameters, *See* kernel, parameters; scheduling class tables
parity bit, 1034
partitions, *See* disk slices; fdisk partitions
passwd file
  ASET checks, 1180
  /etc/d_passwd file and, 1171
password file, 1084
passwords
  See also security
  authentication for remote logins
    ftp, 1084, 1086
    rlogin, 1073, 1078, 1082
  capturing encrypted passwords, 1228
  dial-up passwords, 1168 to 1171, 1227
    basic sequence, 1170
    creating, 1227
    disabling dial-up logins temporarily, 1229
    /etc/d_passwd file, 1169, 1170 to 1171
    /etc/dialups file, 1169
  displaying users with no passwords, 1225
  eeprom security, 1167
  login security, 1156, 1165, 1166, 1167

  - system logins, 1166, 1167
- `PATH` environment variable, 1181
- path variable, 1157 to 1158
- pathnames
  - `rcp` command
    - absolute vs. abbreviated, 1094 to 1095
    - syntax options, 1095
  - tilde (~) in, 1094 to 1095
- `pcred` command, 1371, 1393
- pending signals, 1371, 1393
- `PERCENT` column (daily report), 1254
- `perf` file, 1384, 1438
- performance, 1367 to 1386, 1405 to 1438
  - activities that are tracked, 1378
  - automatic collection of activity data, 1384, 1438
  - books on, 1368
  - buffer activity, 1386, 1417 to 1419
  - cache flushing, 1380, 1410 to 1411
  - CPU, *See* CPU (central processing unit), displaying information on
  - disk management, 1375, 1381 to 1383, 1407, 1412 to 1416, 1421
  - file access, 1386, 1416 to 1417
  - interprocess communications, 1386, 1426
  - interrupts, 1379, 1380, 1381, 1407, 1411, 1436, 1437
  - kernel parameters, 1377 to 1378
  - manual collection of activity data, 1385 to 1386, 1416 to 1437
  - memory management, 1375 to 1376, 1379, 1406 to 1408, 1423 to 1426, 1431, 1435 to 1436, 1454 to 1455
  - paging, 1379, 1386, 1407, 1423
  - process management, 1369, 1370 to 1372, 1374, 1389 to 1404
  - queue, 1386, 1414, 1421, 1429 to 1430
  - reports on, 1378 to 1386
  - swap areas, 1379, 1387, 1407, 1431
  - swapping, 1380, 1386, 1410, 1429 to 1431, 1435 to 1436
  - system activity monitoring, 1378, 1384 to 1386, 1416 to 1438
  - system calls, 1379, 1380, 1386, 1407, 1419 to 1420
  - system event monitoring, 1380, 1409
  - system tables, 1386, 1433 to 1434
  - terminal I/O, 1381, 1386, 1412, 1436 to 1437
  - tools for monitoring, 1378 to 1386, 1387
  - traps, 1380, 1407
- performance meter, 1387
- `PERIODIC_SCHEDULE` variable
  - described, 1193
  - scheduling ASET, 1189, 1194 to 1195, 1234
- permissions
  - *See also* security
  - ACLs and, 1164 to 1165
  - ASET handling of, 1178, 1179
  - changing file permissions, 1163
    - absolute mode, 1204, 1206
    - symbolic mode, 1204, 1205, 1207 to 1208
  - copying requirements, 1096
  - defaults, 1163
  - directory permissions, 1161
  - file permissions
    - absolute mode, 1204, 1206
    - changing, 1204 to 1208
    - described, 1160
    - special permissions, 1162 to 1163, 1208 to 1211
    - symbolic mode, 1204, 1205, 1207 to 1208
  - `setgid` permissions
    - absolute mode, 1208 to 1210
    - described, 1162 to 1163
    - symbolic mode, 1205
  - `setuid` permissions
    - absolute mode, 1208 to 1210
    - described, 1162

        finding files with permissions set, 1210 to 1211
        security risks, 1162
        symbolic mode, 1205
    special file permissions, 1162 to 1163, 1208 to 1211
    sticky bit, 1163
    tune files (ASET), 1186, 1190
    umask settings, 1163 to 1164
    user classes and, 1160
pfiles command, 1371, 1393, 1395, 1396
pflags command, 1371, 1393
pflt/s field (sar command), 1428
pgfree/s field (sar command), 1423
pgin/s field (sar command), 1428
pgout/s field (sar command), 1423
pgscan/s field (sar command), 1423
physical security, 1156
pi field (vmstat report), 1379, 1407
PID field (ps report), 1390, 1394, 1397, 1398
pin configuration in cables, 1038
ping command, 1081, 1440
pldd command, 1371, 1393
plus sign (+)
    /etc/hosts.equiv file syntax, 1075
    file permissions symbol, 1205
pmadm command
    adding a listen service with, 1147
    adding a ttymon service with, 1141
    disabling a listen service with, 1149
    disabling a ttymon service with, 1144
    enabling a listen service with, 1149
    enabling a ttymon service with, 1143
    explained, 1132
    listing a listen service with, 1148
    listing a ttymon service with, 1142
pmap command, 1371, 1393
po field (vmstat report), 1379, 1407
port
    defined, 1104
    disabling, 1125
    initialization process of, 1134
    initializing, 1123
    removing service, 1126
    See also printer port
    states of (table), 1152
port monitor
    checking setup of, 1048, 1058
    configuring, 925, 926
    defined, 1105
    reasons for configuring, 926
    registering listen service with, 926
    states of (table), 1151
    ttymon and listen (defined), 1106, 1135
ports in /etc/dialups file, 1169
PostScript fonts, 992, 996
    installing, 996
PostScript printers, 962
    character sets for, 963
    default print filters, 974
    example of setting up, 925
    file content type for, 876
    printer type for, 874
PostScript Reverse printer, *See* Reverse PostScript printer
pound sign (#) in crontab file, 1345
power cycling, 1246
power failure recoveries, 1253
ppgin/s field (sar command), 1428
ppgout/s field (sar command), 1423
PPID field (ps report), 1390
prdaily command
    files used by, 1263, 1264
        See also specific files
    line usage reporting and, 1335
    overview, 1248, 1332
    runacct command and, 1332, 1335
pread/s field (sar command), 1418
PRI field (ps report), 1390, 1400
PRIME NUMBER CMNDS column (daily command summary), 1258

PRIME TOTAL CPU-MIN column (daily command summary), 1258
PRIME TOTAL REAL-MIN column (daily command summary), 1258
print administration, centralized, 869, 913
print client
  adding access to printers, 919
  checking configuration of, 1042
  defined, 867
  deleting access to printers, 930
  freeing jobs in, 1068
  setting up, 919 to 920
print configuration
  centralized, 866
  using SunOS 5.x and 4.x systems, 867
print daemons, 892
  See also names of specific daemons
print filters
  adding, 973
  bypassing, 1061, 1062
  changing, 973
  characteristics of, 1019 to 1020
  converting file content type, 973
  creating, 973, 1011, 1022
  defined, 901, 973
  defining, 1015
  deleting, 976
  displaying definition of, 977
  download, 994
  fast, 994
  handling special modes, 1013
  managing, 974
  not provided in SunOS 5.x, 975
  options keywords, 1019
  PostScript, 974
  removing, 973
  required for printer fault recovery, 882, 937
  requirements for, 1014
  restoring, 973
  slow, 994
  templates to define options, 1019
  to convert from troff to PostScript, 1024
  TranScript, 974
  types of, 1012
  used to convert files, 1012
  used to put request on hold, 1013
  viewing information about, 977
print forms, *See* forms
print jobs, *See* print requests
print management, 865 to 912
print queue
  checking request log, 1045
  log of, 893
print request ID, *See* request ID
print requests
  accepting, 951, 984
  canceling, 884, 954
    by disabling printer, 953
    for specific user, 956
  changing priority of, 885, 949, 959
  checking status of, 949
  cleaning out from log file, 903
  IDs, 950
  managing, 884, 949
  moving to another printer, 885, 957
  moving to head of queue, 885
  putting on hold, 885
  rejecting, 931, 951
  scheduling, 901
  setting printer to accept or reject, 886
print scheduler
  if not running, 1040
  managing, 930
  restarting, 883
  starting, 901, 936
  stopping, 901, 936
  updating LP system files, 899
print server
  checking access to, 1044
  checking configuration of, 1042
  checking connections of, 1041
  checking if correctly identified, 1045
  defined, 867, 917
  deleting printer from, 931
  hard disk requirements for, 869
  initializing, 925
  memory requirements for, 869

setting definitions for, 870
setting up, 917 to 918
spooling space required for, 868
swap space required for, 869
system resource requirements for, 868

print service, *See* LP print service

print spooler (SVR4), 921

print wheels, 963
alerts for mounting, 964, 968
defining, 965
mounting, 966
naming, 964
tracking, 903
unmounting, 966

printcap entry, 1055

printer class, 880
checking status for, 950
defining with lpadmin command, 943
not valid for enabling / disabling printer, 953
setting, 871

printer description
ability to set with Admintool, 870
setting with Admintool, 871
setting with lpadmin command, 871, 939

printer destination
ability to set with Admintool, 870
setting with Admintool, 878
setting with lpadmin command, 878

PRINTER environment variable, 940

printer interface program, 902

printer interface script
standard, 922
SVR4, 921

Printer Manager, 869, 913

printer name, 870, 871

printer port, 872
ability to set with Admintool, 870
adjusting characteristics of, 1001
characteristics, adjusting, 1000
configuring, 872
enabling multiple ports, 872
parallel, 872
serial, 872

printer status, 883

printer type, 873
ability to set with Admintool, 870
defined in terminfo database, 892
not in terminfo database, 875
troubleshooting setting of, 1060

printer(s)
accepting print requests, 886
access to
adding with Admintool, 915, 919
adding with commands, 924, 928
deleting, 930
access to forms, 981
administering, 929 to 960
alerts, *See* alerts
allow list, 914, 938
allowing user access, 947
baud settings, 1034
choosing method to install, 869
class, *See* printer class
controlling availability of, 886
default, 940
deleting, 883, 933
deleting client access to, 883
deleting from print server, 930, 931
deny list, 914, 938
denying user access, 947
description, adding, 939
disabling, 887, 931, 953
distributing on network, 866
enabling, 887, 953
fault alerts, 914, 945
fault notification, 878
fault recovery, 882, 914, 937
faults
detecting, 1013
messages, 1009
restarting printing, 1067
file content types
effect on filtering, 1061
incorrect output, 1061
font cartridges, 962

fonts, *See* fonts
interface program
    customizing, 1006, 1009, 1010
    exit codes, 1008
    standard, 1007, 1010
local, 867, 904, 914
    adding by using Admintool, 917
    adding by using commands, 924, 927
network, 914, 920
non-PostScript, 962, 1011
parity bit, 1034
planning on network, 866
PostScript
    character sets for, 963
    treatment of text, 962
print wheels, 962
rejecting print requests, 886
remote, 906, 915
setting definitions for, 937
setting up, 913 to 928
    with Admintool, 915 to 920
    with commands, 923 to 928
settings
    baud, 1034
    parity, 1034
    return, 1034
    `stty`, 1000, 1063, 1064
    tab, 1034
status
    checking, 933, 949
    conflicting messages, 1036
`stty` settings, 1007, 1063, 1064
troubleshooting
    backup in printer queue, 1036
    conflicting status messages, 1036, 1068
    hung LP commands, 1065
    hung printers, 1035, 1066
    incorrect output, 1033, 1060
    incorrect printer type, 1060
    not accepting requests, 1040
    only banner page prints, 1061
    print service commands, 1035
type, *See* printer type
unsupported, 1002
    user access, 914
printer-resident fonts, 993
Printers window, 916, 917
printing
    banner pages, 879
    local (diagram), 904
    processing or stopping, 887
    remote, 906
    special modes, 1013
    spooling directory, 897
    status messages, 1068
    user fee calculation for, 1250, 1338
printing (remote), 1338
`priocntl` command
    -c option (scheduling class designation), 1401, 1403
    -e option (execute), 1401
    -i option (`ID` type), 1402, 1403
    -l option (scheduling class display), 1399 to 1400
    -m option (max/min priority), 1401, 1402
    overview, 1373, 1461
    -p option (priority designation), 1401, 1402, 1463
    -s option (priority upper limit/change priority), 1402, 1403
priority (message), 1292
priority (process)
    changing, 1373 to 1374, 1401 to 1402, 1404
        real-time processes, 1460, 1463
        timesharing processes, 1373 to 1374, 1402
    defaults, 1372
    designating, 1401, 1402
    displaying information on, 1390, 1400, 1462
    global
        construction of, 1462
        defined, 1399
        displaying, 1400, 1462
        initial, 1463
        maximum, 1464

real-time processes, 1462, 1470
scheduler and, 1458, 1462 to 1463
system processes, 1462
timesharing processes, 1462, 1468
maximums, 1401, 1464, 1465
overview, 1372 to 1374, 1399
scheduling classes and, 1372 to 1373, 1401 to 1402, 1458
user-mode priority, 1399
PRIORITY LEVEL field (ts_dptbl table), 1468
priority of print requests, 885, 949, 959
private interfaces, 890
problem-solving, *See* troubleshooting
/proc directory, 1370
/proc file system, 1298
proc structure, 1369, 1390
/proc/bin directory, 1370, 1393 to 1398
process accounting, 1249, 1251, 1255 to 1256
See also accounting; pacct*n* file
See also Spacct*n.MMDD* file
process file system (PROCFS), 1370
PROCESS state (runacct command), 1333
process table, 1433 to 1434
processes
See also scheduler
accounting utilities for, 1249, 1251, 1255 to 1256
address space map, 1371, 1393
application threads and, 1369 to 1370
controlling, 1371, 1396 to 1398
credentials, 1371, 1393
current working directory for, 1371, 1393, 1395, 1396
defined, 1369
displaying information on, 1390 to 1396
acctcom command, 1260 to 1262
daily usage report, 1255 to 1256
dead processes, 1260 to 1262
listing processes being executed, 1391 to 1392
LWPs, 1371, 1379, 1380, 1410, 1429, 1430, 1435
priocntl command, 1399 to 1400
ps command, 1370, 1387, 1390 to 1392, 1400
sar -q command, 1429 to 1430
tool commands, 1370, 1393 to 1396
vmstat command, 1379, 1406 to 1408, 1410
fstat and fcntl information for open files, 1371, 1393, 1395, 1396
identification number for, 1372
init, 1249, 1459, 1463
interactive, 1372, 1373
killing, 1370, 1372, 1398
libraries linked into, 1371, 1393
lightweight, *See* LWPs
maximum size of, 1377
maximum user processes per user-id, 1377
nice number of, 1373 to 1374, 1390, 1404
number per user, 1277, 1281, 1377
priority, 1372 to 1374
changing, 1373 to 1374, 1401 to 1402, 1404
changing real-time process priority, 1460, 1463
changing timesharing process priority, 1373 to 1374, 1402
defaults, 1372
designating, 1401, 1402
displaying information on, 1390, 1400
global priorities, 1399, 1400, 1458, 1462 to 1463, 1464, 1468, 1470
maximums, 1401, 1464, 1465
overview, 1372 to 1374, 1399

    - scheduling classes and, 1372 to 1373, 1399, 1401 to 1402, 1458
    - user-mode priority, 1399
  - real-time, *See* real-time processes
  - restarting, 1371, 1396 to 1398
  - runaway, 1374
  - scheduling classes, 1372 to 1373, 1399 to 1404
    - changing, 1403, 1458
    - changing priority of, 1373 to 1374, 1401 to 1402
    - defaults, 1401
    - described, 1372 to 1373
    - designating, 1401
    - displaying information on, 1390, 1399 to 1400
    - priority levels and, 1372 to 1373, 1399, 1401 to 1402, 1458
    - scheduling policies and, 1372 to 1373, 1459 to 1460
    - types of, 1372
  - signal actions, 1371, 1393
  - sleeping, 1459, 1468, 1470
  - stack trace, 1371, 1393
  - stopping temporarily, 1371, 1396 to 1398
  - structures for, 1369, 1390
  - system
    - described, 1372
    - priority of, 1372, 1373, 1458, 1460, 1462, 1464
    - real-time processes and, 1458
    - scheduling policies for, 1460, 1462
    - timesharing processes and, 1458
  - terminology, 1369 to 1370
  - timesharing, *See* timesharing processes
  - timing, 1371, 1393
  - tool commands, 1370, 1393 to 1398
  - tracing flags, 1371, 1393
  - trees, 1371, 1393, 1395, 1396
  - troubleshooting, 1256, 1374
  - UNIX, priority of, 1372
  - user, *See* user processes
  - waiting for, 1371, 1396, 1468
- processing printing, 887
- PROCFS (process file system), 1370
- `procs` fields (`vmstat` report), 1379, 1406 to 1407
- `proc-sz` field (`sar` command), 1433
- `profil` command, 1383
- `.profile` file, 1158
- programs
  - analyzing CPU usage, 1383
  - disk-dependency of, 1417
  - forcing to quit running, 1246
  - interrupting, 1246
- protection page faults, 1428
- `prtconf` command, 1274, 1275 to 1276
- `prun` command, 1371, 1396 to 1398
- `ps` command, 1390 to 1392
  - -c option (scheduling class), 1374, 1390
  - -ecl option (global priority), 1400
  - -ef option (full information), 1372, 1391, 1392
  - -elf option (page daemon cycles), 1423
  - fields in reports from, 1374, 1390 to 1391
  - overview, 1370, 1387, 1390
- `PS` printer type, 874
- pseudo-ttys
  - changing number of, 1277, 1281
  - number configured, 1455
- `psig` command, 1371, 1393
- `PSR` printer type, 874
- `pstack` command, 1371, 1393
- `pstop` command, 1371, 1396 to 1398
- `pswch/s` field (`sar` command), 1435
- `pt_cnt` parameter, 1281, 1455, 1456
- `ptacct`*n.MMDD* file, 1334
- `ptime` command, 1371, 1393
- `ptree` command, 1371, 1393, 1395, 1396
- public directories, 1163
- `put` command

copying to remote systems, 1090 to 1092
described, 1085
example, 1092
pwait command, 1371, 1393, 1396
pwdx command, 1371, 1393, 1395, 1396
pwrit/s field (sar command), 1418

Q

question mark (?) wildcard character, 1196
queue
displaying at queue, 1358
displaying information about, 1386, 1414, 1421, 1429 to 1430
quitting
See also killing; stopping
at command, 1269
crontab command without saving changes, 1347
forcing programs to quit, 1246
quot command, 1304 to 1305
quota command, 1314, 1317, 1321, 1322, 1325
quota structures, number of, 1377
quotacheck command, 1314, 1315, 1319
quotaoff command, 1323
quotaon command, 1314, 1320
quotas, 1313 to 1323
administering, 1323
automatic turning on of, 1247
changing, 1323
changing for individual users, 1325 to 1326
checking, 1321
checking for exceeded user quotas, 1321 to 1322
checking on file systems, 1322 to 1323
configuring file systems for, 1316
consistency checking, 1319
disabling for individual users, 1327 to 1328
displaying information on, 1314, 1321, 1322
initializing, 1315, 1319
overview, 1314
prototype for multiple users, 1318
removing, 1323
requirements, 1314
setting up, 1314 to 1320
soft limit time
changing, 1324 to 1325
exceeding, 1323
turning off, 1315, 1323, 1328
turning on, 1315, 1320
user
changing for individual users, 1325 to 1326
checking for exceeded, 1321 to 1322
disabling for individual users, 1327 to 1328
setting up, 1317 to 1318
soft limit time, 1323 to 1325
verifying, 1314, 1317, 1321, 1325
quotas file, 1247, 1316

R

r field (vmstat report), 1379, 1406
r permissions
directories, 1161
files, 1160
r+w/s field (sar command), 1421
r/s field (iostat report), 1382, 1414
random-access memory (RAM), *See* memory
rawch/s field (sar command), 1436
%rcache field (sar command), 1418
rchar/s field (sar command), 1420
rcp command, 1093 to 1100
authentication, 1175
compared to ftp, 1084
copying between local and remote systems, 1096 to 1100
copying directories, 1096

  - described, 1093
  - examples, 1097 to 1100
  - pathnames
    - absolute vs. abbreviated, 1094 to 1095
    - syntax options, 1095
  - security issues, 1093
  - specifying source and target, 1094 to 1095
- `rcvin/s` field (`sar` command), 1436, 1437
- `rdate` command, 1277, 1278
- `re` field (`vmstat` report), 1379, 1407
- read permissions
  - directories, 1161
  - files, 1160
  - symbolic mode, 1205
- `read` system call
  - buffer cache for, *See* buffer cache
  - statistics on, 1258, 1262, 1419 to 1420
- `readlink` field (`nfsstat` report), 1448
- reads, 1382, 1414, 1417 to 1421
- real-time parameter table, 1469 to 1470
  - default, 1469
  - fields, 1470
  - overview, 1461, 1466
  - sample, 1469
  - scheduler configuration changes and, 1461
- real-time processes
  - *See also* processes; scheduling classes
  - changing class of, 1403
  - described, 1372
  - overview, 1372, 1373
  - priority of
    - changing, 1460, 1463
    - global, 1462, 1470
    - initial, 1463
    - maximum, 1465
    - overview, 1372, 1373, 1458, 1460
    - parameters controlling, 1470
    - specifying relative, 1401
  - scheduling policies for, 1460, 1462
  - system processes and, 1458
  - timesharing processes and, 1458, 1460
- rebooting
  - fails after crash, 1245 to 1246
  - monitoring, 1249, 1251, 1253, 1333
- `reboots` file, 1264, 1333
- receiver hardware interrupts, 1436, 1437
- `Ref` field (`netstat` report), 1445
- registering listen service, 926
- `reject` command, 886
- rejecting, 951
- rejecting print requests, 886, 931
- reliability testing, *See* testing
- reminders, 1266
- remote copying
  - *See also* copying
  - using `ftp`, 1085 to 1093
  - using `rcp`, 1093 to 1100
- remote `ftp` server, 1085
- remote logins
  - authentication, 1175
  - authentication (`ftp`), 1084
  - authentication (`rlogin`), 1073 to 1076
    - `/etc/hosts.equiv` file, 1074 to 1075
    - network authentication vs. remote system authentication, 1073 to 1074
    - `.rhosts` files, 1075 to 1076
  - authorization, 1175
  - closing `ftp` connection, 1087
  - direct vs. indirect (`rlogin`), 1076 to 1078
  - domains, 1072
  - finding who is logged in, 1081 to 1082
  - `ftp` commands, 1085
  - interrupting, 1072
  - linking logins, 1076
  - `nuucp` login account, 1167
  - opening `ftp` connection, 1086 to 1087
  - removing `.rhosts` files, 1080
  - using `rlogin`, 1082 to 1083

verifying remote system operation, 1080
remote printing
diagram of process, 906
user fee calculation for, 1250, 1338
remote systems, 1071 to 1086
communication methods, 1084
defined, 1072
logging in, 1073 to 1087
authentication, 1175
authorization, 1175
nuucp login account, 1167
logging out (exit), 1083 to 1084
remote copying
using ftp, 1085 to 1093
using rcp, 1093 to 1100
verifying operation, 1080
removing form paper, 983
removing, *See* deleting
repairing corrupted files, *See* files, fixing corrupted
repetitive system events, 1266 to 1269, 1344 to 1355
reporting
See also accounting, reports; displaying
automatic, 1384 to 1385
reports
ASET, 1185, 1186
reports (ASET), 1183, 1185, 1186, 1192
reports directory, 1185
repquota command, 1321, 1322 to 1323
request ID, 879, 950
canceling print requests by, 884, 954, 955
changing priority of print requests using, 959
components of, 885
in status of print request, 950
moving print requests by, 957
requests log, 895, 903
reset command, 1085
residentfonts file, 996
resources, *See* system resources
restarting
lpd daemon, 1050
print scheduler, 883
processes, 1371, 1396 to 1398
runacct command, 1332 to 1334, 1341
restoring (ASET), 1191
restoring file systems, 1250, 1256, 1332
restricted shell (rsh), 1171
retrans field (nfsstat report), 1447
return settings, 1034
Reverse PostScript printer
file content type for, 877
method of printing, 874
printer type for, 874, 875, 877
rgn cache name, 1411
.rhosts files
deleting, 1080
described, 1075
linking remote logins, 1076
remote system authentication process, 1074 to 1075
searching, 1080
security issues, 1075 to 1076
rlogin command
authentication, 1073 to 1076, 1175
/etc/hosts.equiv file, 1074 to 1075
network vs. remote system authentication, 1073 to 1074
.rhosts files, 1075 to 1076
compared to ftp, 1084
described, 1072
direct vs. indirect logins, 1076 to 1078
interrupting logins, 1072
process after logging in, 1078 to 1079
using, 1082 to 1083
rm command, 1076, 1308, 1309 to 1310
root access
displaying attempts on console, 1172, 1230

monitoring su command use, 1172, 1229
restricting, 1172, 1176, 1229
root file, 1250, 1267, 1268, 1332, 1337
root login
account
access restrictions, 1172
described, 1167
restricting to console, 1229
tracking, 1158
RPC, 1446, 1447
rprt.*MMDD* file, 1252, 1264, 1335
rpt.*MMDD* file, 1334
rq mount option, 1316, 1319
RS-232-C, *See* serial port
rsh program, 1171
RT processes, *See* real-time processes
RT_DPTBL loadable module, 1461, 1465, 1466, 1469
rt_dptbl parameter table, 1461, 1466, 1469 to 1470
rt_glbpri field (rt_dptbl table), 1470
rt_maxpri parameter, 1465
rt_qntm field (rt_dptbl table), 1470
runacct command, 1332 to 1335
caution, 1334
crontab entry that runs, 1332, 1336
diagnostics file, 1332
error messages, 1341
error protection, 1332 to 1333
failure of, 1341
files used / produced by, 1263 to 1264, 1335
See also specific files
fixing corrupted files, 1265, 1333, 1339
last time executed, 1264
monacct command and, 1332
overview, 1248, 1252 to 1333
prdaily command and, 1332, 1335
progress file, 1332
restarting, 1332 to 1334, 1341
running after midnight, 1334
scheduling running of, 1332
serial use of, 1264
states of, 1333 to 1334
user fee calculation and, 1250, 1256
runacct.local command, 1334
runaway processes, 1374
running, *See* starting
%runocc field (sar command), 1430
runq-sz/s field (sar command), 1430
rusers command, 1081 to 1082

S

S22acct file, 1332
/sa directory, 1384
sa filename prefix, 1384, 1385, 1438
sa1 command, 1384
sa2 command, 1384, 1385
SAC, *See* Service Access Controller
sacadm command
(explained), 1131
adding a listen port monitor with, 1144
adding a ttymon port monitor with, 1137
disabling a listen port monitor with, 1146
disabling a ttymon port monitor with, 1139
enabling a listen port monitor with, 1146
enabling a ttymon port monitor with, 1139
killing a listen port monitor with, 1145
killing a ttymon port monitor with, 1138
listing a listen port monitor with, 1145
listing a ttymon port monitor with, 1138
removing a listen port monitor with, 1147

removing a ttymon port monitor with, 1140
starting a listen port monitor with, 1146
starting a ttymon port monitor with, 1138

sadc command, 1384, 1385, 1387, 1438

sadcnt parameter, 1452

sa*dd* file, 1384, 1385, 1438

SAF, *See* Service Access Facility

sar command, 1385 to 1386, 1416 to 1437
-A option (overall performance), 1386, 1437
-a option (file access), 1386, 1416 to 1417
-b option (buffers), 1386, 1417 to 1419
-c option (system calls), 1386, 1419 to 1420
-d option (block devices), 1386, 1421
-e option (ending time), 1385
-f option (file to extract data from), 1385
-g option (page-out / memory freeing), 1386, 1423
-i option (interval), 1385
-k option (kernel memory), 1386, 1424 to 1426
-m option (interprocess communication), 1386, 1426
options listed, 1385 to 1386
overview, 1375, 1378, 1385, 1387
-p option (page-in / page faults), 1386, 1427 to 1429
-q option (queue), 1386, 1429 to 1430
-r option (unused memory), 1386, 1431
-s option (starting time), 1385
-u option (CPU usage), 1386, 1432 to 1433
-v option (system tables), 1386, 1433 to 1434
-w option (swapping / switching volume), 1386, 1435 to 1436
-y option (terminal activity), 1386

saving
crash dump information, 1244, 1284 to 1286
failed login attempts, 1166, 1226

scall/s field (sar command), 1420

/sched directory, 1460, 1461, 1466

Schedule undefined for periodic invocation message, 1240

scheduler, 1457 to 1470
See also scheduling class tables; scheduling classes
configuration of, 1461 to 1470
global priorities and, 1458, 1462 to 1463
overview, 1457 to 1458
scheduling classes and, 1457, 1459 to 1460
time slices used by, 1459, 1466, 1467
tunable parameters, 1463 to 1465

scheduler, *See* print scheduler

scheduling ASET execution (PERIODIC_ SCHEDULE), 1178, 1189, 1194 to 1195, 1234

scheduling class tables, 1466 to 1470
maxclsyspri parameter and, 1464
overview, 1466
rt_dptbl parameter table, 1461, 1466, 1469 to 1470
rt_maxpri parameter and, 1465
ts_dptbl parameter table, 1460, 1461, 1466 to 1468
ts_kmdpris table, 1466, 1470

scheduling classes, 1372 to 1373, 1399 to 1404
See also interactive processes; real-time processes; system processes; timesharing processes
changing, 1403, 1458
changing priority of, 1373 to 1374, 1401 to 1402
defaults, 1401, 1459
described, 1372 to 1373
designating, 1401

  - displaying information on, 1390, 1399 to 1400
  - priority levels and, 1372 to 1373, 1399, 1401 to 1402, 1458
  - scheduling policies and, 1372 to 1373, 1459 to 1460
  - types of, 1372 to 1373
- scheduling one-time system events, 1269 to 1270, 1344, 1356 to 1363
- scheduling repetitive system events, 1266 to 1269, 1344 to 1355
- SCSI disk controllers *See* disk controllers
- SCSI disk drives, 1422
- `sd` fields (`iostat` report), 1381, 1382, 1412
- searching
  - files with `setuid` permissions, 1210 to 1211
  - `.rhosts` files, 1080
  - users logged in to remote system, 1081 to 1082
- searching, *See* finding
- Secure RPC authentication, 1175
- security
  - *See also* passwords
  - `at` command, 1270
  - copy operation issues, 1093
  - `crontab` command, 1268 to 1269
  - `/etc/hosts.equiv` file issues, 1075
  - `.rhosts` file issues, 1075 to 1076, 1080
- `seg` cache name, 1411
- selectable character sets, 962 to 963, 970 to 972
- `sema/s` field (`sar` command), 1426
- semaphores, *See* interprocess communication
- `seminfo_`*xxxxxx* parameters, 1282, 1453, 1454
- serial port
  - adapter board, 1105
  - defined, 1104
- Serial Port Manager
  - description of, 1107
  - Modify window field descriptions, 1111
  - prerequisites for running, 1115
  - starting, 1115
  - when to use, 1107
- Serial Port Manger, 1106
- serial printer, adding to system, 872
- `serv` field (`iostat` report), 1382, 1412
- servers
  - displaying information about, 1440, 1446 to 1448
  - scheduling policies and, 1460
  - tracing client calls to, 1440, 1442
- `Service 0` listen service, 926
- Service Access Controller, 1131
- Service Access Facility
  - description, 1106
  - files associated with (table), 1150
  - overview of, 1108, 1130
  - programs associated with (table), 1130
  - services controlled by
    - states of (table), 1151
  - uses for, 1108, 1130
  - when to use, 1107
- service times, 1382, 1414, 1421
- `setenv` command
  - ASET security level specification, 1194
  - ASET working directory specification, 1193
- `setfacl` command
  - adding ACL entries, 1218
  - deleting ACL entries, 1219
  - described, 1165
  - examples, 1216, 1218, 1219
  - modifying ACL entries, 1218
  - setting ACL entries, 1215 to 1216
  - syntax, 1215
- `setgid` permissions
  - absolute mode, 1208 to 1210
  - described, 1162 to 1163
  - symbolic mode, 1205

set-group identification permissions, *See* setgid permissions
setting
  alerts to mount font cartridges, 968
  alerts to mount print wheels, 968
  alias for selectable character set, 970 to 972
setuid permissions
  absolute mode, 1208 to 1210
  described, 1162
  finding files with permissions set, 1210 to 1211
  security risks, 1162
  symbolic mode, 1205
setuid programs, 1158
SETUP state (runacct command), 1333
set-user identification permissions, *See* setuid permissions
sh program, 1170 to 1171
share command, 1176
shared memory
  increasing number of segments, 1277, 1282
  interprocess communication parameters, 1453, 1454
  process virtual memory, 1370
shared memory record table, 1433 to 1434
sharing files (network security), 1176
shell, 1258
shell programs
  See also specific shell programs
  ASET security level specification, 1194
  ASET working directory specification, 1193
  /etc/d_passwd file entries, 1170 to 1171
shminfo_*xxxxxx* parameters, 1282, 1453, 1454
showing, *See* displaying
shutacct command, 1251, 1252
shutdown command, 1252
shutdowns, 1251, 1252, 1253
si field (vmstat report), 1380, 1410
signal actions, 1371, 1393
simple file content type, 876
size
  directory, 1303 to 1305
  file, 1299 to 1300, 1303 to 1305
  free list, 1379
  system table, 1433 to 1434
slash, *See* backslash (\)
sleeping processes, 1459, 1468, 1470
slices, *See* disk slice tables; disk slices
slock/s field (sar command), 1428
slow print filters, 994
slowdowns, *See* disk drives, slowdowns
SMD disk controllers *See* disk controllers
sml_mem field (sar command), 1425
snoop command, 1440, 1442
so field (vmstat report), 1380, 1410
soft limit time
  changing, 1324 to 1325
  exceeding, 1323
software locks, 1428
Solaris environment, 867
Solstice AdminSuite, 1176
Solstice AdminSuite software, 913, 1176
Spacct*n.MMDD* file, 1263, 1333
space (disk)
  amount of free, 1296 to 1298, 1383, 1415 to 1416
  optimizing usage, 1295 to 1312
/spool directory, *See* /var/spool directory
spooling directory
  creating for a printer, 1055
  print service, 897
spooling space required for print server, 868
spray command, 1440, 1441
sr field (vmstat report), 1379, 1407
sread/s field (sar command), 1420
srtt field (nfsstat report), 1448
stack trace, 1371, 1393

standard interface script, 922
standard printer interface program, 1010
starting
- Admintool, 915
- ASET
  - initiating sessions from shell, 1177
  - running interactively, 1232 to 1233
- lpd daemon, 1050
- print scheduler, 936
- *See* restarting

startup command, 1251
statefile file, 1264, 1333, 1341
states, (runacct command), 1333 to 1334
statistics, *See* displaying
status
- of print requests, 949
- of printer, 933, 949

status reporting, *See* displaying
statvfs structure, 1296
sticky bit permissions
- absolute mode, 1208 to 1210
- described, 1163
- symbolic mode, 1205

STIME field (ps report), 1374, 1390
stopping
- *See also* killing; quitting
- dial-up logins temporarily, 1229
- ftp command interpreter, 1085
- print scheduler, 936
- printer, 887
- processes temporarily, 1371, 1396 to 1398

strctlsz parameter, 1452
STREAMS
- KMA resources and, 1424 to 1425
- parameters, 1452

strmsgsz parameter, 1452
strthresh parameter, 1452
structures
- process, 1369, 1390
- quota, 1377
- statvfs, 1296

stty option, 1001
stty settings
- customizing, 1007
- defaults, 1000, 1063
- suggestions for, 1064
- troubleshooting, 1033

su command
- displaying use on console, 1172, 1230
- monitoring use, 1172, 1229

su file
- displaying su command use on console, 1172, 1230
- monitoring su command, 1229

subdirectories, *See* directories
sulog file, 1172, 1229
/sum directory, *See* /var/adm/acct/sum directory
SunOS 5.x, 867
SunOS operating systems, 867
SunSoft print client software, 866, 869, 913
svc_t field (iostat report), 1382, 1414
SVR4 LP print spooler, 921
SVR4 printer interface script, 921
swap areas
- adding, 1376
- defined, 1376
- displaying information on, 1379, 1387, 1407, 1431
- unused disk blocks, 1431

swap command, 1387
swap field (vmstat report), 1379, 1407
swap space (print server), 869
swapping
- *See also* paging
- defined, 1375
- displaying information on, 1380, 1386, 1410, 1429 to 1431, 1435 to 1436
- LWPs, 1375, 1429, 1430, 1435
- minimum memory to avoid deadlock, 1454
- priority of, 1372

reducing, 1375
switching
See also context switches
displaying information on, 1386, 1435 to 1436
swpin/s field (sar command), 1435
%swpocc field (sar command), 1430
swpot/s field (sar command), 1435
swpq-sz field (sar command), 1430
swrit/s field (sar command), 1420
sy field
iostat report, 1382, 1413
vmstat report, 1380, 1407
symbolic links
file permissions, 1160
file type, 1161
latest directory (ASET), 1185
symbolic mode
changing file permissions, 1205, 1207 to 1208
described, 1204
synchronizing command-reply sequencing with remote ftp server, 1085
synchronizing date and time with another system, 1278
syntax (crontab files), 1344 to 1345
%sys field (sar command), 1432
sys file, 1268, 1384, 1438
sys login account, 1167
SYS processes, *See* system processes
sys_name parameter, 1464
sysadmin group, 915
sysconf.rpt file, 1181, 1185
sysdef command, 1450
setup file, 1244, 1286
syslog.conf file, 1291 to 1292
syslogd daemon, 1244
system activities
See also system events
automatic collection of data on, 1384, 1438
list of activities tracked, 1378
manual collection of data on, 1385 to 1386, 1416 to 1437
system authentication for remote logins, 1073 to 1074, 1078
system calls
See also read system call; write system call
displaying information on, 1379, 1380, 1386, 1407, 1419 to 1420
system crashes, *See* crashes
system events
See also system activities
displaying information on, 1380, 1409
scheduling
one-time events, 1269 to 1270, 1344, 1356 to 1363
repetitive events, 1266 to 1269, 1344 to 1355
system exit status, 1261
system file, *See* /etc/system file
system logging (customizing), 1291 to 1292
system messages
See also error messages
specifying storage location for, 1244
system mode (CPU), 1380, 1382, 1407, 1413, 1432
system performance, *See* performance
system processes
See also processes
described, 1372
priority of, 1372, 1373, 1458, 1460, 1462, 1464
real-time processes and, 1458
scheduling policies for, 1460, 1462
timesharing processes and, 1458
system resources
allocating for print server, 868
critical, 1470
monitoring, 1243 to 1270
accounting, 1331 to 1341
accounting system for, 1248 to 1265

automatic, 1266 to 1270
crashes, 1244 to 1247, 1283 to 1292
quotas, 1313 to 1323
overview, 1368 to 1376
system security, 1165 to 1172, 1223 to 1230
dial-up passwords, 1168 to 1171, 1227
basic sequence, 1170
creating, 1227
disabling dial-up logins temporarily, 1229
/etc/d_passwd file, 1169, 1170 to 1171
/etc/dialups file, 1169
displaying
user's login status, 1224 to 1225
users with no passwords, 1225
login access restrictions, 1165 to 1166
overview, 1155, 1165
passwords, 1167
restricted shell, 1171
restricting root login to console, 1172, 1229
root access restrictions, 1172, 1176, 1229
saving failed login attempts, 1166, 1226
special logins, 1166 to 1167
su command monitoring, 1172, 1229
system tables
checking status, 1386, 1433 to 1434
space allocation for, 1377
system usage, *See* accounting
SZ field (ps report), 1390

T

tab settings, 1034
tables, *See* disk slice tables; scheduling class tables; system tables
tacct file, 1265, 1334, 1335, 1340
tacct.*MMDD* file, 1265, 1334, 1340
tacct.prv file, 1340
tacct*n* file, 1265
tacctprev file, 1265
TASKS variable
configuring ASET, 1188, 1195
described, 1193
error messages, 1239, 1240
taskstat utility (ASET), 1179, 1183
TCP protocol, 1444
TCP/IP traffic, 1440, 1443 to 1444
technical support, 1245, 1288
templates (print filters), 1019
temporary directories, 1306, 1309 to 1310
terminal characteristics, 1007
terminals
displaying I/O information for, 1381, 1386, 1412, 1436 to 1437
line usage monitoring, 1249, 1253 to 1254, 1264, 1333, 1335
menu items in Serial Port Manager, 1112
overview of Serial Port Manager, 1110
process controlling, 1391
setting up, 1117
tools for managing, 1106
troubleshooting bad lines, 1254, 1335
troubleshooting tips for, 1127
terminfo database
character set names, 963
entries in, 873
terminfo entry
adding, 1002
for unsupported printer, 1005
list of required items, 1002
selectable character sets in, 962
testing packet reliability, 1440
threads, *See* application threads; kernel thread
tilde (~)
abbreviated pathnames, 1094 to 1095
rcp command syntax, 1096 to 1100
time
changing, 1277, 1279

CPU usage, 1255, 1374, 1379, 1380, 1382, 1383, 1386, 1391, 1407, 1412, 1432 to 1433
disk slowdowns, 1375, 1432
displaying, 1274, 1276
process timing, 1371, 1393
processes accumulating large amounts of CPU time, 1374
service, 1382, 1414, 1421
soft limit, 1323 to 1325
synchronizing with another system, 1277, 1278

`TIME` field (`ps` report), 1374, 1391

time fields (`crontab` file), 1344

time slices, 1459, 1466, 1467

`timeout` field (`nfsstat` report), 1447

`timers` field (`nfsstat` report), 1448

timesharing parameter table, 1466 to 1468
default, 1466
fields, 1467
overview, 1460, 1466
sample, 1467
scheduler configuration changes and, 1461

timesharing processes
See also processes; scheduling classes
changing scheduling parameters, 1402
described, 1372
overview, 1372, 1373
priority of
changing, 1373 to 1374, 1402
global, 1462, 1468
overview, 1372, 1373, 1399, 1458
parameters controlling, 1468
range of, 1399, 1464
real-time processes and, 1458, 1460
scheduling policies for, 1372, 1373, 1459 to 1460, 1462
sleeping, 1468, 1470
system processes and, 1458

`tin` field (`iostat` report), 1381, 1412

`/tmp` directory, clearing out, 1306, 1309 to 1310

`/tmp` file system, 1298

`/tmp/disktacct.`*MMDD* file, 1334

`tmpfs` file system, 1163

`tmpwtmp` file, 1264, 1333, 1335

tools
process, 1370, 1393 to 1398
system performance monitoring, 1378 to 1386, 1387

total command summary, 1253, 1259, 1264, 1265

`TOTAL DURATION` column (daily report), 1254

`TOTAL KCOREMIN` column (daily command summary), 1258

`tout` field (`iostat` report), 1381, 1412

`tps` field (`iostat` report), 1382, 1412

tracing flags, 1371, 1393

TranScript filters, 974

translation page faults, 1428

transmitter hardware interrupts, 1436, 1437

traps, 1380, 1407

trees, 1371, 1393, 1395, 1396

`troff` filter to convert to PostScript, 1024

Trojan horse, 1157 to 1158

troubleshooting
See also printer(s), troubleshooting
networks, 1446, 1448
printing problems, 1029 to 1068
processes, 1256, 1374
`tty` lines, 1254, 1335

trusted hosts, 1174

trusting network environment
remote login
authentication process, 1074
process after logging in, 1078 to 1079

`TS` processes, *See* timesharing processes

`TS_DPTBL` loadable module, 1460, 1461, 1464, 1466, 1470

ts_dptbl parameter table, 1460, 1461, 1466 to 1468
ts_kmdpris parameter table, 1466, 1470
ts_lwait field (ts_dptbl table), 1468
ts_maxupri parameter, 1464
ts_maxwait field (ts_dptbl table), 1468
ts_quantum field (ts_dptbl table), 1467
ts_slpret field (ts_dptbl table), 1468
ts_tqexp field (ts_dptbl table), 1468
TTY field (ps report), 1391
tty fields (iostat report), 1381, 1412
tty lines
  displaying I/O information for, 1381, 1386, 1412, 1436 to 1437
  troubleshooting bad lines, 1254, 1335
  usage monitoring, 1249, 1253 to 1254, 1264, 1333, 1335
ttyadm command, 1135
ttymon command, 1134, 1249
ttymon port monitor
  (figure), 1133
  adding, 1137
  bidirectional modem service and, 1134
  disabling, 1139
  enabling, 1139
  killing, 1138
  listing, 1138
  overview of function of, 1132
  removing, 1140
  starting, 1138
ttymon service
  adding, 1141
  disabling, 1144
  enabling, 1143
  listing, 1142
ttys (pseudo)
  changing number of, 1277, 1281
  number configured, 1455
tune files (ASET)
  described, 1186, 1190
  example files, 1196 to 1198
  format, 1196
  modifying, 1190
  rules, 1196 to 1198
tune.rpt file, 1179, 1185
tune_t_flckrec parameter, 1282, 1454
tune_t_fsflushr parameter, 1454
tune_t_gpgslo parameter, 1455
tune_t_minarmem parameter, 1454
tune_t_minasmem parameter, 1454
tuning
  daily command summary and, 1256
  kernel parameters, 1454 to 1456, 1463 to 1465
turnacct switch command, 1251, 1252, 1333
turning off
  See also disabling
  quotas, 1315, 1323, 1328
turning on
  See also enabling
  quotas, 1315, 1320

U

UDP protocol, 1444
UFS file systems
  displaying information about
    space usage, 1298
    user allocation, 1304 to 1305
UFS parameters, 1377, 1452
%ufs_ipf field (sar command), 1424
ufs_ninode parameter, 1377, 1452
UID column (daily usage report), 1255
UID field (ps report), 1390
uid_aliases file
  described, 1186
  specifying, 1190
UID_ALIASES variable
  aliases file specification, 1190, 1195
  described, 1187, 1193
umask, 1163 to 1164
UMASK environment variable, 1181

- `uname` command, 1274 to 1275
- universal address, 926
- UNIX processes (priority), 1372
- UNIX systems (crash information), 1244
- unmounting
  - font cartridges, 966
  - forms, 983
  - print wheels, 966
- unsupported printers, 877, 1002
- `us` field
  - `iostat` report, 1382, 1412
  - `vmstat` report, 1380, 1407
- `Usage: aset` message, 1237
- `Use` field (`netstat` report), 1445
- `used` field (`df` report), 1383, 1415
- user
  - access to printer
    - ability to limit with Admintool, 871
    - limiting with Admintool, 880
    - limiting with `lpadmin` command, 880
  - canceling print requests for, 956
- user accounts
  - ASET check, 1180
  - displaying login status, 1224 to 1225
- `user` ACL entries
  - default entries for directories, 1214
  - described, 1213
  - setting, 1215 to 1216
- user allocation, 1304 to 1305
- user classes of files, 1160
- `user` facility, 1292
- user fees, 1250, 1251, 1256, 1332, 1338
- `%user` field (`sar` command), 1432
- user logins
  - last login monitoring, 1253, 1260, 1334, 1335
  - number of logins, 1256
  - time monitoring, 1249, 1255, 1333
- user mode (CPU), 1380, 1382, 1407, 1412, 1432
- user names
  - current user, 1095
  - direct vs. indirect logins (`rlogin`), 1076 to 1078
  - finding users logged in to remote system, 1081 to 1082
- user processes
  - changing priority, 1373 to 1374
  - CPU usage by, 1255
  - in kernel mode, 1460
  - maximum per user-id, 1377
  - number per user, 1277, 1281
  - priority of, 1372, 1373, 1399
- user quotas
  - changing for individual users, 1325 to 1326
  - checking for exceeded, 1321 to 1322
  - disabling for individual users, 1327 to 1328
  - setting up, 1317 to 1318
  - soft limit time, 1323
- `user` structure, 1369
- `USEREXIT` state (`runacct` command), 1334
- user-mode priority, 1399
- users (maximum number), 1377
- `usr` cache name, 1411
- `/usr/adm` directory, 1244 to 1245
- `/usr/adm/acct` directory, 1248
- `/usr/adm/messages` file, 1245, 1288, 1292
- `/usr/adm/messages.`*n* file, 1288
- `/usr/adm/sa` directory, 1384
- `/usr/aset` directory, 1177
- `/usr/aset/asetenv` file
  - described, 1187
  - error messages, 1239, 1240
  - modifying, 1188
  - running ASET periodically, 1234
- `/usr/aset/masters/tune` files
  - described, 1186, 1190
  - example files, 1196 to 1198
  - format, 1196
  - modifying, 1190

rules, 1196 to 1198
/usr/aset/reports directory
structure, 1183 to 1185
/usr/aset/reports/latest
directory, 1185
/usr/bin/nice command, *See* nice command
/usr/lib/acct directory, 1248
/usr/lib/acct/ckpacct command, *See* ckpacct command
/usr/lib/acct/dodisk command, *See* dodisk command
/usr/lib/acct/prdaily command, *See* prdaily command
/usr/lib/acct/runacct command, *See* runacct command
/usr/lib/acct/runacct.local
command, 1334
/usr/lib/acct/startup
command, 1251
/usr/lib/lp directory, 974
/usr/lib/lp/model directory, 902
/usr/lib/lp/postscript
directory, 901, 973
/usr/lib/sa/sa1 command, 1384
/usr/lib/sa/sa2 command, 1384, 1385
/usr/lib/sa/sadc command, 1384, 1385, 1387, 1438
/usr/proc/bin directory, 1370, 1393 to 1398
/usr/sbin/crash utility, 1288, 1290 to 1291
/usr/spool/cron/atjobs
directory, 1267, 1270, 1344
/usr/spool/cron/crontab directory, *See* /var/spool/cront/crontab directory
usrgrp.rpt file
described, 1180, 1185
example, 1185
format, 1185
utmp2wtmp command, 1333
uucico program, 1170 to 1171
uucp login account, 1167
UUCP program, 1167

V

validity faults, 1428
/var partition, 868
/var/adm directory
controlling size of, 1309
described, 1263
controlling size of, 1244 to 1245
raw accounting data in, 1251
/var/adm/acct directory, 1263
/var/adm/acct/fiscal
directory, 1265, 1263
/var/adm/acct/nite directory, 1263 to 1264
/var/adm/acct/nite/active
file, 1263, 1332, 1341
/var/adm/acct/nite/active.*MMDD*
file, 1263, 1332
/var/adm/acct/nite/cms file, 1263, 1334
/var/adm/acct/nite/ctacct.*MMDD*
file, 1263, 1333
/var/adm/acct/nite/ctmp file, 1263
/var/adm/acct/nite/daycms
file, 1263, 1334, 1335
/var/adm/acct/nite/daytacct
file, 1256, 1263, 1333, 1334, 1335
/var/adm/acct/nite/disktacct
file, 1250, 1251, 1252, 1263, 1334
/var/adm/acct/nite/disktacct.*MMDD* file, 1334
/var/adm/acct/nite/fd2log
file, 1263, 1332, 1341
/var/adm/acct/nite/lastdate
file, 1264, 1332
/var/adm/acct/nite/lineuse
file, 1264, 1333, 1335
/var/adm/acct/nite/lock file, 1264, 1332, 1341

/var/adm/acct/nite/lockl file, 1332
/var/adm/acct/nite/log file, 1264
/var/adm/acct/nite/log.*MMDD* file, 1264
/var/adm/acct/nite/owtmp file, 1264
/var/adm/acct/nite/reboots file, 1264, 1333
/var/adm/acct/nite/statefile file, 1264, 1333, 1341
/var/adm/acct/nite/tmpwtmp file, 1264, 1333, 1335
/var/adm/acct/nite/wtmp file, *See* wtmp file
/var/adm/acct/nite/wtmp.*MMDD* file, 1264, 1333, 1339
/var/adm/acct/nite/wtmperror file, 1264
/var/adm/acct/nite/wtmperror.*MMDD* file, 1264
/var/adm/acct/nite/xwtmp file, 1339
/var/adm/acct/sum directory, 1252, 1263, 1264
/var/adm/acct/sum/cms file, 1264, 1334, 1335
/var/adm/acct/sum/cmsprev file, 1264
/var/adm/acct/sum/daycms file, 1264, 1334, 1335
/var/adm/acct/sum/loginlog file, 1264, 1334, 1335
/var/adm/acct/sum/rprt.*MMDD* file, 1252, 1264, 1335
/var/adm/acct/sum/rpt.*MMDD* file, 1334
/var/adm/acct/sum/tacct file, 1265, 1334, 1335, 1340
/var/adm/acct/sum/tacct.*MMDD* file, 1265, 1334, 1340
/var/adm/acct/sum/tacct.prv file, 1340
/var/adm/acct/sum/tacct*n* file, 1265
/var/adm/acct/sum/tacctprev file, 1265
/var/adm/acct/sum/xtacct file, 1340
/var/adm/dtmp file, 1263
/var/adm/fee file, 1251, 1263, 1334, 1338
/var/adm/loginlog file
- overview, 1166
- saving failed login attempts, 1166, 1226

/var/adm/messages file, 1245, 1288, 1292
/var/adm/messages.*n* file, 1288
/var/adm/pacct file, *See* pacct*n* file
/var/adm/pacct*n* file, *See* pacct*n* file
/var/adm/sa/sa*dd* file, 1385, 1438, 1384
/var/adm/Spacct*n*.*MMDD* file, 1263, 1333
/var/adm/wtmp file, *See* wtmp file
/var/crash directory, 1244, 1284 to 1285, 1311
/var/lp/logs directory, 903
/var/lp/logs/lpNet log file, 1032
/var/lp/logs/lpsched file, 902, 1032
/var/spool directory, 1306, 1309 to 1310
/var/spool/cron/atjobs directory, 1267, 1270
/var/spool/cron/crontab directory, 1266, 1268
/var/spool/cron/crontabs/adm file, 1268
/var/spool/cron/crontabs/lp file, 903
/var/spool/cron/crontabs/root file, 1250, 1267, 1268
/var/spool/cron/crontabs/lp file, 1268
/var/spool/cron/crontabs/sys file, 1268, 1384, 1438
/var/spool/lp directory, 868, 897
/var/spool/lp/requests directory, 893

/var/spool/lp/tmp directory, 893
/var/tmp directory, 1306, 1309 to 1310
/var/tmp/deadfiles file, 1309
variables
  ASET environment variables, 1193 to 1196
    ASETDIR, 1193
    ASETSECLEVEL, 1194
    CKLISTPATH_*level*, 1187, 1189, 1196
    PERIODIC_SCHEDULE, 1189, 1194 to 1195, 1234
    summary table, 1193
    TASKS, 1188, 1195
    UID_ALIASES, 1187, 1190, 1195
    YPCHECK, 1190, 1195
verifying
  at jobs, 1357
  crontab file existence, 1347
  quotas, 1314, 1317, 1321, 1325
  remote system operation, 1080
vflt/s field (sar command), 1428
vfstab file, quotas and, 1316, 1319
viewing, *See* displaying
virtual caches, 1380, 1410 to 1411
virtual memory
  displaying information on, 1274, 1275 to 1276, 1379, 1387, 1406 to 1408
  process, 1370
viruses
  See also security
  Trojan horse, 1157 to 1158
vmstat command, 1379 to 1381, 1406 to 1408
  -c option (cache flushing), 1380, 1410 to 1411
  fields in reports from, 1379 to 1380, 1406 to 1407, 1410
  -i option (interrupts), 1381, 1411
  overview, 1379, 1387, 1406
  -S option (swapping), 1380, 1410
  -s option (system events), 1380, 1409

W

%w field (iostat report), 1382, 1414
w field (vmstat report), 1379, 1407
w permissions
  directories, 1161
  files, 1160
w/s field (iostat report), 1414
wait field
  iostat report, 1382, 1414
  nfsstat report, 1448
wait mode (CPU), 1382, 1413, 1432
waiting
  for disk transfer requests, 1421
  for processes, 1371, 1396, 1468
warning messages, *See* error messages
Warning! Duplicate ASET execution scheduled message, 1240
Watchdog reset ! message, 1288
%wcache field (sar command), 1418
WCHAN field (ps report), 1390
wchar/s field (sar command), 1420
weekly tasks, scheduling with crontab, 1266
wildcard characters, 1196
windows (frozen), 1246
%wio field (sar command), 1432
working directory
  defined, 1095
  ftp commands, 1085
Workstation Info command, 1272
Workstation Information window, 1272 to 1273
write permissions
  directories, 1161
  files, 1160
  symbolic mode, 1205
write system call
  buffer cache for, *See* buffer cache
  statistics on, 1258, 1262, 1419 to 1420
writes, 1382, 1414, 1417 to 1421
w/s field (iostat report), 1382

wt field (iostat report), 1382, 1413
wtmp file
  daily report and, 1253
  fixing corrupted, 1265, 1333, 1339
  overview, 1249, 1251, 1265, 1333
  shutdowns and, 1252
wtmp.*MMDD* file, 1264, 1333, 1339
wtmperror file, 1264
wtmperror.*MMDD* file, 1264
wtmpfix command, 1264, 1265, 1333
WTMPFIX state (runacct command), 1333

X

x permissions
  directories, 1161
  files, 1160
XD7053 disk controllers *See* disk controllers
xmtin/s field (sar command), 1436, 1437
xtacct file, 1340
xwtmp file, 1339
Xylogics disk controllers*See* disk controllers

Y

YPCHECK variable
  described, 1193
  specifying system configuration file tables, 1190, 1195

Des parties de ce produit pourront etre derivees du système UNIX®, licencié par UNIX Systems Laboratories Inc., filiale entierement detenue par Novell, Inc. ainsi que par le système 4.3. de Berkeley, licencié par l'Université de Californie. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fourmisseurs de Sun.

LEGENDE RELATIVE AUX DROITS RESTREINTS : l'utilisation, la duplication ou la divulgation par l'administation americaine sont soumises aux restrictions visées a l'alinéa (c)(1)(ii) de la clause relative aux droits des données techniques et aux logiciels informatiques du DFAR 252.227- 7013 et FAR 52.227-19.

Le produit décrit dans ce manuel peut Être protege par un ou plusieurs brevet(s) americain(s), etranger(s) ou par des demandes en cours d'enregistrement.

MARQUES
Sun, Sun Microsystems, le logo Sun, Solaris sont des marques deposées ou enregistrées par Sun Microsystems, Inc. aux Etats-Unis et dans certains autres pays. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays, et exclusivement licenciée par X / Open Company Ltd. OPEN LOOK est une marque enregistrée de Novell, Inc., PostScript et Display PostScript sont des marques d'Adobe Systems, Inc.

Toutes les marques SPARC sont des marques deposées ou enregitrées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. SPARCcenter, SPARCcluster, SPARCompiler, SPARCdesign, SPARC811, SPARCengine, SPARCprinter, SPARCserver, SPARstation, SPARCstorage, SPARCworks, microSPARC, microSPARC II et UltraSPARC sont exclusivement licenciées a Sun Microsystems, Inc. Les produits portant les marques sont basés sur une architecture développée par Sun Microsytems, Inc.

Les utilisateurs d'interfaces graphiques OPEN LOOK® et Sun™ ont été développés par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnait les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique, cette licence couvrant aussi les licencies de Sun qui mettent en place OPEN LOOK GUIs et qui en outre se conforment aux licences écrites de Sun.

Le système X Window est un produit du X Consortium, Inc.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A REPONDRE A UNE UTILISATION PARTICULIERE OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.

CETTE PUBLICATION PEUT CONTENIR DES MENTIONS TECHNIQUES ERRONEES OU DES ERREURS TYPOGRAPHIQUES. DES CHANGEMENTS SONT PERIODIQUEMENT APPORTES AUX INFORMATIONS CONTENUES AUX PRESENTES, CES CHANGEMENTS SERONT INCORPORES AUX NOUVELLES EDITIONS DE LA PUBLICATION. SUN MICROSYSTEMS INC. PEUT REALISER DES AMELIORATIONS ET / OU DES CHANGEMENTS DANS LE(S) PRODUIT(S) ET / OU LE(S) PROGRAMME(S) DECRITS DANS DETTE PUBLICATION A TOUS MOMENTS.