



Trusted Solaris Administrator's Procedures

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

Part Number 805-8120-10
December 2000

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, Trusted Solaris, Solaris Management Console and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. PostScript(TM) is a trademark or registered trademark of Adobe Systems, Incorporated, which may be registered in certain jurisdictions.

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

Federal Acquisitions: Commercial Software – Government Users Subject to Standard License Terms and Conditions

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. 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 fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, Solaris Management Console et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. PostScript est une marque de fabrique d'Adobe Systems, Incorporated, laquelle pourrait être déposée dans certaines juridictions. in the United States and other countries.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît 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 Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE “EN L'ETAT” ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Contents

23

1. **Assuming a Role and Working in a Role Workspace 29**
 - Accessing the Administration Tools 29
 - Administering Remote Systems 31
 - Administrative Role Procedures 32
 - ▼ To Log In and Assume an Administrative Role 32
 - ▼ To Launch the Solaris Management Console 39
 - To Log in Remotely From the Command Line 41
 - ▼ To Switch Among Administrative Role Workspaces and Normal User Workspaces 41
 - ▼ To Work at Multiple Labels While in an Administrative Role 41
 - ▼ To Launch Administrative Actions from a Local Application Manager 43
 - ▼ To Bring up the Application Manager on a Remote Computer Using `dtappsession` 44
 - ▼ To Use the Admin Editor Action to Edit a File 46
2. **Miscellaneous Tasks and Procedures 49**
 - Enforcing Security Requirements 50
 - Training Users About Security Requirements 50
 - Enforcing Password Requirements 51
 - Protecting Information 52

Protecting Passwords	53
Creating Groups	54
Deleting Users	54
Deleting Groups	54
Administering the Maximum Allowable Number of Bad Password Entries	55
▼ To Change the Maximum Allowable Number of Incorrect Password Entries	56
▼ To Prevent Account Locking for Individuals	57
▼ To Prevent Account Locking for All User Accounts	58
Enabling Keyboard Aborts	59
▼ To Enable Shutdowns Using the Keyboard Escape Sequence	59
Preventing Logins From Being Disabled After a Reboot	59
▼ To Prevent Logins From Being Disabled After a Reboot	60
Changing Rules for Upgrading and Downgrading Files and Selections	61
sel_config File Sections	64
▼ To Modify the Selection Configuration File	65
Changing Configurable Trusted Solaris Kernel Switches	66
▼ To Change Configurable Kernel Switch Settings in the /etc/system File	68
Extending Extendable Security Mechanisms	69
Adding New Authorizations	69
▼ To Add An Authorization	71
Extending Privileges	72
▼ To Add a Privilege	74
Getting a Hexadecimal Equivalent for Labels and Clearances	76
▼ To Get a Hexadecimal Equivalent for a Label	77
Changing the Front Panel	77
Changing the Workspace Menu	78
To Change the Workspace Menu	79
3. Managing User Accounts	81

Before Setting Up User Accounts	82
Making Decisions About User Accounts	83
Default User Security Attributes	85
Precedence Relationships for Attributes	88
Precedence of policy.conf and Related Attribute Sources	88
Precedence of label_encodings and Related Attribute Sources	89
Precedence of audit_control and Related Attribute Sources	89
Managing Remote Logins	90
Managing Initialization Files	91
Controlling Which Startup Files Are Read by the Window System	92
dtprofile Files	93
How the Reading of Start Up Files is Controlled for the Profile Shell User	94
Controlling Which Startup Files Are Read When a Shell Comes Up	94
Forcing dtterm to Source \$HOME/.login or .profile	95
Administering Skeleton Directories	96
Accessing All Bundled Man Pages	97
Using .copy_files and .link_files	97
Worksheet for Copy and Link Files	98
Administering the Automatic Running of Jobs Using cron, at, and batch	99
Running a Job with a Profile Shell	99
Running Privileged Commands in at or cron Jobs	100
How the UNIX Domain Socket is Used for Communications	100
Allowing Access to Jobs Owned by Others	101
Conditions for Access to Other's Jobs	101
Miscellaneous	101
Specifying an Authorization for a User or Role	102
Assigning the SMC to Normal User Accounts	103

User Setup Procedures	104
▼ To List All the Roles	104
▼ To List All Rights Profiles	105
▼ To Set Up System-Wide User Attributes	106
▼ To Make .login or .profile Looked at During Login	107
▼ To Force dtterm to Launch New Shells as Login Shells	108
▼ To Separate the Shell Initialization Files for Each Shell	109
▼ To Propagate Startup Files to Everyone's Home Directory SLDs	109
▼ To Directly Specify an Authorization for a User	110
4. Managing Roles	111
When to Create a New Administrative Role	111
Trusted Path Attribute	112
Caveats About Changing the Recommended Roles	112
Allowing Remote Logins by Administrative Roles	113
Creating a New Role	113
Customizing Profiles for the Recommended Roles	113
Aliasing vi to adminvi	114
Assigning trusted_edit as a Role's Default Editor	115
Procedures for Administering Roles	115
▼ To Make Changes to a Role	115
▼ To Configure a New Role	116
▼ To Enable Root or a New Role to Administer NIS+	117
▼ To Enable Any Role to Login Remotely	118
▼ To Assign the trusted_edit Editor to a Role	118
5. Using the SMC User Manager to Manage User and Role Accounts and Profiles	121
Adding or Modifying a User Account	121
Assigning General Attributes to Users	123

Assigning Groups to Users	125
Configuring Users' Home Directories	126
Assigning Passwords to Users	127
Configuring Password Options for Users	128
Configuring Mail Options for Users	129
Assigning Rights to Users	130
Assigning Roles to Users	131
Assigning Trusted Solaris Attributes to Users	132
Assigning Audit Classes to Users	134
Adding or Modifying a Role Account	136
Assigning General Attributes to Roles	136
Assigning Passwords to Roles	138
Assigning Users to Roles	139
Assigning Groups to Roles	139
Configuring Roles' Home Directories	140
Assigning Rights to Roles	141
Assigning Trusted Solaris Attributes to Roles	141
Assigning Audit Classes to Roles	143
Adding or Modifying a Rights Profile	144
Specifying General Attributes of Rights	145
Assigning Commands to Rights	146
Assigning Actions to Rights	147
Assigning Authorizations to Rights	149
Assigning Supplementary Rights to Rights	149
Procedures	150
▼ To Create a User Template	150
▼ To Add or Modify a User Account	150
▼ To Add or Modify a Role	151

▼	To Add a Rights Profile	151
▼	To Modify a Rights Profile	152
	To Create a Help File for a Rights Profile	152
6.	Managing Mail	155
	Managing Trusted Solaris Mail Features	156
	Allowing Users to List the Entire Mail Queue	157
▼	To Allow Listing of the Mail Queue	157
	Troubleshooting Mail Problems	158
	Tracing sendmail's Activities	158
▼	To Trace sendmail for Trusted Solaris Information	160
	Tracing Mail Delivery Difficulties	161
▼	To Check for a Properly Configured Network Connection for Sending Mail	161
	Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels	165
▼	To Configure Mail Delivery Options for Mail Below Users' Minimum Labels	166
	Substituting an Alternate Mail Application	166
	Precaution When Modifying the Default Mail Icon	167
▼	To Substitute an Alternate Mail Application in the Front Panel for All Users	168
▼	To Create a Multilevel Action for the Alternate Mail Application	171
▼	To Install an Alternate Mailer in the Front Panel	175
7.	Managing Computers and Networks	177
	Managing Trusted Network Communications	177
	SMC Tools for Administering Computers and Networks	178
	Meeting the Goals of Trusted Networking	180
	Understanding Security Attributes Assigned to Computers	180
	Host Types	181
	Computer Accreditation Range	183
	DOI	183

DOIs in Trusted Solaris IPv4 Packets	184
DOIs in Trusted Solaris IPv6 Packets	184
Default Label	185
Default Clearance	185
Forced Privileges	185
Allowed Privileges	185
Advanced Security Attributes	186
Using IP Labels in Trusted Routing	187
Default Templates	187
Default Templates for Trusted Solaris Computers	188
Default Templates for Unlabeled or RIPS0 Computers	188
Wildcard Entry and Prefix Length	189
CIPSO Labels in Packets	190
RIPS0 Labels in Packets	190
Understanding Security Attributes Assigned to Network Interfaces	192
Network Interface Accreditation Range	193
Default Security Attributes	193
Accreditation Checks	194
MAC Enforcement on Outgoing Messages	195
MAC Checks on Messages Being Forwarded	195
MAC Enforcement on Incoming Messages	196
Administering Routing	197
Background	197
Choosing Routers	198
Setting Up Trusted Routing	201
Allowing a Single-label Gateway to Forward Packets at Multiple Labels	202
8. Specifying Security Attributes for Remote Computers and Setting Up Routing	203

Assigning Security Attributes to Remote Computers and Network Interfaces	204
Setting Up Templates	205
Before Assigning Templates to Computers	205
Making Decisions About Templates	205
What You Need to Know About Trusted Network Databases	206
Modifying the Boot-time Trusted Network Databases	206
Setting Up Tunneling	207
Procedures	208
▼ To Use the Security Families Tool	208
▼ To Specify Templates for Security Families	208
▼ To Assign Templates to Computers	209
▼ To Change the Default Entry in the Boot-Time Files	210
▼ To Create a Wildcard Entry for All Computers Not Otherwise Specified	211
▼ To Configure a Network Interface	212
▼ To Set Up Static Routes with Optional Emetrics for Specific Hosts or Networks	213
▼ To Set Up Tunneling	215
9. Managing Files and File Systems	217
Specifying Security Attributes on Files and File Systems	219
Security Attributes on Files and Directories	219
Specifying Security Attributes on Files and Directories	220
Changing Labels and Privileges	221
Changing File and Directory Attribute Flags	222
Security Attributes on File Systems	222
The Label Attribute	224
Specifying Security Attributes on Variable File Systems	224
Specifying Security Attributes on Fixed File Systems	225
Mounting Various Types of File Systems in the Trusted Solaris System	226

Mount Options Used for Protection	228
Summary of Attributes on Various File System Types	229
Trusted Solaris Attribute Precedence Rules	231
Trusted Solaris and NFS	232
Exporting Directories	233
Troubleshooting Mount Failures	233
File and File System-related Procedures	234
▼ To Back Up Files	234
To Restore Files	234
▼ To Change Labels and Privileges on Files and Directories Using the File Manager	234
▼ To Specify Alternative Security Attributes While Creating a Local File System	236
▼ To Set Security Attributes on a File System	237
▼ To Specify Mount-time Security Attributes on the Command Line	238
▼ To Specify Mount-time Security Attributes in the vfstab_adjunct File	239
▼ To Share a Directory for Mounting by Other Computers	240
▼ To Mount a TMPFS-type File System Using the Command Line	242
▼ To Mount a CD-ROM with a HSFS-type File System	242
▼ To Automatically Launch a CD Player for an Audio CD-ROM	242
▼ To Listen to an Audio CD as any User or Role	243
▼ To Troubleshoot Mount Failures	243
10. Managing Name Services	245
Managing Multiple Trusted Solaris Computers in a Security Domain	246
Managing Standalone Trusted Solaris Computers	246
Allowing the root Role or a New Role to Administer A Name Server	247
Trusted Solaris NIS Maps and NIS+ Tables	247
Adding Trusted NIS Maps or NIS+ Tables	248
Adding a New Host and Giving It Credentials	248

	Name Service-Related Procedures	248
	To Save NIS Maps and Restore Them After Reinstalling the Trusted Solaris Environment	248
	▼ To Save NIS+ Tables and Restore Them After Reinstalling the Trusted Solaris Environment	249
11.	Managing Printing	253
	Configuring Printers (Trusted Solaris Tasks and Roles)	254
	Allowing the Printing of PostScript Files	255
	Adding Support for Additional File Types	256
	Setting Up Printers Without Support for Page Labels or Labeled Banner/Trailer Pages	256
	Managing Network Printers	257
	Controlling Whether Labels and Other Information are Printed on Print Jobs	257
	Labels, Job Numbers, and Handling Information on Banner and Trailer Pages	258
	Permitting Publicly-readable Jobs to Be Printed by Default Without Labeled Pages	261
	Printing-related Procedures	261
	▼ To Set Up Printing to a Non-Trusted Solaris Server	261
	▼ To Access the Printer Administrator Action	261
	▼ To Configure an Attached Printer	262
	▼ To Configure a Network Printer for Labeled Output	263
	▼ To Configure a Restricted Label Range for a Printer Managed by a Trusted Solaris Print Server	264
	266	
	▼ To Add Access to a Remote Printer	266
	▼ To Allow Some Users to Print Without Banners and Trailer Pages	266
	▼ To Assign Printing-related Authorization(s) to an Account	267
	▼ To Suppress the Printing of Page Labels on All Print Jobs	267
	▼ To Allow Some Users to Print Jobs Without Page Labels	268

▼ To Set Up Publicly-Readable Print Jobs from an Unlabeled Print Server	268
12. Managing Devices	271
Managing Allocation of Devices	272
Setting a Label Range on a Workstation	273
Setting a Label Range on a Local Printer	273
Managing Device Access Policies Set in the device_policy File	273
Managing Device Configuration and Setting Device Label Ranges	274
Understanding the Device Allocation Manager	275
When a Device is Not Available	276
Using the Device Administration and Configuration Dialogs	276
Managing Remote Devices	283
Training Authorized Users, Defining, and Enforcing Security Procedures	283
Device-related Authorizations	284
Tools for Device Management	284
Ancillary Files for Allocatable Devices	285
Allocate Error State	285
Device-Clean Scripts	286
Device-Clean Script for Tape Devices	286
Device-Clean Scripts for Floppy Disks and CD-ROM	286
Handling of CD-ROM Devices	287
Handling of Floppy Devices	287
Device-Clean Script for Audio	287
Writing New Device-Clean Scripts	288
Handling of Allocated Devices at Boot	288
Device-related Commands and Databases	289
Device Management Procedures	289
▼ To Allocate a Tape Device and Use tar to Save Security Attributes on Exported Information	289

- ▼ To Set Device Policy on a New Device or Modify Policy on an Existing Device 291
- ▼ To Use the Device Allocation Administration Dialog Box 292
- ▼ To Add or Configure a Device 293
- ▼ To Configure a Serial Line for Logins 295
- ▼ To Assign Device-related Authorization(s) to an Account 296
- ▼ To Prevent Automatic Display of File Manager After Device Allocation 298
- ▼ To Change or Add a Device Clean Script 299

13. Adding Software 301

- Types of Software 302
- System Shell 303
- The boot and inetd Profiles 304
- Types of Privileges 304
- Trusted Processes in the Window System 304
- Processes, Programs, and Their Privileges 305
 - How Programs Without Privileges Can Pass Privileges to Other Programs 306
 - When a Program File Has No Forced Privileges 307
- Assigning Privileges to Commands and Actions 307
 - Giving Forced Privileges to an Executable File 308
 - Giving Inheritable Privileges to a Command or Action 309
- Making Shared Library Directories Trusted 309
- Security Administrator Role's Tasks in Adding Software 310
 - When Adding Existing Programs 311
 - Things to Think About When a Program Fails Without Privileges 312
 - When Applications Need to Run As Root 313
 - When Adding a New Trusted Program 313
 - When Adding Actions 314
- Finding Which Privileges a Program Needs 315

Creating and Using Shell Scripts	316
Summary of Shell Script Behavior in Trusted Solaris Systems	317
More about Shell Scripts that Invoke the Profile Shell	319
How Edited Program Files Are Prevented from Being Able to Use Inheritable Privileges	320
Overview: Adding Boot Commands	320
Adding New Commands to the inittab File	321
Adding New Commands to /etc/init.d Scripts	321
Adding New Services to /etc/inet.d	322
Procedures for Adding Software	323
▼ To Create a New Administrative Action for Editing an Administrative File	323
▼ To Add Actions Outside of the System_Admin Folder	325
▼ To Mount a CD-ROM for Adding a Package	325
▼ To Find Out Which Privileges an Application Needs	326
▼ To Give Forced Privileges to a Command	329
▼ To Find Which Library Directories Are Used by an Application	330
▼ To Make a Library Directory Trusted	332
▼ To Write a Profile Shell Script that Runs Privileged Commands	333
▼ To Write a Standard Shell Script that Runs Privileged Commands When Executed in a Profile Shell	335
▼ To Add Commands to /etc/inittab	336
▼ To Start RC Scripts with Security Attributes During Boot	336
▼ To Add Services to /etc/inet/inetd.conf	338
▼ To Save and Restore Privileges When Editing a File	339
Index	341

Tables

TABLE P-1	Typographic Conventions	27
TABLE 2-1	Security Administrator and System Administrator Email Suggestions	51
TABLE 2-2	Required Attributes of <code>/etc/shadow</code>	53
TABLE 2-3	Rules and Authorizations Required for File Manager Copy and Paste, Cut and Paste, and Drag and Drop	63
TABLE 2-4	Rules and Authorizations Required for Copy and Paste, Cut and Paste and Drag and Drop of Selections Between Windows	63
TABLE 2-5	Format of Automatic Confirmation Section in <code>sel_config</code>	64
TABLE 2-6	Customer Configurable Switches in <code>/etc/system</code> File	66
TABLE 3-1	Account Security Attributes Defined in the Default <code>label_encodings</code>	86
TABLE 3-2	User Attributes and Defaults in <code>policy.conf</code>	87
TABLE 3-3	Startup Files Read by the Window System for Each Type of Login Shell	92
TABLE 3-4	Startup Files Read at Shell Initialization	95
TABLE 3-5	Planning Worksheet for Copying and Linking Startup Files Between SLDs	98
TABLE 5-1	Users Attributes	121
TABLE 5-2	General User Attributes and Where They are Entered	124
TABLE 5-3	Where Groups are Assigned to User Accounts	126
TABLE 5-4	Home Directory Properties and Where They are Specified for Users	127
TABLE 5-5	Password Properties and Where They are Specified	128

TABLE 5-6	Password Options and Where They are Specified	129
TABLE 5-7	Mail Options and Where They are Specified	129
TABLE 5-8	Rights Profiles and Where They are Specified for Users	131
TABLE 5-9	Roles and Where They are Assigned to User Accounts	131
TABLE 5-10	Trusted Solaris Attributes and Where They are Specified for Users	133
TABLE 5-11	User Audit Options and Where They are Specified for Users	135
TABLE 5-12	Roles Attributes	136
TABLE 5-13	General Role Properties and Where They are Entered	137
TABLE 5-14	Password Properties and Where They are Specified for Roles	138
TABLE 5-15	Where Users are Assigned to Roles	139
TABLE 5-16	Where Groups are Assigned to Roles Accounts	139
TABLE 5-17	Home Directory Properties and Where They are Specified	140
TABLE 5-18	Rights and Where They are Specified for Role Accounts	141
TABLE 5-19	Trusted Solaris Attributes and Where They are Specified for Roles	142
TABLE 5-20	Audit Options for Roles and Where They are Specified	144
TABLE 5-21	Rights Profiles	145
TABLE 5-22	Attributes Set in the Rights Manager General Tab	145
TABLE 5-23	Attributes Set in the Rights Manager Commands Tab	146
TABLE 5-24	Attributes Set in the Rights Manager Actions Tab	147
TABLE 5-25	Attributes Set in the Rights Manager Authorizations Tab	149
TABLE 5-26	Attributes Set in the Rights Manager Supplementary Rights Tab	149
TABLE 6-1	Features Affecting Mail and How to Change Defaults	156
TABLE 7-1	Host Types, Protocols, and Notes	181
TABLE 7-2	Wildcard Address, Netmask, and Prefix Length	189
TABLE 7-3	Supported Classifications for RIPS0 Labels	191
TABLE 7-4	Protection Authority Flags that Can Be Specified in the RIPS0 Send PAF or RIPS0 Return PAF Fields	191

TABLE 9-1	File and Directory Attributes From Trusted Solaris Operating System	219
TABLE 9-2	Variable File System Security Attributes with Defined Settings	223
TABLE 9-3	Attributes Assignable to Fixed File Systems	225
TABLE 9-4	Mount Types, Examples, and Notes	227
TABLE 9-5	Mount Restrictions, Default Values	228
TABLE 9-6	Attributes Supported by the Supported File System Types	229
TABLE 9-7	KEY to Table 9-6	230
TABLE 10-1	Trusted Solaris Added NIS+ Tables	247
TABLE 11-1	Tasks for Configuring Printers	254
TABLE 11-2	Modifiable Printing Features	260
TABLE 12-1	Default Device Access Policy	273
TABLE 12-2	Specifying Only Local Allocation of the Audio Device	279
TABLE 12-3	Treating Local and Remote Allocation of a Tape Device the Same	280
TABLE 12-4	Requiring Two Different Authorizations for a CDROM Device	280
TABLE 12-5	Device Allocation, Configuration, and Management Authorizations	284
TABLE 12-6	Required Ancillary File Characteristics for Allocatable and Allocated Devices	285
TABLE 12-7	Device-related Commands and Databases	289
TABLE 12-8	Default Device Policy	292
TABLE 12-9	Default Device Allocation Authorization and Default Profiles that Include It	297
TABLE 12-10	Device Deallocation and Reclamation Authorization, Default Profiles that Include It, and Default Roles Assigned It	297
TABLE 12-11	Device Configuration Authorization, Default Profiles that Include It, and Default Roles Assigned It	298
TABLE 13-1	Differences in Creating and Using Actions Under Trusted Solaris Restraints	314

Figures

Figure 2-1	File Manager Selection Confirmer	62
Figure 3-1	How \$HOME/.dtprofile is installed	93
Figure 3-2	How \$HOME/.dtprofile is Bypassed When Users have a Profile Shell	94
Figure 6-1	Sendmail Data Flow Example	159
Figure 7-1	SMC Tools	179
Figure 7-2	Interface Manager with Default Security Attributes	192
Figure 7-3	How a Host Determines Which Type of Routing to Do	201
Figure 8-1	Interface Manager with Default Security Attributes	213
Figure 9-1	File Manager Selected Menu for an Authorized User	222
Figure 9-2	Trusted Solaris Attribute Precedence Rules	232
Figure 11-1	Job's Label Printed on Body Pages	258
Figure 11-2	Typical Print Job Banner Page	259
Figure 11-3	Differences on a Trailer Page	259
Figure 12-1	Device Allocation Configuration Dialog	278
Figure 12-2	Clicking the Authorizations Button Displays the Device Allocation: Authorizations Box	282
Figure 12-3	SMC Tools	295
Figure 13-1	How a Program that Cannot Use Privileges Can Pass Them to a Program that Can	306

Figure 13-2 How Forced Privilege Shell Scripts Are Prevented from Passing Forced Privileges to Their Commands 307

Figure 13-3 How Normal Shell Scripts Invoked in `pfsh` Can Pass Inheritable Privileges to Their Commands 318

This *Trusted Solaris Administrator's Procedures* manual provides procedures for managing users and hosts while maintaining the security of information within the Trusted Solaris™ environment.

Who Should Use This Book

This book is used by administrators who are able to assume any of the Trusted Solaris administrative roles. This book describes how to do the unique Trusted Solaris administrative tasks that are an essential part of protecting the security of the system.

Before You Read This Book

- ◆ **Understand Solaris 8 administration, CDE, Solaris Management Console™, and NIS+**

The procedures in this manual are unique to Trusted Solaris administration. An administrator must already understand how to work within and administer the Solaris 8 operating environment, upon which the Trusted Solaris system is based, and understand how to use and administer the Common Desktop Environment (CDE) window system, and Solaris Management Console administration tools.

Note - AnswerBooks for the above-mentioned products that are bundled into Trusted Solaris are available on the *Trusted Solaris 8 AnswerBook* CD, which is shipped with the Trusted Solaris 8 product CD.

- ◆ **Read and understand the basic concepts and procedures for using the system, as described in the *Trusted Solaris User's Guide***

Administrators should know how to work in the Trusted Solaris environment as a normal user.

- ◆ **Read and understand the administrative concepts described in the *Trusted Solaris Administration Overview***
- ◆ **Understand how administrative tasks are divided among roles at your site**
Each procedure identifies which role is assigned to the task in the default configuration. The security administrator is responsible for informing administrators if the default administrative roles have been reconfigured.

How This Book Is Organized

Chapter 1

Reviews how to assume an administrative role, work in an administrative role workspace, launch administrative actions from the workspace, invoke administrative commands in the profile shell, and prevent logins from being disabled after a reboot.

Chapter 2

Describes security processes for administrators to put into effect and provides procedures for how to change the Front Panel, change the Workspace Menu, change configurable kernel switches, distribute changed configuration files to all hosts, change the rules for upgrades and downgrades, enable keyboard aborts, change the maximum number of bad password entries before an account is locked, get hexadecimal equivalents for labels and clearances, and extend authorizations.

Chapter 3

Describes what decisions to make and what to do before setting up accounts for users and roles, and how to administer startup files and batch jobs.

Chapter 4

Describes the differences between user and role accounts and how the responsibilities for managing roles are divided. Describes the Custom Role Profiles, and gives an overview of when and how to create a new role or modify an existing role. Also describes how to perform role-specific procedures.

Chapter 5

Describes the information that must be provided in each of the fields of the User Accounts tool to configure user accounts, in the Administrative Roles tool to configure role accounts, and in the Rights tool to configure rights profiles.

Chapter 6

Describes the differences between standard Solaris and Trusted Solaris mail administration, including the new Trusted Solaris `sendmail` debugging options and the new privacy options in the `sendmail.cf` file for handling mail that is received below an account's minimum label.

Chapter 7

Reviews concepts that apply to managing communications and shows how trusted communications are configured between the Trusted Solaris distributed system and multiple networks. Describes routing and trusted routing.

Chapter 8

Describes how to specify the security attributes for all hosts with which communications are allowed and how to set up routing for trusted network communications.

Chapter 9

Describes the extended file system security attributes, how to set up mounts across the distributed system, and how to specify the extended security attributes.

Chapter 10

Describes how NIS and NIS+ name services can be used to centrally administer the Trusted Solaris system.

Chapter 11

Describes how to configure printing, how to add and delete labeled and unlabeled printers, and provides pointers to directions about specifying handling caveats for printer banner and trailer pages.

Chapter 12

Devices that can be used to export and import data are not available to everyone. Access to devices is controlled by the device allocation mechanism. *Managing Devices* describes how to set up device allocation, how to make devices allocatable, how to write and specify device clean scripts to run when a device is deallocated and how to respond to the allocate error state. This chapter also includes how to set the label range on printers and workstations, which are not allocatable devices.

Chapter 13

Describes how to add Sun unbundled products, other UNIX™ applications, new trusted programs, CDE actions, and shell scripts, and how to assess what privileges a new program needs and to decide whether to give a program the needed privileges. This chapter describes the two ways that privileges are made available to software.

Related Books

- *Trusted Solaris User's Guide*

The rest of the Trusted Solaris administrator's document set:

- *Trusted Solaris Administration Overview*
- *Trusted Solaris Audit Administration*
- *Trusted Solaris Installation and Configuration*
- *Trusted Solaris Label Administration*
- *Trusted Solaris Developer's Guide*
- *Trusted Solaris 7 Reference Manual*
- *Trusted Solaris 7 Release Notes*
- *Trusted Solaris 7 Transition Guide*
- *Compartmented Mode Workstation Labeling: Encodings Format*

Ordering Sun Documents

Fatbrain.com stocks documentation from Sun Microsystems, Inc.

For a list of available documents and how to order them, visit <http://www1.fatbrain.com/documentation/sun>.

Accessing Sun Documentation Online

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

Type Styles Used in Text and Examples

The following table shows and explains the type styles used in this manual.

TABLE P-1 Typographic Conventions

Type Face	Meaning	Example
Literal	The names of commands, files, and directories, on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>hostname%</code> <code>You have mail.</code>
UserType	What you type, contrasted with on-screen computer output	<code>hostname% su</code> <code>Password:</code>
Variable	Argument name in a command-line. You replace the argument with a real name or value.	To delete a file, enter <code>rm filename</code> . <code>hostname% rm myfile</code>
Title or Emphasis	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class options</i> . <code>You must be root to do this.</code>

Trusted Solaris Prompts

The following table shows the Trusted Solaris prompts.

Shell	Prompt
C shell prompt	<code>hostname%</code>
Bourne shell and Korn shell prompt	<code>\$</code>
Profile Shell prompt	<code>\$</code>

Shell	Prompt
root prompt (with any shell)	#
PROM mode prompt (SPARC only)	>

Assuming a Role and Working in a Role Workspace

This chapter contains the following sections.

- “Accessing the Administration Tools” on page 29
- “Administering Remote Systems” on page 31

This chapter contains the procedures listed here:

- “To Log In and Assume an Administrative Role” on page 32
- “To Launch the Solaris Management Console” on page 39
- “To Log in Remotely From the Command Line” on page 41
- “To Switch Among Administrative Role Workspaces and Normal User Workspaces” on page 41
- “To Work at Multiple Labels While in an Administrative Role ” on page 41
- “To Launch Administrative Actions from a Local Application Manager” on page 43
- “To Bring up the Application Manager on a Remote Computer Using `dtappsession`” on page 44
- “To Use the Admin Editor Action to Edit a File” on page 46

Accessing the Administration Tools

As described in the *Trusted Solaris Administration Overview*, administrative tasks are performed by multiple administrative roles. The user who is configured to assume a role does the following sequence (which is detailed in “To Log In and Assume an

Administrative Role” on page 32) to access the tools needed to perform the tasks assigned to the role.

- The user logs in by supplying his or her own username and password, and then a normal user workspace becomes active without the trusted path attribute.

(The trusted path attribute is checked by many administrative programs.)

- The user chooses the `Assume rolename Role` option from the Trusted Path (TP) menu in the Front Panel and enters the role password in the role password dialog box.

As soon as the user is authenticated and the role is assumed, an administrative role workspace becomes active with the trusted path attribute. The person who has assumed the role then can perform administrative tasks. Each role is constrained to use only those tools listed in the rights profile(s) that are in effect for that role. See the following table for a description of the administrative tools and references to where their use is described.

Solaris Management Console tool or equivalent <code>sm*</code> commands, such as <code>smuser(1M)</code> and <code>smrole(1M)</code> .	Used for most configuration of user accounts, hosts, and networks. Can update local files or name service databases. Can also launch legacy applications: <code>dtterm(1)</code> and <code>dtappsession(1)</code> .	Note - Authorizations are used to control which tools or fields can be accessed by each role in the Solaris Management console and which options can be used in the <code>sm*</code> commands. See “To Launch the Solaris Management Console” on page 39.
Trusted Solaris administrative actions in the System_Admin Folder in the Application Management Folder	Used to edit files not managed with the SMC.	See “To Launch Administrative Actions from a Local Application Manager” on page 43.
Miscellaneous administrator’s utilities executed in the administrator’s profile shell. Miscellaneous actions launched from the workspace menu and the front panel.	Used to perform tasks not done by the first two types of tools above.	See <i>man pages section 1M: System Administration Commands</i> .

Administering Remote Systems

Administrators can do administration on remote hosts in several ways that are described in this manual, as summarized below:

- After logging into the local host and assuming a role, administrators can log into a remote host from a terminal in an administrative workspace and use `rlogin(1)`, `telnet(1)`, or `ftp(1)`. If either `rlogin` or `telnet` are used, a role can enter commands on the remote host that are in that role's rights profiles.

See “To Log in Remotely From the Command Line” on page 41 for how roles can log in remotely and work on the command line.

- From a local host's CDE login screen, anyone can log directly into the CDE window system on a remote host (as described in Step 1 on page 32).

After CDE remote login is complete, the CDE window environment from the remote host displays on the screen of the local host. An administrator can then assume a role from the Trusted Path menu and work as if logged in directly to the remote host.

- Administrators can launch a Solaris Management Console (SMC), server that is running on a remote host.

Accessing the SMC is described in “To Launch the Solaris Management Console” on page 39. The CDE Application Manager can be started remotely by double-clicking the Application Manager icon from the Legacy Application list. (See the next bullet for more about using the Application Manager to make changes on a remote host.)



Legacy Application

- While working in the local host's administrative workspace, the role can use the `dtappsession(1)` command to launch an Application Manager that runs and makes changes on the remote host.

The `dtappsession(1)` script is used to start an independent instance of the CDE Application Manager that runs on the remote host and displays on the local host. Unlike CDE remote login, `dtappsession` enables the administrator to work remotely within a local login session. (CDE remote login, in contrast, requires the administrator to completely log out before starting a remote session.) The procedure is described in “To Log In and Assume an Administrative Role” on page 32.

`dtappsession` is useful when a remote host does not have a monitor. For example, `dtappsession` is often used instead of CDE remote login when administering domains on large servers, such as the E10000. The remote host

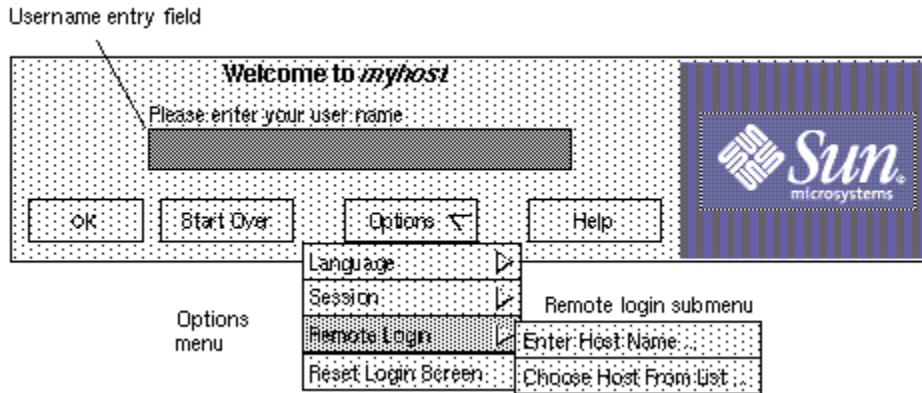
being administered by means of the remote Application Manager can be either a Trusted Solaris or regular Solaris host running CDE.

Administrative Role Procedures

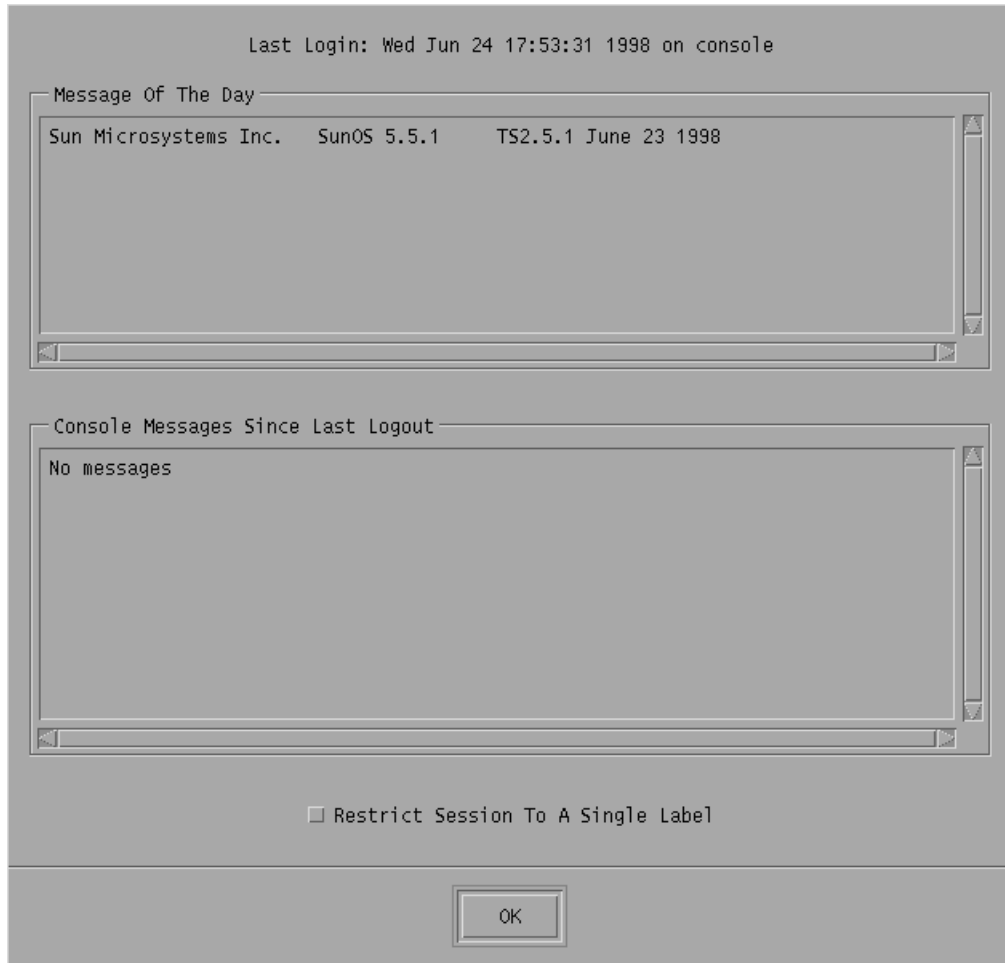
▼ To Log In and Assume an Administrative Role

1. To start a login session on a remote host, if desired, use the Remote Login option from your local host's Login Screen.

The following figure shows the Login Screen with username entry field and the Options button's pull-down menu.

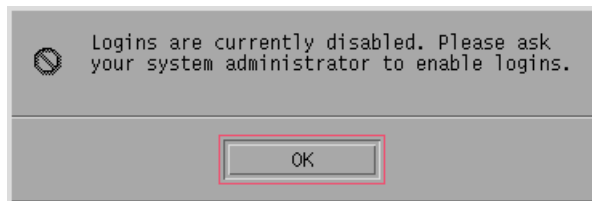


- a. Choose Remote Login from the Options button pull-down menu.
 - b. Choose either Enter Host Name or Choose Host from List.
2. Whether using the CDE remote login option or logging directly into the local host, enter your own username and supply a password when prompted.
 3. If the workstation information dialog box displays as shown below, go to Step 5 on page 34.

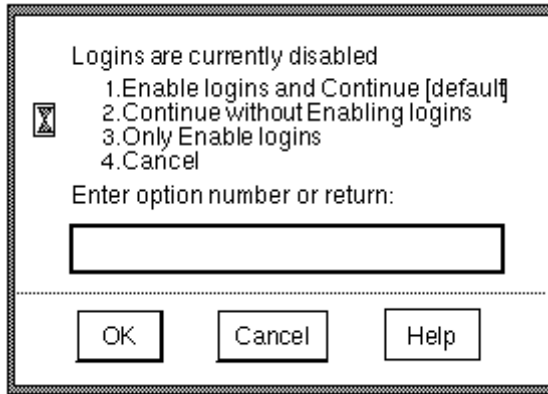


4. Enable logins if prompted to do so.

- a. If the following dialog box displays, your account is not authorized to enable logins. Ask the security administrator role to give you the needed authorization or ask an authorized person to enable logins.**



- b. If your account is authorized to enable logins, choose one of the options in the following dialog or click OK to enable logins.**

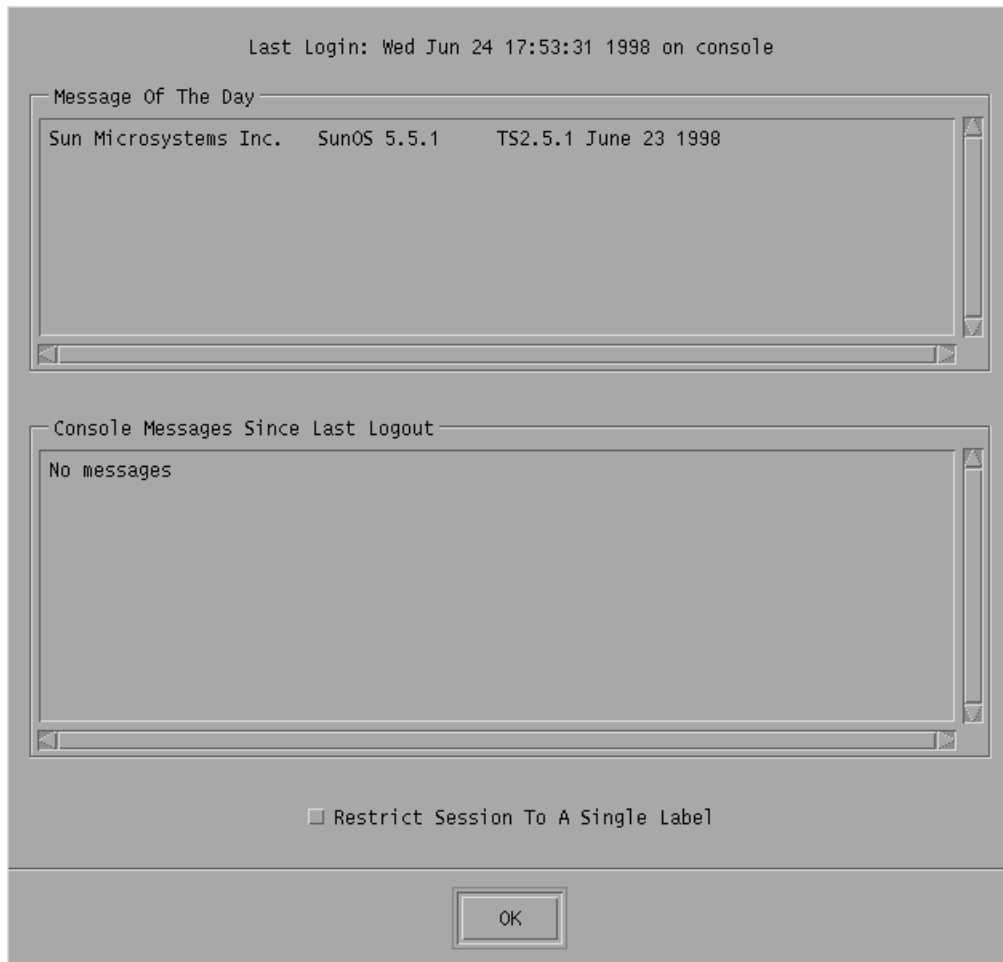


When the workstation information dialog box displays, go to the following step.

5. Review the information provided on the workstation information dialog box and, if allowed, choose between a single label or multiple label session.

If your account is configured to be able to work with multiple labels, the Restrict Session to a Single Label toggle box displays at the bottom of the workstation information dialog box to give you the option to work at a single label for the duration of the login session. If your account is configured to work at only one label, Single Level Session SL: displays at the bottom of the dialog box followed by the name of the label.

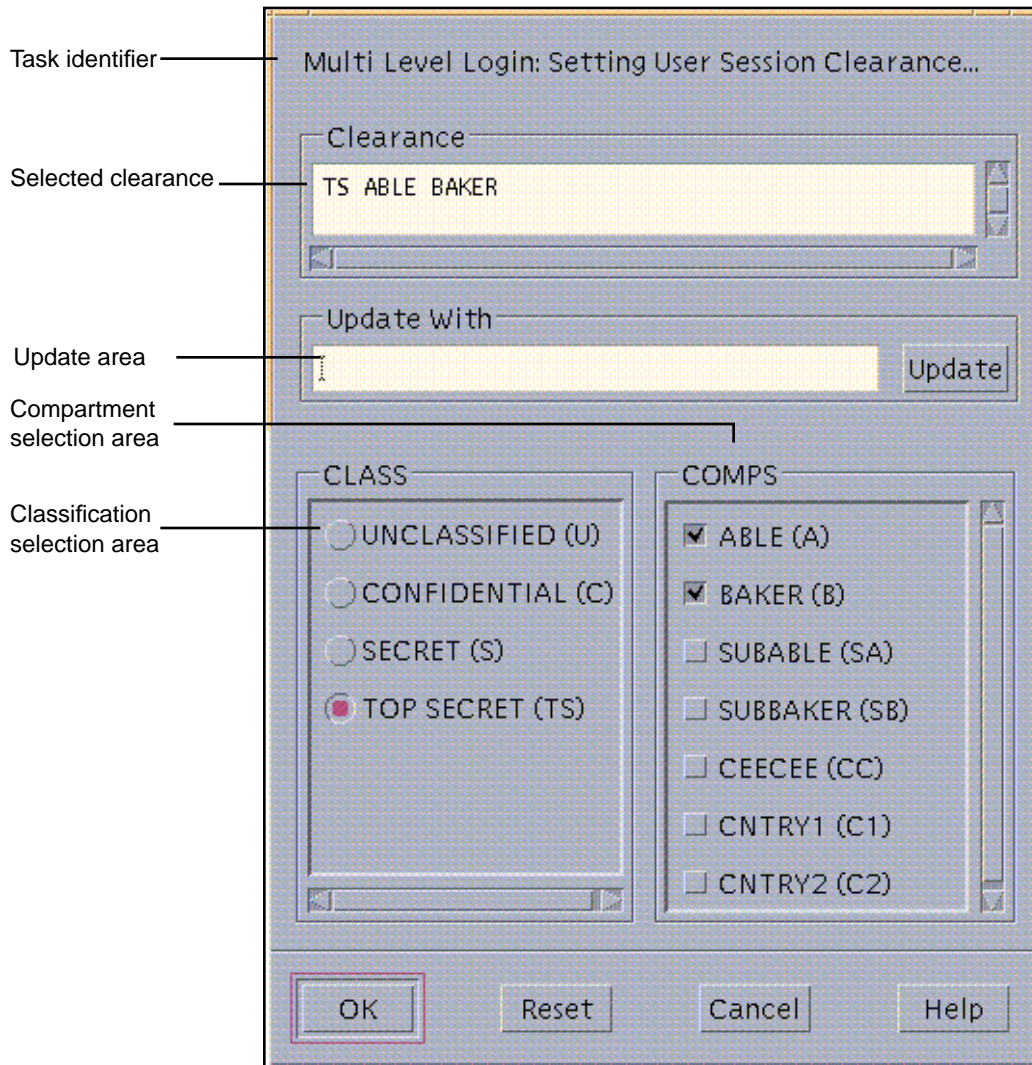
- a. **Check the date and time of the last login to ensure there is nothing to arouse suspicion about the last login, such as an unusual time of day.**
- b. **Read the message of the day.**
- c. **Check console messages since last logout.**
- d. **Investigate any suspicious logins, messages that could indicate inappropriate activities or other problems.**
- e. **If your user account is configured to be able to work with multiple labels, check or ignore the Restrict Session to a Single Label toggle box.**



6. Press Return or click OK to close the workstation information dialog box.

If you checked the box next to Restrict Session to a Single Label, the Single-Label Login: Setting Session Label dialog box displays. Go to Step 7 on page 36.

If you are allowed to work at multiple labels and if you did not restrict the session to a single label, the Multilabel Login: Setting Session Clearance dialog box displays. Go to Step 8 on page 37.



7. To specify a label for a single-label session, either accept the default label or specify a label in the Single Level Session: Setting User Session SL Label Builder.
 - a. To type in the label, use the text entry field under Update With, and click the Update button when done.
 - b. To use the mouse to build the label, highlight a classification on the menu and select the compartments by checking the boxes next to the compartment names.

- c. Click OK.
- d. Go to Step 9 on page 37.

8. To specify the clearance for a multilabel session, either accept the default clearance displayed in the Multi Level Login Setting Session Clearance dialog or specify another clearance.
- a. To type in the clearance, use the text entry field below Update With and click Update when done.
 - b. To use the mouse to build the clearance, choose a classification from the Class menu and select the compartment components by checking the boxes next to the compartment names in the Comps column.

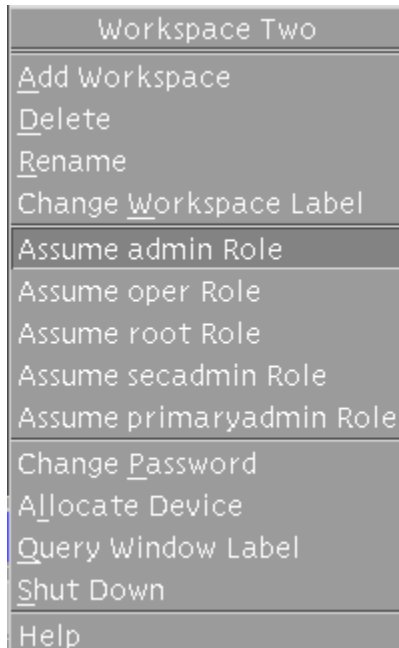
Note - Other words may be configured at your site to appear instead of Class and Comps. See “Changing Label Component Names on Label Builders” in *Trusted Solaris Label Administration*, if desired, for how the words can be changed.

- c. Click OK.
- d. Go to the following step.

9. Choose the Assume *role_account_name* Role option from the Trusted Path menu.

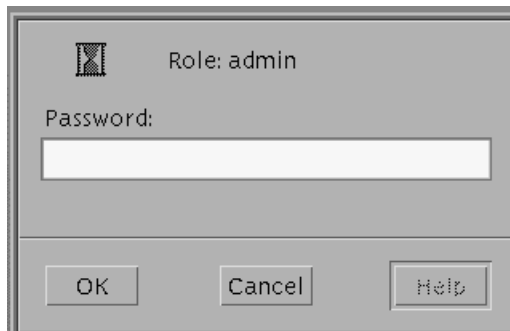
The following figure shows the Trusted Path (TP) menu for a user who is configured to assume the System Administrator role.

Note - The option Assume *role_account_name* Role appears on the Trusted Path menu only for users who are configured to assume a role.



The Role Password dialog box displays.

10. Enter the role password in the role password dialog box, and click OK.



An administrative role workspace becomes active, and a new administrative role workspace button is added to the workspace switch area.

11. When ready to log out, click the EXIT button in the center Front Panel.

▼ To Launch the Solaris Management Console

Note - The first time the Solaris Management Console is launched and the load button is clicked, a delay occurs while the tools are registered and the `/var/sadm/smc/` directory and its subdirectories are created.

When a naming service is being used, the toolbox with the appropriate scope (either NIS or NIS+) must be edited on the name service master and can also be edited on the clients. On a name server client, you must edit the toolbox when you want to be able to use a toolbox from the local computer with the NIS or NIS+ scope. For the procedure for editing the toolboxes, see “To Edit Name Service Toolbox Definitions” in *Trusted Solaris Installation and Configuration*.

Note - Use of a name service is recommended because it supports centralized administration of all user, host, and network information, which is important for both user accountability and trusted administration. The `Files` scope should be used to manage users and hosts only in specialized circumstances where a knowledgeable security administrator decides that local accounts are both needed and allowable within your organization’s security policy—even though they can make the system both harder to protect and to maintain.

1. **Assume a role that is configured to use the Solaris Management Console (SMC) and launch the tool in an administrative role workspace at ADMIN_LOW in any of the following ways:**
 - a. **From the Front Panel -> Tools menu, choose the Solaris Management Console option.**
 - b. **On the Front Panel -> Applications menu, click the Applications option, and then double-click the Solaris Management Console icon.**
 - c. **Invoke the `smc(1M)` command in a terminal.**
 - d. **From the Workspace Menu->Tools sub-menu, choose the Solaris Management Console option.**

2. **Select the name of a computer where the SMC server software is running from the Server list, or type the name of the computer in the Server field, and then click the Load button.**

The term *server* in this context is used to refer to a computer where the SMC server software is running.

Note - Ignore the Host Not Found error that displays in the Toolboxes field. When you click the Load button, the error is replaced with a list of toolboxes available on the computer whose name is in the Server field.

The names of any SMC toolboxes on the specified server are loaded into the Toolboxes field.

3. Select a toolbox with the desired scope.

The console or individual toolbox you selected display in the navigation pane. The name of each toolbox starts with the name of the host where the SMC server software is running followed by one of three different scopes (Files, NIS, or NIS+), and then by a policy assignment. For example, the following shows the toolbox with the Files scope and the TSOL Policy for the Server domus:

domus: Scope=Files, Policy=TSOL

Note - When you are working on a Trusted Solaris computer, make sure that the Policy=TSOL on the toolbox you select. Only if you were administering a Trusted Solaris system remotely from a Solaris host would you select a toolbox whose Policy=SUSER. If no policy is specified in the toolbox name, the default is SUSER.

Scope Name	Updates
Files	Local files on the current computer.
NIS	NIS maps on the NIS name server for a NIS client host.
NIS+	NIS+ tables on the NIS+ name server for a NIS+ client host.

4. Save the current toolbox as a preference (if desired, to save reloading next time):

- a. Choose Console->Preferences.
- b. On the Console tab, click the Use Current Toolbox button.
- c. Click OK.

5. Bring up the desired SMC tool.

The `Login: Role Name` dialog box displays with the name of the server in the `SMC Server` field, with your username in the `User Name` field, and with the name of the current role in the `Role Name` field.

6. **Type the role's password in the `Role Password` field.**
See other chapters in this manual for how to use the `User Manager`, `Interface Manager`, and `Security Families` tools. See the `SMC` help for additional information about the above-named tools and all other `SMC` tools.
7. **When done, choose `Exit` from the `Console` menu.**

To Log in Remotely From the Command Line

1. **Make sure that remote logins are enabled for roles.**
See “Allowing Remote Logins by Administrative Roles” on page 113, if desired.
2. **Log into the local computer and assume a role.**
3. **Bring up a terminal in the administrative role workspace and either enter `rlogin(1)` or `telnet(1)`, or `ftp(1)` in the terminal to log into a remote host**
4. **If `rlogin` or `telnet` are used to log in, enter on the command line of the login shell on the remote computer any commands assigned in the current role's rights profiles. If `ftp` is used, see the `ftp(1)` man page for the commands you can use.**

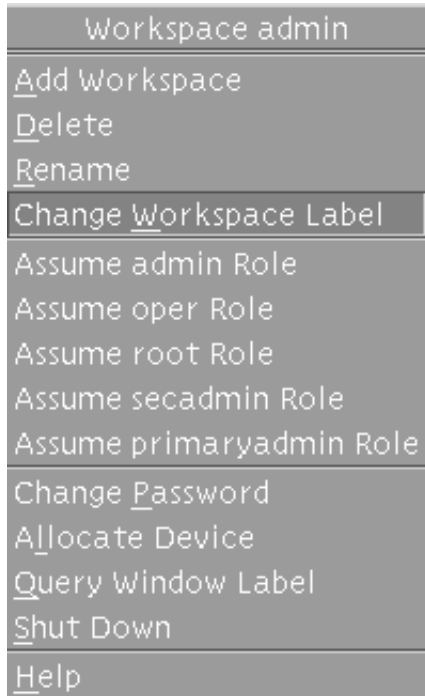
▼ To Switch Among Administrative Role Workspaces and Normal User Workspaces

- ◆ **Click the desired workspace's button.**

▼ To Work at Multiple Labels While in an Administrative Role

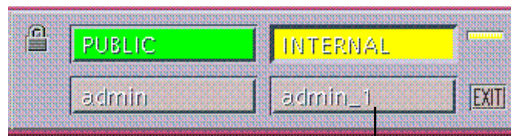
Working at multiple labels requires creating new administrative role workspaces and relabeling them.

1. **Add a new administrative role workspace.**
 - a. **With the cursor over an administrative role workspace button, click and hold the right (menu) mouse button to bring up the `Trusted Path` menu.**



b. Choose Add Workspace from the `Workspace role_name` menu.

A new administrative role workspace becomes active, and a new administrative role workspace button appears in the workspace switch area in the Front Panel.



New workspace button

By default, the name of new workspace is the name of the role account followed by an underline followed by a number. As shown in the example, the name of a second administrative workspace created for the admin role is `admin_1`.

2. Change the label of the workspace.

a. Click the cursor on the new role workspace button.

b. Hold down the right (menu) mouse button on the workspace switch button to bring up the `Trusted Path` menu.

- c. **Choose Change Workspace Label from the menu.**
The Label Builder displays.
- d. **On the Label Builder dialog box, do the following:**
 - i. **Type the desired label in the text entry field under Update With.**
 - ii. **Click the Update button.**
 - iii. **Click OK at the bottom of the dialog box when done.**

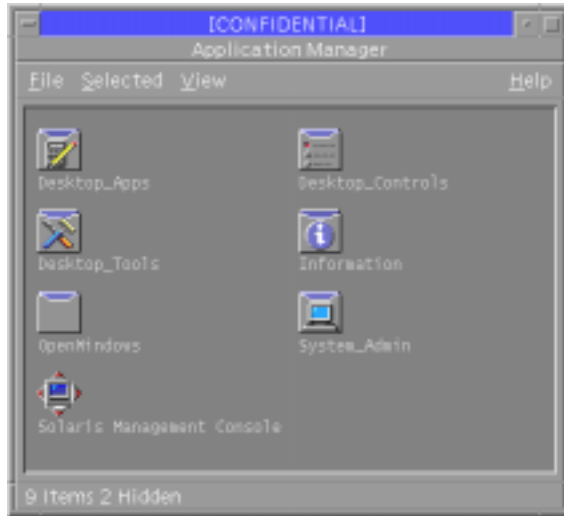
The label of the workspace changes to the label you specified in the label builder.

▼ To Launch Administrative Actions from a Local Application Manager

1. **Log in as a user who is able to assume an administrative role and assume the role.**
See “To Log In and Assume an Administrative Role” on page 32 if needed.
2. **Click the Application Manager icon from the Applications subpanel on the Front Panel.**



The Application Manager folder displays.



3. Double-click the `System_Admin` icon in the Application Manager folder.
4. Double-click the icon for the desired action.

▼ To Bring up the Application Manager on a Remote Computer Using `dtappsession`

1. Make sure the administrative role can remotely log into the computer (as described in “Allowing Remote Logins by Administrative Roles” on page 113).
2. If administering NIS+ from any NIS+ clients, make sure that each NIS+ client’s name is entered in the NIS+ admin group on the domain’s NIS+ master, as described in “To Enable Root or a New Role to Administer NIS+” on page 117.
3. Make sure that the `CONSOLE=/dev/console` line in the `/etc/default/login` file is commented out on the current host (see the procedure described in “To Enable Any Role to Login Remotely” on page 118).
4. Assume an administrative role that either has `dtappsession(1)` in one of its rights profiles or that has the authorizations to use the SMC.

Note - The `dtappsession` command is in the Remote Administration profile that is included in the default profiles for all the recommended roles. The command can be launched from an administrative role workspace or can be launched as a Legacy Application in the SMC. In the list of Legacy Applications, you can differentiate the tool for the `dtappsession` command by looking for the Application Manager icon that appears to the left of the words Legacy Application.

5. To use the `dtappsession` command from the SMC, double-click the File Manager icon in the list of tools.



Legacy Application

Go to Step 7 on page 46

6. To use the `dtappsession` command in a terminal, do the following:
 - a. To avoid confusion between the remote CDE applications and any local ones, dedicate an administrative role workspace to this procedure.
See “To Work at Multiple Labels While in an Administrative Role ” on page 41 for how to add an administrative role workspace, if needed.
 - b. In the new dedicated workspace, use the `rlogin(1)` command followed by the name of the remote host where you wish to do the remote administration.

The following screen shows the `rlogin` command entered with the name of an E10000 domain called `e10000domain1`.

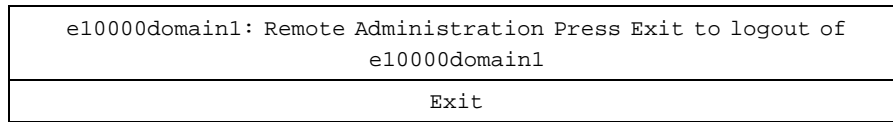
```
# rlogin e10000domain1
```

- c. Enter `dtappsession` followed by the name of the local host.
Either the name of the local host must be specified on the `dtappsession` command line, or the `DISPLAY` environment variable must be set on the remote host with the name of the local host. The following screen shows the command entered with the local host name of `ssp_host`.

```
# /usr/dt/bin/dtappsession ssp_host
```

An Application Manager that is running on the remote host displays on the local host.

As shown in the following figure, dtappsession brings up a Remote Administration dialog box with the name of the remote host followed by the words: Remote Administration. An Exit button displays at the bottom of the screen. The example shows the wording when the remote host's name is e1000domain1:



7. When finished using the remote Application Manager, click the Exit button on the Remote Administration dialog box.



Caution - Be aware that closing the Application Manager does not end the session.

8. If you launched dtappsession from a terminal, do the following:
 - a. Exit the remote login session by entering exit in the terminal where you entered the dtappsession command.

```
# exit
```

- b. Enter the hostname command to verify that terminal is returned to the local host.

```
# hostname  
ssp_host
```

▼ To Use the Admin Editor Action to Edit a File

1. Launch the Admin Editor action to open the file for editing.
See "To Launch Administrative Actions from a Local Application Manager" on page 43, if needed.

a. Double-click on the Admin Editor icon.

An Action: Admin Editor prompt displays.

b. In the File to Edit field, type in the pathname for the file

c. Click the Okay button.

2. Use `adminvi(1M)` text-editing commands to edit the file.

`adminvi` commands are in most cases the same as `vi(1)` commands, with exceptions as noted on the `adminvi(1M)` man page.

3. When finished editing, save the changes and quit the file.

`:wq`

If you get an error when you try to write the file with `:wq`, use `:wq!` instead.

Miscellaneous Tasks and Procedures

This chapter provides information about tasks and procedures for miscellaneous tasks that do not fit neatly into other chapters in this manual. It includes the following main topics:

- “Enforcing Security Requirements” on page 50
- “Administering the Maximum Allowable Number of Bad Password Entries” on page 55
- “Enabling Keyboard Aborts” on page 59
- “Preventing Logins From Being Disabled After a Reboot” on page 59
- “Changing Rules for Upgrading and Downgrading Files and Selections” on page 61
- “Extending Extendable Security Mechanisms” on page 69
- “Adding New Authorizations” on page 69
- “Extending Privileges” on page 72
- “Changing the Front Panel” on page 77
- “Changing the Workspace Menu” on page 78
- “Changing Configurable Trusted Solaris Kernel Switches” on page 66

This chapter includes the following procedures:

- “To Change the Maximum Allowable Number of Incorrect Password Entries” on page 56
- “To Prevent Account Locking for Individuals” on page 57
- “To Prevent Account Locking for All User Accounts” on page 58
- “To Enable Shutdowns Using the Keyboard Escape Sequence” on page 59
- “To Prevent Logins From Being Disabled After a Reboot” on page 60
- “To Modify the Selection Configuration File” on page 65

- “To Change Configurable Kernel Switch Settings in the /etc/system File” on page 68
- “To Add An Authorization” on page 71
- “To Add a Privilege” on page 74
- “To Get a Hexadecimal Equivalent for a Label” on page 77

Enforcing Security Requirements

To ensure that the security of the system is not compromised, administrators need to protect passwords, files and audit data and to train computer users to do their part. To be consistent with the requirements for an evaluated configuration, follow the guidelines in this section.

Training Users About Security Requirements

Each site's security administrator ensures users are trained. The security administrator should hand off the following rules to new employees and remind existing employees of these rules from time to time.

Note - Your organization may wish to provide additional suggestions beyond those shown below.

Users' Security Rules

Anyone who knows your password can access the same information that you can without being identified and therefore without being accountable.

Do not tell anyone else the password.

Do not write the password down or include it in an email message.

Choose passwords that are hard to guess.

Do not leave your workstation unattended without locking the screen or logging off.

Be aware that sender information in email can be forged.

Remember that administrators do not rely on email to send instructions to users. Do not ever follow instructions from administrators in an email without first double-checking with the administrator.

Do not send your password to anyone by email.

Because you are responsible for the access permissions on files and directories that you create, make sure that the permissions on your files and directories do not allow unauthorized users to read or change a file or list the contents of or write into a directory.

Enforcing Password Requirements

The system administrator role is responsible for specifying the original password for each account and for handing off the passwords to new accounts. The system administrator role must specify a unique user name and a unique user ID when creating a new account. When choosing the name and ID for a new account, the administrator must ensure that both the user name and associated UID are not duplicated anywhere on the network and have not been previously used.

Security Administrator Password Administration Rules

Make sure that the accounts for users who are able to assume the security administrator role are configured so that the account cannot be locked, by setting `Lock Account After Maximum Failed Logins` to `No` in the Account Usage dialog under User → Properties in the SMC User Manager.

This ensures that at least one account can always log in and assume the security administrator role to reopen everyone's account if it ever happens that all other accounts are locked.

Hand over the password to an account in such a way that the password cannot be eavesdropped by anyone else.

Change an account's password if there is any suspicion that the password has been discovered by anyone who should not know it.

Never reuse user names or UIDs over the lifetime of the system.

Ensuring that user names and UIDs are not reused prevents possible confusion over:

- Which actions were performed by which user when audit records are analyzed
- Which user owns which files when archived files are restored

TABLE 2-1 Security Administrator and System Administrator Email Suggestions

Never use email to instruct users to take an action.

Tell users not to trust email with instructions that purport to come from an administrator.

This prevents the possibility that spoofed email messages could be used to fool users into changing a password to a certain value or divulging the password, which could subsequently be used to log in and compromise the system.

Roles Changing Their Own Passwords

All the roles are able to change their own passwords using the Change Password option from the TP (Trusted Path) menu.

Note - The Security Administrator and the System Administrator can change their own passwords on the command line using `passwd(1)`. The `passwd(1)` command updates the name service database, and so the use of either `yppasswd(1)` for NIS or `nispasswd(1)` for NIS+ is not necessary and is strongly discouraged on the man pages.

Roles Changing Others' Passwords

The security administrator role can change any account's password at any time except for the password of the root role. Because root's UID 0 is below 100, the SMC considers root to be a "system account," and the SMC does not allow any changes to be made to system accounts. If root's password needs to be changed, root must make the change using the TP menu Change Password option. The security administrator role changes user passwords by launching the SMC User's tool and modifying the properties for a user. The security administrator role changes role passwords by launching the Role tool and modifying the properties for a role.

The Primary Administrator role can type the `passwd(1)` command in a privileged shell to change any password, but this should only be necessary in exceptional circumstances.

Protecting Information

Administrators are responsible for correctly setting up and maintaining DAC and MAC protections for security-critical files, such as the `shadow(4)` file containing encrypted passwords, the local `prof_attr(4)`, `exec_attr(4)`, and `user_attr(4)` databases, and the audit trail.



Warning - Because the protection mechanisms for NIS maps and NIS+ tables are not subject to the access control policy enforced by the Trusted Solaris system, the default NIS maps and NIS+ tables should not be extended, and their access rules should not be modified.

Protecting Passwords

In local files, passwords are protected from viewing by DAC and from modifications by both DAC and MAC. Passwords for local accounts are maintained in the `shadow(4)` file that is readable only by root:

```
trusted4% ls -l /etc/shadow
-R.----- 1 root
```

The security administrator role should ensure that the DAC and MAC attributes are not changed for the `/etc/shadow` file. The attributes that must be maintained on the `shadow` file are shown in the following table.

TABLE 2-2 Required Attributes of `/etc/shadow`

MAC	ADMIN_LOW		
DAC	<i>owner</i>	<i>group</i>	<i>permissions</i>
	root	sys	400

The password field in the NIS+ `passwd.org_dir` table is protected by NIS+ restrictions on access to fields within tables. When any user or administrator tries to view the `passwd.org_dir` table, the only encrypted password that displays is the one belonging to the account.

The following example shows that while user `ricc`'s password field shows as `*NP*` when the user `roseanne` invoked the `niscat(1)` command, `roseanne` can see the encrypted password for her own account.

```
trusted5% whoami
roseanne
trusted6% niscat passwd.org_dir
. . .
ricc:*NP*:33333:10:Ric Cheshire:/home/ricc:/bin/csh:*NP*
roseanne:0dk1EW44:10:Roseanne Sullivan:/home/roseanne:/bin/csh:38442::::::
```

Also, as shown in the following example, there is no `shadow.org_dir` table.

```
trusted5% niscat shadow.org_dir
shadow.org_dir: Not Found, no such name
```

With NIS, configure the `shadow` database as a secure map. Secure maps are only readable from a privileged port, thus only a privileged program could access the encrypted password. Sites that need more security than provided by NIS should use NIS+.

Creating Groups

The `admin` needs to verify on the local system and on the network that all groups have a unique group ID (GID).

Deleting Users

When an account is deleted from the system, the administrator and security administrator must take the following actions:

- The account's home directory must be deleted.
- Any processes or jobs belonging to the deleted account must be removed:
 - Any objects owned by the account must be deleted or the ownership must be assigned to another user.
 - Any `at` or `batch` jobs scheduled on behalf of the user must be deleted. See the `at(1)` and `crontab(1)` man pages, if needed.
- The user (account) name and UID must be retired and not reused.

Deleting Groups

When a local group is deleted from the system, the administrator role must ensure the following:

- All objects with the GID of the deleted group must be deleted or assigned to another group.
- All users who have the deleted group as their primary group must be reassigned to another primary group.

Administering the Maximum Allowable Number of Bad Password Entries

By default, the Trusted Solaris system allows a maximum of five failed attempts to enter the correct password during a single access attempt. If a user or role account enters the wrong password one time too many during a single attempt, the account is locked. Having such a limit helps forestall brute force attempts to gain access by guessing multiple different passwords.

A count of incorrect passwords entered during a single attempt is kept in the `flag` field of the user or role's entry in the local `shadow(4)` file or in the NIS+ table. If the user or role enters the correct password before the count exceeds the maximum, the flag is re-set to zero (0). If an account enters the wrong password one time too many during a single session, the account is locked by the insertion of the `*LK*` string into the password field of the account's entry either in the `passwd` file or in the equivalent name service database, as described in the `passwd(4)` man page.

Note - Because NIS does not make the `flag` field of the `shadow` database available, a count of failed retries cannot be maintained on a system that relies on the NIS name service. If enforcing a maximum number of failed login attempts is essential to your site's security policy, use either NIS+ or local files.

The number of retries allowed applies only for multiple bad passwords entered in sequence in either of the following two occasions:

- When logging into a host
- When re-authenticating oneself in order to change a password or to assume a role.

If an account is ever closed by inadvertent error, the security administrator role can open the account by giving the user a new password using the Password tab in the SMC `User Accounts` tool or by using the appropriate options with the `smuser(1M)` command line interface.

The security administrator can change the `RETRIES` limit system-wide and can also change whether the limit applies to all users or individual users. Following are the actions that can be taken to change the default:

- The security administrator role can change the maximum number of retries to be any number that is consistent with the site's security policy.

See “To Change the Maximum Allowable Number of Incorrect Password Entries” on page 56, if desired, for how to set the `RETRIES` value in the local `/etc/default/login` file.

- The security administrator role can specify that any individual user's account cannot be locked or can change the default system wide, so that account locking does not occur for anyone. By default, role accounts are not locked. The locking of individual role accounts can be specified using the Administrative Role tool.

See “To Prevent Account Locking for Individuals” on page 57, if desired, for how to update a user's account with the User Accounts tool to prevent the account from being locked. When the administrative role specifies `Lock account after maximum failed logins No` for a user or role account, the key value pair `lock_after_retries=no` is stored in the account's entry in the `user_attr(4)` file. See “To Prevent Account Locking for All User Accounts” on page 58 for how to specify a system-wide setting using the key value pair `LOCK_AFTER_RETRIES=no` in the `policy.conf(4)` file.

The entries for individual users take precedence over system-wide entries in `policy.conf(4)`, which in turn take precedence over the system default.

It is recommended that the accounts for administrative roles and the account of at least one user who is able to assume the security administrator role should have `Lock account after maximum failed logins` set to `No`.

▼ To Change the Maximum Allowable Number of Incorrect Password Entries

1. **Log in and assume the security administrator role.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. **At `ADMIN_LOW`, use the Admin Editor action to open the `/etc/default/login` file for editing.**

See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.

3. **Search for the string `#RETRIES`.**

```
# RETRIES sets the number of consecutive authentication failures
# allowed before the user is locked out.
#
#RETRIES=5
```

4. **Change the value set for `RETRIES` as desired and remove the `#` sign.**

5. Save and close the file.

:wq!

▼ To Prevent Account Locking for Individuals

1. **Assume the security administrator role and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Bring up the Solaris Management Console on the desired server with the desired scope, either Files, NIS, or NIS+.**
See “To Launch the Solaris Management Console” on page 39, if needed, for how to bring up the Solaris Management Console.
3. **Bring up the User Manager.**
 - a. **Double-click the Trusted Solaris Configuration icon.**
 - b. **Double-click the Users icon.**

Note - Preventing account locking is done by modifying the properties of an existing user account. If the user account does not exist, create it by performing Step 4 on page 57. If the account already exists, go directly to Step 5 on page 58 to modify its properties.

4. **Create the account.**
 - a. **To add a user account, double-click the User Accounts icon in the left-hand pane, click Add User in the Action menu, and then choose With Wizard or From Template.**
Follow the instructions in the help text for how to fill in the fields to add a user account.
 - b. **To add a role account, double-click the Administrative Role icon in the left-hand pane, and then click Add Administrative Role in the Action menu.**
Follow the instructions in the help text for how to fill in the fields in the wizard to add a role account.

5. **Access the User Properties or Administrative Role Properties tool, whichever is appropriate.**
 - a. **Double-click the User Accounts icon or the Administrative Role icon, as appropriate.**

An icon displays for each user or role account.
 - b. **Double-click the name of the user or role.**

The Properties dialog displays.
6. **Click the Trusted Solaris Attributes tab.**
7. **In the Account Usage section, select No from the pull-down menu next to Lock account after maximum failed logins.**

▼ To Prevent Account Locking for All User Accounts

1. **Assume the security administrator role and go to an ADMIN_LOW workspace.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **At ADMIN_LOW, use the Admin Editor action to open the /etc/security/policy.conf file for editing.**

See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
3. **Search for the string LOCK_AFTER_RETRIES.**

```
LOCK_AFTER_RETRIES=yes
```

4. **Change yes to no, or if the string is not present in the file, add the following.**

```
LOCK_AFTER_RETRIES=no
```

5. **Save and quit the file.**

```
:wq
```

Enabling Keyboard Aborts

In the default Trusted Solaris operating environment, the keyboard abort sequence in the `/etc/default/kbd` file is disabled:

```
KEYBOARD_ABORT=disable
```

By default therefore, Trusted Solaris systems can only be brought down by an orderly shutdown through the `Shut Down` option from the `Trusted Path` menu.

Where the site's security policy allows, the default setting can be changed. On hosts that are used by administrators for debugging, this setting can be changed to allow access to the `kadb(1M)` kernel debugger.

▼ To Enable Shutdowns Using the Keyboard Escape Sequence

- 1. Log in and assume the security administrator role.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
- 2. At `ADMIN_LOW`, use the `Admin Editor` action to open the `/etc/default/kbd` file for editing.**
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
- 3. Search for the string `KEYBOARD`.**

```
KEYBOARD_ABORT=disable
```

- 4. Enter a pound sign at the start of the line to comment it out.**

```
#KEYBOARD_ABORT=disable
```

Preventing Logins From Being Disabled After a Reboot

The use of the `/etc/nologin` file was extended to disable logins after every reboot. In both Trusted Solaris and Solaris operating environments, the `/etc/nologin` file

is created while the system is being rebooted to prevent logins, and the file is removed when booting is complete. In Trusted Solaris, the `/etc/nologin` file is created again after boot and is not removed until a user with the Enable Logins authorization enables logins.

If your site's security policy allows, the security administrator role can edit the `RMTMPFILES` script in `/etc/init.d` to comment out the lines that recreate the `/etc/nologin` file. See "To Prevent Logins From Being Disabled After a Reboot" on page 60, if changing the default is consistent with your site's security policy.

▼ To Prevent Logins From Being Disabled After a Reboot

1. Assume the security administrator role.

See "To Log In and Assume an Administrative Role" on page 32, if needed.

2. Use the Admin Editor action to open the `/etc/init.d/RMTMPFILES` for editing.

See "To Use the Admin Editor Action to Edit a File" on page 46, if needed.

Note - Do not create a backup file in the `/etc/init.d` directory. Because all files in the startup directories are executed, the backup file would be executed after the changed version, so the `/etc/nologin` file would be re-created, and the effect of this procedure would be undone.

3. Comment out the lines that disable logins after a reboot.

Comment out the active lines as shown in the following screen.

```
# cp /dev/null /etc/nologin
# echo "" >> /etc/nologin
# echo "NO LOGINS: System booted" >> /etc/nologin
# echo "Logins must be enable by an authorized user." >>
# /etc/nologin
# echo "" >> /etc/nologin
```

4. Save and quit the file.

```
!wq
```

Changing Rules for Upgrading and Downgrading Files and Selections

By default, normal users can perform cut and paste, copy and paste, and drag and drop operations on both files and selections *as long as the source and destination have the same label and have the same user ID.*

The `/usr/dt/config/sel_config` file is consulted to determine which actions will be taken when an operation would upgrade or downgrade a label. (The comments and keywords in the file use the terms sensitivity label and label interchangeably.)

Note - The rules that apply when some operations are performed on file icons differ from the rules that apply when the same operations are performed on selections made in windows. Drag and drop of *selections* always requires equality of labels and ownership.

The `sel_config` file defines:

- A list of selection types to which automatic replies are given
- Whether certain types of operation should be automatically confirmed or
- Whether a selection confirmer dialog should be displayed

The following figure shows the selection confirmer for drag and drop operations between File Managers. Other slightly-different selection confirmers display for cut and paste and copy and paste operations between File Managers and between windows at varying labels.

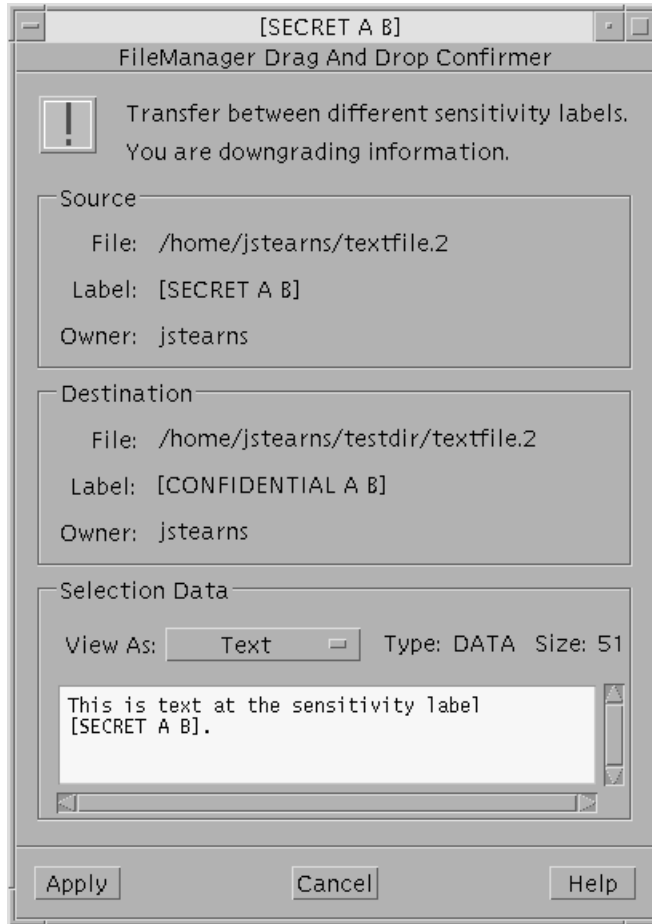


Figure 2-1 File Manager Selection Confirmer

The security administrator role can either accept the defaults or change them by using the Selection Configuration action in the System_Admin folder in the Application Manager. The action opens the `/usr/dt/config/sel_config` file for editing using `adminvi(1M)`. Any new settings become effective the next time anyone logs in.

With specific authorizations, users can perform the following types of operations:

- Cut and paste, copy and paste, and drag and drop of *files* between File Managers having the differing owners or differing labels
- Cut and paste and copy and paste (but not drag and drop) of *selections* between windows having differing owners and differing labels. Drag and drop of *selections* always requires equality of labels and ownership, the rules in the `sel_config` file do not apply to drag and drop of selections. Also, the selection confirmer does not have any rules defined for equality of labels, since the transactions are always allowed when the labels are equal.

The types of operations that may be performed on files with varying label and ownership relationships are summarized and shown with the authorizations needed, in the following table.

TABLE 2-3 Rules and Authorizations Required for File Manager Copy and Paste, Cut and Paste, and Drag and Drop

Transaction Description	Label Relationship	Owner Relationship	Authorizations
Copy and paste, cut and paste, or drag and drop of files between File Managers	Same label	Same UID	None required
	Downgrade	Same UID	Downgrade file label
	Upgrade	Same UID	Upgrade file label
	Downgrade	Differing UIDs	Downgrade file label AND Act as file owner
	Upgrade	Differing UIDs	Upgrade file label AND Act as file owner

The types of operations that may be performed on selections between *windows* with varying label and ownership relationships are summarized and shown with the authorizations needed in the following table.

TABLE 2-4 Rules and Authorizations Required for Copy and Paste, Cut and Paste and Drag and Drop of Selections Between Windows

Transaction Description	Label Relationship	Owner Relationship	Authorizations
Copy and paste, or cut and paste, of selections between windows	Same label	Same UID	None required
	Downgrade	Same UID	Paste to a downgraded window
	Upgrade	Same UID	Paste to an upgraded window

TABLE 2-4 Rules and Authorizations Required for Copy and Paste, Cut and Paste and Drag and Drop of Selections Between Windows *(continued)*

Transaction Description	Label Relationship	Owner Relationship	Authorizations
	Downgrade	Differing UIDs	Paste to a downgraded window AND Act as file owner
	Upgrade	Differing UIDs	Paste to an upgraded window AND Act as file owner
Drag and drop of selections between windows	Same SL always required	Same UID always required	None applicable

sel_config File Sections

The `sel_config` file has two sections described below:

- Automatic confirmation
- Automatic reply

The rules in the `sel_config` file apply to cut and paste, copy and paste, and drag and drop of files between file managers. (See `dtfile(1)` and the *Trusted Solaris User's Guide* for more about the File Manager application.) The rules in the `sel_config` file also apply to cut and paste and copy and paste (but not drag and drop) of selections between windows, which are mediated by the `/usr/dt/bin/sel_mgr` application.

Automatic Confirmation Section

The format of each line in the automatic confirmation section of the `sel_config` file is shown in the following table:

TABLE 2-5 Format of Automatic Confirmation Section in sel_config

Transfer Type	Automatically confirm? n= display the selection confirmer
<i>label- relation</i> (upgrade downgrade equal disjoint)	y n

label-relation refers to the relationship between the label of the source and the label of the destination

Automatic Reply Section

The `autoreply` field defines the type of reply for all the named types of selections that follow. This section provides a way to automatically reply to several types of selections at once instead of having to respond to each individually. The following table shows the default `autoreply` section with the setting of `y` for yes for the following four types of selections.

autoreply: y
replytype: TARGETS
replytype: Pixel Sets
replytype: LENGTH
replytype: Type Of Monitor

If the value is `y`, the remaining entries of the set are used as attributes for the selection data (rather than the actual contents) to complete the operation without confirmation. If value is `n` (for no), then the remaining entries are ignored. Entries can be specified for any `Type` field that appears in the Confirmer window.

▼ To Modify the Selection Configuration File

1. Assume the system administrator role and go to an `ADMIN_LOW` workspace.
2. Go to the `System_Admin` folder in the Application Manager.
3. Double-click the Selection Configuration action to open the `sel_config` file for editing with `adminvi(1M)`.

See “Changing Rules for Upgrading and Downgrading Files and Selections” on page 61 for what the fields mean, if needed.

4. After making changes, save and quit the file:

```
␣:wq
```

Note - Changes go into effect the next time anyone logs in.

Changing Configurable Trusted Solaris Kernel Switches

The following table shows the customer-configurable kernel switches in the `system(4)` file. For how to change the switches see , “To Change Configurable Kernel Switch Settings in the `/etc/system` File” on page 68.

TABLE 2-6 Customer Configurable Switches in /etc/system File

<code>tsol_admin_high_to_cipso</code>	The <code>tsol_admin_high_to_cipso</code> switch is not in the default <code>/etc/system</code> file, but it can be added if needed. The default setting in the kernel is 0. To enable communications with TSIX-type hosts that have the IP Label Field specified as CIPSO, this switch must be set to 1. This causes the label on a packet to be mapped to a valid CIPSO label with the highest classification and all compartments turned on, instead of being dropped. See “CIPSO Labels in Packets” on page 190 for more information.
<code>tsol_clean_windows</code>	To support object reuse, the <code>tsol_clean_windows</code> switch is set to 1 by default, to clear inactive register windows on return from each system call. Setting the switch to 0 disables the cleaning of inactive windows after each system call, allowing the possibility that a system call can return kernel information from an inactive register window.
<code>tsol_flush_buffers</code>	Between the time when blocks are linked to an inode and written to disk, a crash could leave old disk blocks (possibly of a higher label) linked to a file system after <code>fsck(1M)</code> recovers the file system. To ensure that data blocks are flushed before inodes are updated on disk, the <code>tsol_flush_buffers</code> switch is set to 1 by default. There is a small performance penalty. Setting this switch to 0 disables the forced data flushing before inode updates.

TABLE 2-6 Customer Configurable Switches in /etc/system File (continued)

tsol_hide_upgraded_names	<p>Actions by users with the Upgrade File Label authorization and by processes with the <code>file_mac_write</code> and <code>file_upgrade_sl</code> privileges can either create a new file or subdirectory or relabel an existing file or subdirectory at a label that strictly dominates the label of the containing directory; these files and subdirectories are said to be upgraded and the names of the upgraded files and subdirectories are referred to as <i>upgraded names</i>.</p> <p>At sites that consider upgraded names to be sensitive information, the <code>tsol_hide_upgraded_names</code> switch allows the security administrator role to configure the system so that upgraded names are hidden. Setting this flag prevents upgraded file names from being returned with <code>getdents(2)</code>. Because all directory entries must be examined before the results are returned to the calling process, there is a performance penalty. This switch is set to 0, and therefore upgraded names display by default.</p>
tsol_privs_debug	<p>The <code>tsol_privs_debug</code> switch allows the administrative use of <code>runpd(1M)</code> to characterize a program's use of privilege. Requires additional setup; for the complete procedure, see Chapter 13 under "To Find Out Which Privileges an Application Needs" on page 326. After the application(s) have been privileged debugged, this variable should be reset and the machine rebooted. This switch is set to 0; therefore, privilege debugging is disabled by default.</p>

▼ To Change Configurable Kernel Switch Settings in the /etc/system File

1. **Assume the security administrator role and use the Admin Editor action from the System_Admin folder in the Application Manager to open /etc/system for editing.**
See "To Log In and Assume an Administrative Role" on page 32, if needed.
2. **Set variables as desired.**
3. **Save and quit the file.**

lwm:

4. Shut down the system using the `Shut Down` option from the `Trusted Path` menu, and enter the `boot` command at the monitor prompt.

```
OK boot
```

Extending Extendable Security Mechanisms

The following Trusted Solaris security mechanisms are extendable:

- Audit events
- Audit classes
- Rights profiles
- Roles
- Authorizations
- Privileges

Adding audit events and audit classes is described in the *Trusted Solaris Audit Administration*. Adding rights profiles is described in “Adding or Modifying a Rights Profile” on page 144. Adding roles is described in “Creating a New Role” on page 113. The rest of this section describes how to add authorizations and privileges.

Adding New Authorizations

Adding a new authorization consists of:

- Adding an entry for the authorization into the `auth_attr(4)` database.
- Adding a grant authorization that can be used to enable a role to grant the new authorization to others into the `auth_attr(4)` database.
- Populating the new authorization entry into the `auth_attr` name service database.
- Writing or modifying an application to check for the new authorization.

The device allocation mechanism is the only existing feature of the default Trusted Solaris system that can use a new authorization, but any site can write other applications that check for new authorizations.

- Assigning the new authorization to user or role accounts, as desired

The example in this section makes use of the fact that the device allocation authorization is not hard-coded; any desired authorization can be specified in each device's entry (which is stored in the `device_allocate(4)` file), and then in order to allocate a device a user is required to have the specified authorization.

The format for an entry in the `auth_attr(4)` file is:

```
name:res1:res2:short_desc:long_desc:attr
```

The `short_desc` field is a brief description of the activity permitted by the authorization. The `long_desc` is used by the Solaris Management Console when it displays authorizations. A help file, which is specified in the `attr` field using the keyword value pair `help=filename`, displays in the online help. The help file must be located in this directory ending with the name of the locale: `/usr/lib/help/auths/locale/localename`.

The following screen shows the default device allocation authorization in the `auth_attr` file in the C locale. The help file in the C locale is under `/usr/lib/help/auths/locale/C`.

```
solaris.device.allocate::Allocate Device::help=DevAllocate.html
```

The example below shows two finer-grained device allocation authorizations that could be used to replace the default one above, one for tape devices and one for floppy devices. In the example, the authorizations' names start with the Internet domain name of the NewCo company.

```
com.newco.device.allocate.tape::Allocate Tape Device::help=TapeAllocate.html  
com.newco.device.allocate.floppy::Allocate Floppy Device::help=FloppyAllocate.ht  
ml
```

The next example shows the `solaris.allocate.device` authorization replaced in the `device_allocate(4)` file entry for `floppy_0` with `com.newco.device.allocate.floppy`. This change would be made using the Device Allocation Manager, Device Administration dialog, as described in "To Add An Authorization" on page 71. After this substitution, any user attempting to allocate the floppy device must have the new authorization.


```
com.newco.grant:::Grant All NewCo Authorizations:::help=GrantNewco.html
com.newco.grant.device:::Grant NewCo Device Authorizations:::help=GrantNewcoDevice
.html
com.newco.device.allocate.tape:::Allocate Tape Device:::help=TapeAllocate.html
com.newco.device.allocate.floppy:::Allocate Floppy Device:::help=FloppyAllocate.ht
ml
```

4. Save and close the file.

```
:wq
```

5. If you are using a naming service, update the `user_attr(4)` NIS map or NIS+ table.

See the `nistbladm(1)` man page and the *Solaris Naming Administration Guide* for how to update the `user_attr(4)` map or table.

6. Add the authorization to the database that defines which authorization are used by the application.

For example, to assign the new device allocation authorization, you would use the Device Allocation: Configuration dialog, as described in “To Add or Configure a Device” on page 293. For example, if the administrator assigns the new `com.newco.device.allocate.floppy` authorization to a floppy disk device, any user attempting to allocate the floppy device must have the `com.newco.device.allocate.floppy` authorization.

As described in Chapter 12, the Device Allocation Manager looks in the `device_allocate(4)` file to find out which authorizations are required for allocating devices.

7. Use the SMC Rights tool to add the new authorizations to the `Custom_rolename_Profile`, and make sure the `Custom_rolename_Profile` is assigned to the role.

Extending Privileges

Adding a new privilege consists of adding an entry for the privilege into these two files:

- `priv_names.h`

Pathname: `/usr/include/sys/tsol/priv_names.h`

■ priv_name

Pathname: /usr/lib/tsxol/locale/C/priv_name

priv_names.h

The /usr/include/sys/tsxol/priv_names.h header file contains manifest constants and associated numbers for privileges. Up to 128 possible privileges are allowed. As shown in the following screen example, the definitions for the default privileges range from 1 to 85 (with 0 meaning no privileges). Not all 85 privileges are defined since some have been retired.

The manifest constants and numbers for default privileges in priv_names.h are:

```
PRIV_FILE_AUDIT = 1, /* operational */
PRIV_FILE_CHOWN = 2, /* operational */
PRIV_FILE_DAC_EXECUTE = 3, /* policy */
.
.
.
PRIV_WIN_SELECTION = 84, /* operational */
PRIV_WIN_UPGRADE_SL = 86, /* operational */
```

As shown in the next example, the privileges identified with the reserved name are available for any site to extend.

Privileges available for extension are:

```
/* Reserved for ISV, GOTS, integrator, ... use */

reserved90 = 90,
reserved91 = 91,
reserved92 = 92,
.
.
.
reserved126 = 126,
reserved127 = 127,
reserved128 = 128
```

Note - If you wish to interoperate with other systems, try to reserve a privilege before you start by contacting your Trusted Solaris representative.

priv_name(4)

The format for an entry in `/usr/lib/tsol/locale/locale_name/priv_name` is as follows:

constant:name:description

In the `priv_name(4)` file, the constant field must have exactly the same manifest constant name that was assigned to the privilege in the `/usr/include/sys/tsol/priv.h` file. The name must be concise and descriptive for display in user interfaces.

The description field describes the activity permitted by the privilege. The definition is used wherever the privilege chooser comes up, to guide the security administrator role when assigning privileges to programs.

Example 2-1 gives an example of a privilege in the default `priv_name` file. Note that the manifest constant name is in all capital letters, the name is in lowercase letters, and the description is continued from line to line with the backslash character (`\`).

EXAMPLE 2-1 Definition for the `file_audit` privilege in the `priv_name` File

```
PRIV_FILE_AUDIT:file_audit:Allows a process to get or set a file's or \
directory's audit preselection information. The auditing preselection \
information may override the preselection information associated with \
a process' access to a file or directory. \
Allows a process to get or set a file's or directory's public object \
flag. The public object flag may override the successful read/search \
access preselection information associated with a process' access to \
a file or directory.
```

▼ To Add a Privilege

Note - If possible, before you start, contact your Trusted Solaris representative to reserve a privilege number.

- 1. Log in and assume the security administrator role.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
- 2. At `ADMIN_LOW`, use the Admin Editor action to open the `/usr/include/sys/tsol/priv_names.h` file for editing.**
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
- 3. Read the comment at the top of the `priv_names.h` file.**
The comment is shown in the following figure.

```

/ *
* ***** IMPORTANT *****
*
* The privilege names should be maintained in alphabetical order
* not numeric order.
*
* When a privilege is retired it should be placed in the appropriate
* reserved area in the form "tsol_reserved## = ##," or
* "reserved## = ##".
*
* When a new privilege is needed, it should be taken from the first
* available privilege in the appropriate reserved area.
*
* ISVs, GOTS', integrators who need privileges are encouraged to
* request and retire them by contacting their respective Trusted
* Solaris support representative.
*
* This file is parsed by the priv_to_str(3) functions.
*
* In order to guarantee correct parsing, the format of the
* following priv_t definition must be preserved.
*
* Specifically, the following guidelines must be followed:
*
* 1. All privileges must have an explicitly assigned id.
*    DO NOT RELY ON COMPILER TO ASSIGN IDs.
*
* 2. One privilege id assignment per line.
*    DO NOT CONCATENATE OR BREAK LINES.
*
* 3. Do not use the '=' character at anywhere other than
*    the privilege id assignment.
*    For example, DO NOT use '=' in the comments.

```

4. Create an entry in the `priv_names.h` file with the manifest constant for the privilege.

A sample entry is below.

```
PRIV_RISKY = 90,
```

5. Save and close the file.

```
:wq
```

6. Use the Admin Editor **action to open the** `/usr/lib/tsxol/locale/locale_name/priv_name` **file for editing.**

For example, for the C locale, you would edit the `/usr/lib/tsxol/locale/C/priv_name` file.

7. **Create an entry with the manifest constant, name, and definition for the privilege in the** `priv_name` **file.**

Note - Make sure that the first field has the same manifest constant defined for the privilege in the `priv_names.h` file.

A sample entry is below.

```
PRIV_RISKY:override everything:Allows a process to bypass all MAC and \
DAC checks and auditing flag settings and be otherwise totally \
unaccountable.
```

8. **Save and close the file.**

```
!wq
```

9. **Copy the changed** `priv_names.h` **and** `priv_name` **files or make the same change in these files on all computers in the Trusted Solaris system.**

Getting a Hexadecimal Equivalent for Labels and Clearances

Use `atohexlabel(1M)` with the options shown in the following steps to get the hexadecimal equivalents of a label or clearance. The `atohexlabel(1M)` command is in the default Object Label Management profile, which is assigned by default to the security administrator role.

▼ To Get a Hexadecimal Equivalent for a Label

1. **Login and assume the security administrator role.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **In an ADMIN_HIGH workspace, bring up a terminal.**
3. **To get the hexadecimal value for a sensitivity label, use `atohexlabel` with the `-s` option followed by the name of the sensitivity label.**

```
$ atohexlabel -s SECRET  
0x00060c0000000000000000000000000000000000000000000000003fffffffffff0000
```

4. **To get the hexadecimal value for a clearance label, use `atohexlabel` with the `-c` option followed by the name of the clearance.**

```
$ atohexlabel -c SECRET  
0x00060c0000000000000000000000000000000000000000000000003fffffffffff0000
```

Changing the Front Panel

Anyone can drag and drop a pre-existing action from the Application Manager to the Front Panel as long as the account doing the modification has the action in its profile. Only actions in the `/usr/dt/*` or `/etc/dt/*` directories can be added to the Front Panel, while applications in the `$HOME/.dt/appconfig` directories cannot be added. Users can use the Create Action action, but the user cannot write into any of the directories where the system-wide actions are stored, so new actions cannot be made available system-wide.

In Trusted Solaris, the actions’ search path has been changed so that actions in any individual’s home directory are processed last instead of first. Therefore, no one can customize existing actions.

The security administrator role has the Admin Editor action, and the security administrator role should make any needed modifications to the `/usr/dt/appconfig/types/C/dtwm.fp` file and the other configuration files for the Front Panel subpanels. This manual contains two procedures that exemplify how to modify existing files to create new actions. “To Create a New Administrative Action

for Editing an Administrative File” on page 323 describes how one to create an alternate mail application that can run with privilege in the `Front Panel`. “Substituting an Alternate Mail Application” on page 166 describes how to add an administrative action that can run with inherited privileges to the `System_Admin` folder for the purpose of editing another configuration file.

Roles can drag and drop actions from the `System_Admin` folder to the `Front Panel`, but normal users may be confused because only the roles can use these actions.

Changing the Workspace Menu

The workspace menu is the menu accessed by clicking and holding the right mouse (Menu) button on the background of the workspace. Using the `Customize Menu` and `Add Item to Menu` options on the `Workspace Menu` is the same as in the base CDE window system, with some Trusted Solaris caveats having to do with labels, MAC, MLDs, and the profile mechanism.

The following apply when a user is allowed to work at multiple labels:

- Changes to the `Customize Menu` and `Add Item to Menu` options need to be made at the session clearance.



Caution - Changes made at other labels than the session clearance are not recognized by the window system.

- Changes can only be made in a normal user workspace.
- Changes to the `Workspace Menu` persist when the user assumes a role.
- Changes made to the `Workspace Menu` are stored in the user’s MLD home directory in the SLD created at the working label (which should be the session clearance).
- If a user is able to log in at multiple labels, the user has the potential for multiple session clearances during different login sessions. Therefore, make any changes at each of the potential session clearances if you want the changes to apply to all potential login sessions.
- The items in the workspace menu are stored in the `.dt/wsmenu` directory within the user’s MLD home directory in the SLD that corresponds to the working label.

For example, to change the `Workspace Menu` when the user’s only possible session clearance is `NEED_TO_KNOW ENG`, the user would go to a workspace at the `NEED_TO_KNOW ENG` label. If the user adds an item to the `Applications` menu using the `Add Item to Menu` option, the item would be stored in:

```
/home/username/.dt/wsmenu/Applications
```

The directory above corresponds to the real MLD path shown below, where `.SLD.3` in the example is the SLD that corresponds to the `NEED_TO_KNOW ENG` label for user `gfaden`.

```
/home/.MLD.gfaden/.SLD.3/.dt/wsmenu/Applications
```

Another caveat is connected to the profile mechanism:

- Any option added to the workspace menu must be handled by one of the user's execution profiles or the option will fail when invoked and an error message will display.

For example, anyone with the Run action can double click on the icon for any executable and run it, even if the action or any commands it invokes are not in one of the account's execution profiles. By default, roles do not have the Run action, and all executable actions require the Run action, and therefore, any item that requires the Run action fails when executed by a role.

To Change the Workspace Menu

Note - To make the changes apply to every possible login session, users with multiple labels need to repeat these steps at every label that corresponds to a potential session clearance.

1. **Log in and go to a workspace whose label is the same as the session clearance—without assuming a role.**
2. **Choose the `Customize Menu` or `Add Item to Menu` option from the `Workspace Menu` and make any desired changes.**

Managing User Accounts

In the *Trusted Solaris Installation and Configuration* manual, a limited number of user accounts are set up to assume the recommended roles that are needed for initial configuration of the Trusted Solaris system. This chapter describes essential decisions that the roles should make before configuring user accounts. This chapter also covers other tasks for planning and setting up for all the remaining users on the system that are not covered in the *Trusted Solaris Installation and Configuration* manual. This chapter gives additional background needed for ongoing management of user accounts.

- “Before Setting Up User Accounts” on page 82
- “Making Decisions About User Accounts” on page 83
- “Default User Security Attributes” on page 85
- “Precedence Relationships for Attributes” on page 88
- “Managing Remote Logins” on page 90
- “Managing Initialization Files” on page 91
- “Accessing All Bundled Man Pages” on page 97
- “Using `.copy_files` and `.link_files`” on page 97
- “Administering the Automatic Running of Jobs Using `cron`, `at`, and `batch`” on page 99

This chapter includes the following procedures:

- “To List All the Roles” on page 104
- “To List All Rights Profiles” on page 105
- “To Set Up System-Wide User Attributes” on page 106
- “To Make `.login` or `.profile` Looked at During Login” on page 107
- “To Force `dterm` to Launch New Shells as Login Shells ” on page 108
- “To Propagate Startup Files to Everyone’s Home Directory SLDs” on page 109

Before Setting Up User Accounts

Have the following information available:

- A list of the available rights profiles

Following are the ways you can view the rights profiles.

- You can view the rights profiles in the SMC Users' Rights tool, but there is no way to save the information from the Rights dialog to a file.
- You can search the two files where profiles are defined: `prof_attr(4)` and `exec_attr(4)`, but it may be difficult to get a clear definition of the profiles from looking at the two files.
- You can use the `smprofile(1M)` command and some of its options to list profiles. See "To List All Rights Profiles" on page 105.

You cannot use the `more(1)` command with `smprofile`, and you cannot redirect the output into a file because `smprofile` needs you to enter a password on the command line. To get a manageable amount of output, it is a good idea to first run `smprofile list` with the options that list all the profiles or refer to "Rights Profile Descriptions" in *Trusted Solaris Administration Overview*. After you have a list of profiles you can then run `smprofile list` with `-l` and `-n profile_name` to get the list of programs and security attributes for one profile at a time.

If additional rights profiles have been created at your site, see your own internal documentation for a description.

- A list of the available roles.

The `smrole(1M)` command with the `list` option lists all the roles. See "To List All the Roles" on page 104. If your site has added or modified roles, see your internal documentation for a description of the roles defined for users at your site.

- A list with the name of each person who needs an account, along with the responsibilities, roles, clearances and minimum labels assigned to each.

Getting this information together may simply involve collecting data on the functions that each person performs in your organization and deciding which labels they need to work at and the rights profiles each should have. You also need to decide who is going to be able to assume roles. At government organizations, you may need to review the government clearances that have been given to each person, and use these to determine which roles each person may assume and the labels at which he or she may work.

Making Decisions About User Accounts

Make the following decisions before starting because they affect how you configure user accounts. Some decisions are the same as those you would make when installing a network of standard Solaris or other UNIX machines. However, because you are configuring the Trusted Solaris environment, make all decisions in the light of implications they might have for your site's security and of any special requirements you might have.

- Decide whether to accept or change the system-wide settings in the `audit_control(4)`, `policy.conf(4)` and the `label_encodings(4)` files that together define default Trusted Solaris security attributes for all accounts. See “Default User Security Attributes” on page 85.
- Decide whether to use the wizard or a template to add user accounts.

You can create new accounts using either of the following:

- The User Accounts tool's Action menu options:
 - Add User->With Wizard or
 - Add Multiple Users->With Wizard

The wizards allow the system administrator role to specify only a small set of attributes when creating a new user account.

- The User Accounts tool's Action menu options:
 - Add User->From Template or
 - Add Multiple Users->From Template

Before adding users with a template, you first create the template using the User Templates tool's Action menu option:

- Add User Template option

Because the User Accounts wizards do not allow you to specify many attributes, most administrators would probably need to change some of the default attributes the wizard assigns. For example, you cannot specify a login shell, which defaults to the Bourne shell.

Creating one or more User Template(s) is a recommended practice because you can use a template to set a reasonable set of defaults for all new user accounts. Once the template is created, its defaults combine with the Trusted Solaris default attributes (described in “Default User Security Attributes” on page 85) to allow you to create users whose attributes suit your preferences and your site's security policy.

If the system-wide settings and template are configured properly, the following benefits occur:

- Adding users can be done by the System Administrator role without the intervention of the Security Administrator role.
- The System Administrator role can change most attribute assignments (except for the Trusted Solaris attributes: roles, profiles, auditing, and the labels and account-availability options on the Trusted Solaris Attributes tab).
- The Security Administrator needs to get involved only if individual users need non-standard attributes.

For example, the Security Administrator is the only role that can assign roles or profiles or a non-default set of Trusted Solaris security attributes to an individual user account.

See Chapter 5 for details about the attributes of user and role accounts. Also review the user and role-specific help topics from the SMC Help menu.

- Decide how to handle email that is sent at an label below the recipient's minimum sensitivity label.

Make sure that the Trusted Solaris-specific privacy options in the `sendmail(1M)` configuration file `sendmail.cf` have values consistent with your site's security policy.

- Decide whether to allow remote logins.

As described in "Managing Remote Logins" on page 90, certain remote logins require an authorization. Decide if you wish to allow users to remotely log in where an authorization is required, and if you so decide, assign the needed authorization to accounts.

- Decide whether to allow sourcing of shell initialization files and whether you want to control the initial contents of the files

See "Managing Initialization Files" on page 91.

- Decide which files, if any, should be copied or linked from the minimum-label home directory SLD created for a user into subsequent SLDs, and then modify the `.copy_files`, and `.link_files` in `/etc/skel` or an alternate skeleton directory.

See "To Propagate Startup Files to Everyone's Home Directory SLDs" on page 109.

- Decide whether accounts should be locked if a bad password is entered more than the default maximum number allowed and whether to accept or change the maximum number.

The system default is 5. "Administering the Maximum Allowable Number of Bad Password Entries" on page 55 describes the Trusted Solaris mechanism for locking accounts after a certain number of failed passwords have been entered during a single attempt to log in. "To Prevent Account Locking for Individuals" on page 57

also describes how the security administrator role can change the setting for the number of failed attempts that cause an account to be locked.

- Choose the overall method of password generation, automatic or manual
- Decide which existing groups to use and whether to create new groups.

Review default groups and decide whether to add any groups. For each group, use a unique name and GID that is not duplicated anywhere on the network and that has not been used previously on the system. Group names and GIDs need to be unique to ensure traceability of activities back to a specific group.

To define new groups, the administrator role uses the `Group Manager` from the `Solaris Management Console` action. No new fields have been added for Trusted Solaris group administration.

- Decide which device are allocatable for any users, if any.
- If any devices are allocatable, decide which users can allocate them and which authorizations to use.

See “Understanding the Device Administration Dialog” on page 277 for what you need to know about assigning device-related authorizations and making devices available to users.

- Decide whether devices should be deallocated when the user who allocated them logs out or when the system is rebooted.

See “Deallocation Options” on page 282 for more information

- Decide whether to control a device differently when it is allocated from the trusted path than when they are allocated outside the trusted path.

“For Allocations From: Trusted Path or Non-Trusted Path” on page 279

- Decide whether to allow users to see the names of files whose labels have been upgraded.

An authorized can create a file that has a higher label than the directory it contains. See “Changing Configurable Trusted Solaris Kernel Switches” on page 66 for how to change the `tsol_hide_upgraded` names switch that controls whether the names of upgraded files are visible.

Default User Security Attributes

Settings in these `audit_control(4)`, `policy.conf(4)` and the `label_encodings(4)` files together define default Trusted Solaris security attributes for all user accounts. The specifications in these files combine with the values that can be specified in a user template to define a full set of attributes for all user accounts. Some of the values set in these files also apply to role accounts. The default values are described in detail in “Adding or Modifying a User Account” on page 121 and “Adding or Modifying a Role Account” on page 136. See “Precedence

Relationships for Attributes” on page 88 for how the values obtained from the various sources of these attributes are combined.

Audit Control File Defaults

The security administrator role can use the Audit Control action to edit the local `audit_control(4)` file in `/etc/security` on each computer to specify machine-wide auditing. As described in the `audit_control` man page, audit flags can be set in the `flags:` line to specify which classes of events to audit for both user and role accounts. A class can be audited for failure only, for success only, or for both success and failure. The default `audit_control` file does not specify any audit flags. See the `audit_class(4)` man page for a description of the file where audit classes are defined and see the precedence relationships described in “Precedence of `audit_control` and Related Attribute Sources” on page 89.

Label Encodings File Defaults

The following table shows the settings in the default `label_encodings` file that define the Minimum Label, Clearance, and Default Label View that are applied to a user account if the attributes are not otherwise explicitly specified for the account. The values shown are those in the installed version of the `label_encodings` file, which is usually replaced during system configuration with a version that has the site’s own labels. (See “How to Install a Label Encodings File” in *Trusted Solaris Installation and Configuration* and “Readying the Label Encodings File Before the NIS+ Master or Standalone System is Configured” in *Trusted Solaris Label Administration*. The keywords shown in the following table can be specified in the optional `LOCAL DEFINITIONS` section of the `label_encodings` file. If the keywords are not defined, then a user’s minimum label defaults to the mandatory `minimum sensitivity label=` value, and a user’s clearance defaults to the mandatory `minimum clearance=`, both of which are defined in the `ACCREDITATION RANGE` section. The Default Label view defaults to External. See the precedence relationships described in “Precedence of `label_encodings` and Related Attribute Sources” on page 89.

TABLE 3-1 Account Security Attributes Defined in the Default label_encodings

Trusted Solaris Attribute	Keyword in LOCAL DEFINITIONS Section	Default
Minimum Label	Default User Sensitivity Label= u;	In ACCREDITATION RANGE Section: minimum sensitivity label=u;
Clearance	Default User Clearance= c;	In ACCREDITATION RANGE Section: minimum clearance= c nationality: cntry1/cntry2;
Default Label View	Default Label View is External;	Not applicable. System default is external.

policy.conf File Defaults

The following table shows the default settings in the `policy.conf` file. See the precedence relationships described in “Precedence of `policy.conf` and Related Attribute Sources” on page 88.

TABLE 3-2 User Attributes and Defaults in `policy.conf`

Attribute	Keyword with Default Setting	System Default
authorizations (from <code>auth_attr(4)</code> database)	#AUTHS_GRANTED=	none
idle action: logout lock	IDLECMD=lock (applies to users only)	lock for users, do not lock for roles
idle time: 1 - 120 minutes or Forever	IDLETIME=30 (applies to users only)	30 for users; Forever for roles
show or hide labels: hidesl showsl	LABELVIEW=showsl	showsl for all
lock after bad password limit is exceeded: yes no	LOCK_AFTER_RETRIES=yes (applies to users only)	yes for users, no for roles

TABLE 3-2 User Attributes and Defaults in `policy.conf` (continued)

Attribute	Keyword with Default Setting	System Default
method of password generation: manual auto	PASSWORD=manual	manual for all
profiles (from <code>prof_attr(4)</code> database)	PROFS_GRANTED=Basic Solaris User (applies to users and roles)	none

Any authorizations and profiles defined in this file are in effect for all accounts in addition to any authorizations and profiles assigned to individual accounts.

Precedence Relationships for Attributes

Precedence of `policy.conf` and Related Attribute Sources

The precedence relationship between the values stored in the `user_attr` database, the values stored in the `policy.conf` file, and system defaults shown below applies to all the attributes in `policy.conf` except for authorizations and profiles. The authorizations and profiles apply to both user and role accounts.

- Is the attribute assigned already in the `user_attr` database entry for the user account?
 - If yes, use the attribute from `user_attr`.
 - If no, is the attribute specified in `policy.conf`?
 - If yes, use the attribute specified in `policy.conf`.
 - If no, use the system default.

Note -

Precedence of label_encodings and Related Attribute Sources

- Is the attribute assigned in the `user_attr` database entry for the account?
 - If yes, use the attribute from `user_attr`.
 - If no, is the attribute specified in the `label_encodings` file?
 - If yes, use the attribute specified in the `label_encodings` file.
 - If no, use the system default.

The label and clearance default to the mandatory settings for the minimum sensitivity label and minimum clearance in the ACCREDITATION RANGE section. If no value is set for the LABEL VIEW in the LOCAL DEFINITIONS section, the system default is used.

Precedence of audit_control and Related Attribute Sources

The security administrator can fine-tune auditing for individual accounts, if desired, using the Audit Tab in the User Account's Properties dialog box. The Audit dialog allows the security administrator role to specify which audit classes should always be audited and should never be audited for the user. Classes can be specified to be audited or never audited depending on whether they succeed or fail, as shown in the following table:

Always Audit	Never Audit
<ul style="list-style-type: none">■ For both success and failure■ For success only■ For failure only	<ul style="list-style-type: none">■ For both success and failure■ For success only■ For failure only

The settings made in the Audit dialog are stored in the account's entry in the `audit_user(4)` database. (The `audit_user` database may be administered as a local file in `/etc/security` or by NIS or NIS+.)

When a user logs in or assumes a role, any system-wide audit flags from the `audit_control` file are combined with any audit flags assigned to the account in the `audit_user` file to form an audit preselection mask:

$$(\text{flags in audit_control} + \text{user's always-audit flags}) - \text{user's never-audit flags}$$

If desired:

- Set up system-wide auditing by specifying flags in the `audit_control` file to be audited for both success and failure, for success only, or for failure only.

- Set up auditing for individual users by choosing flags to be always audited or never audited for both success and failure, for success only, or for failure only.

The precedence relationship between the various sources for auditing is:

- Are audit flags specified in either the `audit_user` database entry for the account or in the `audit_control` file or in both?
 - If yes, combine the audit flags from both sources or use the audit flags from the single source
 - If no, do not audit user actions.

See *Trusted Solaris Audit Administration* for more about how to set up auditing.

Managing Remote Logins

When a remote login is allowed under the certain conditions (described in the following list) between two Trusted Solaris hosts, the remote login is considered to be an extension of the current login session, and an authorization is not required.

The types of remote logins not requiring an authorization are:

- Use of the `rlogin` command when the user is not prompted for a password

The user is not prompted for a password when either an `/etc/hosts.equiv` file or a `.rhosts` file in the user's home directory on the remote host lists either the username or the host from which the remote login is being attempted. (See the `rhosts(4)` and `rlogin(1)` man pages for more information.)

If desired, the security administrator role in the default Trusted Solaris system can use the `Admin Editor` action to create an entry in an `rhosts` file to allow users to log in without a password. Any user who is able to use a text editor can create a similar entry in a `hosts.equiv` file in that user's home directory.

- Use of the Remote Login option from the CDE login screen

For all other remote logins, including logins with the `telnet(1)` command, the `remote login` authorization is required.

Remote logins requiring an authorization are shown in the following list:

- When `rlogin` is used, if the user is prompted for a password.

A user is not prompted for a password when either an `/etc/hosts.equiv` file or a `.rhosts` file in the user's home directory on the remote host lists either the username or the host from which the remote login is being attempted. (See the `rhosts(4)` and `rlogin(1)` man pages for more information.)

- When `telnet` or `ftp(1)` or other remote access commands are used and the user is required to enter a password.

See “Specifying an Authorization for a User or Role” on page 102 for how to assign an existing profile or create a new profile that includes the `Remote Login` authorization.

Managing Initialization Files

Administrators who are setting up shell initialization files must consider certain details that are either not as important in standard UNIX systems or do not apply. The differences exist because of the following aspects of the Trusted Solaris implementation:

- Home directories are multilevel directories.
- A profile shell can be used to restrict an account’s access to commands.
- While a role workspace is being created, any commands that are executed from an account’s `profile(4)` file are not restricted by the profile shell mechanism.

For that reason the execution of the `profile` file is restricted in Trusted Solaris.

Any files in the skeleton directory are copied into the first home directory SLD created at the account’s minimum label. The user or role can then modify the files.

The security administrator needs to set up skeleton file entries for accounts that work at multiple labels to enable the following:

- The appropriate initialization files can be copied into each new home directory SLD that is created at each label.
- The `adminvi(1M)` command can be aliased to `vi(1)` for roles.

A default `.profile` file that is copied into the role’s home directories has a function to alias `adminvi(1M)` command to `vi(1)`, but because `dtterm` does not execute shells as login shells, the `.profile` file is not read. The security administrator who wants to put the aliasing into effect for roles should do the procedure to set up `/etc/skel` files to copy the `.Xdefaults-hostname` file into home directory SLDs created on behalf of roles.

- The `updatehome` command has been created to read two new files to link or copy initialization files from the SLD that is created in a user or role’s home directory MLD at the account’s minimum label. When a user is allowed to work at more than one label, some work must be done by the security administrator role or by the individual user to ensure that shell initialization files are copied into each subsequent SLD created on behalf of the account at another label.

This section provides the background information needed to understand how startup files are administered in the Trusted Solaris environment and provides procedures

for doing the setup. Also see the man pages for the `cs(1)`, `ksh(1)`, `sh(1)`, or `pfexec(1)`.

The word *source* is often used as a verb to mean *to read in* or *to execute* the commands in a startup file. (The `cs` even has a built-in `source` command for executing the commands in dot scripts.) A set of startup files is sourced by the window system as it comes up. Which startup files are read depends on the login shell that was assigned to the user when the user's account was created. See the following table.

The `.profile` or `.login` files are only if the shell is identified as the account's login shell. The shell is invoked with a prefix of `--` (for example: `-- cs`) to indicate the shell is the login shell. This means, for example, that when a C shell is started using `cs` (without a `--` prefix), the `.login` file is not executed.

TABLE 3-3 Startup Files Read by the Window System for Each Type of Login Shell

Login Shell	Startup File Read by Window System
C shell	<code>/etc/.login</code> and <code>\$HOME/.login</code>
Bourne shell or Korn shell	<code>/etc/.profile</code> and <code>\$HOME/.profile</code>
Profile Bourne, Profile C, or Profile K shell	<code>/etc/.profile</code> and <code>\$HOME/.profile</code>

Another set of startup files is read whenever a user brings up a shell in a terminal emulator, such as the `cmdtool`, `shelltool`, or `dtterm` (see “Controlling Which Startup Files Are Read When a Shell Comes Up” on page 94).

Controlling Which Startup Files Are Read by the Window System

In the extended Trusted Solaris CDE window system, as in the base CDE window system, accounts get an editable `$HOME/.dtprofile` file whose basic job is to control whether the `.login` or `profile` files are read by the desktop when the account logs in and starts a session (see also the man pages for `login(1)`, and `profile(4)`). One exception is that when an account has a profile shell, `pfexec(1)`, as its login shell, the `.dtprofile` file is handled in a different manner, which is described in “How the Reading of Start Up Files is Controlled for the Profile Shell User” on page 94.

dtprofile Files

In the Trusted Solaris system, by default the `.login` or `.profile` files are not sourced by the window system. The sourcing of these files is controlled by one of several possible `dtprofile` files.

One of the following is copied into each account's `$HOME/.dtprofile`:

- An `/etc/dt/config/sys.dtprofile` file that was created by the site's security administrator role, if the file exists, or
- The default `/usr/dt/config/sys.dtprofile`

The following figure illustrates how `$HOME/.dtprofile` is installed.

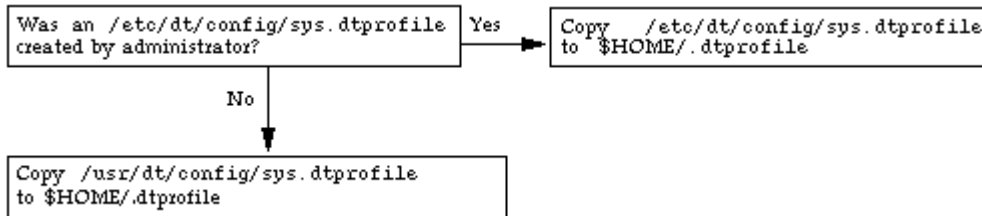


Figure 3-1 How `$HOME/.dtprofile` is installed

In the default `/usr/dt/config/sys.dtprofile`, the variable that enables the sourcing of either file is commented out. See the following figure.

```
# DTSOURCEPROFILE=true
```

Removing the `#` before the `DTSOURCEPROFILE` definition in any of the versions of the `sys.dtprofile` file causes the appropriate startup file to be read by the window system. A `*.dtprofile` file can also potentially be modified to do the same types of things done by other startup files, such as setting environment variables and search paths for commands and actions, changing where the standard error and standard out are written, and invoking commands or functions.

Comments in the default `*.dtprofile` file discourage the site's administrator or individual users from allowing the startup files to do either of the following types of actions:

- Anything that requires a terminal emulator
- Anything that requires user interaction while the window system is coming up

See the comments in the `/etc/dt/config/sys.dtprofile` file and “To Make `.login` or `.profile` Looked at During Login” on page 107, if changing the default is consistent with your site's security policy.

Note - If any modifications to a `.login` or `.profile` accidentally prevent the user from logging in, the user may use the Failsafe Session option on the login dialog. Failsafe Session allows a log in without reading any startup files—to enable the user to fix the problem file.

How the Reading of Start Up Files is Controlled for the Profile Shell User

The algorithm used for reading `dtprofile` files when an account has a profile shell as its login shell is different from the algorithm that is used when another type of shell is specified. The following algorithm prevents an account from being able to cause commands that are not in one of the account's rights profiles to be launched before the profile shell is in effect.

When a user's login shell is specified as the `pfsh`, `pfssh`, or `pfksh`, a personal `.dtprofile` that may exist in the account's home directory will never be looked at. Either the default `/usr/dt/config/sys.dtprofile` or a version modified by the security administrator role in `/etc/dt/config/sys.dtprofile` is used instead of `$HOME/.dtprofile` file.

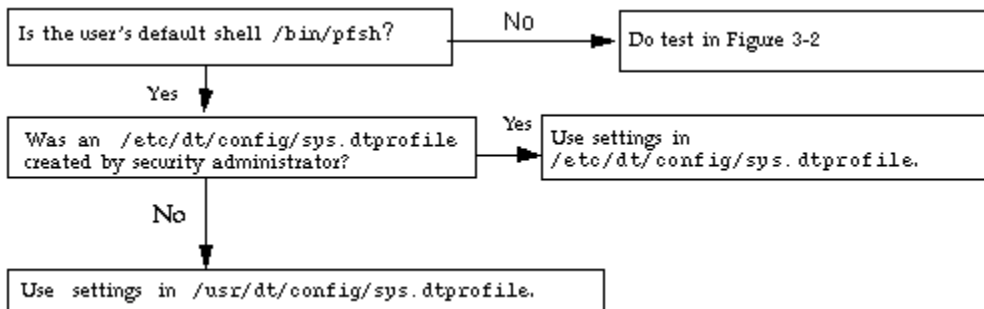


Figure 3-2 How `$HOME/.dtprofile` is Bypassed When Users have a Profile Shell

Controlling Which Startup Files Are Read When a Shell Comes Up

As in the base Solaris system, shell initialization files are used to set search paths and other environment variables and to execute some useful commands and functions. The following table shows which startup files are read by default when each type of shell is launched.

TABLE 3-4 Startup Files Read at Shell Initialization

Shell	Startup File
C shell	<code>\$HOME/.cshrc</code>
	<code>\$HOME/.login</code>
Bourne shell	<code>\$HOME/.profile</code>
Korn shell	<code>\$HOME/.profile</code>
	file specified with ENV variable
Profile Bourne, Profile C, or Profile K shell (see <code>pfexec(1)</code> man page)	<code>\$HOME/.profile</code>

The `.profile` or `.login` files are invoked only if the shell is identified as the account's login shell. The shell is invoked with a prefix of `--` (for example: `-- csh`) to indicate the shell is the login shell. This means, for example, that when a C shell is started using `csh` (without a `--` prefix), the `.login` file is not executed.

Forcing dtterm to Source `$HOME/.login` or `.profile`

Any shell started by `dtterm(1)` is not launched as a login shell, and so the `$HOME/.login` and `$HOME/.profile` files are not read. To cause `dtterm` to launch a login shell, any account, user or role, can create the following entry in the `$HOME/.Xdefaults-hostname` file :

```
Dtterm*LoginShell: true
```

Logging out is required to put the change into effect. See “To Force `dtterm` to Launch New Shells as Login Shells ” on page 108. The same entry must be made in an `.Xdefaults-hostname` file in every home directory SLD at every label at which the account works. To automate the copying or linking of `.Xdefaults-hostname` into all SLDs, see “Using `.copy_files` and `.link_files`” on page 97.

Note - The default `.profile` file for all roles contains a function to alias the `adminvi(1M)` command to `vi(1)`, but the function does not take effect unless the `Dtterm*LoginShell: true` entry is made in the `$HOME/.Xdefaults-hostname` file. See “Aliasing `vi` to `adminvi`” on page 114.

Administering Skeleton Directories

The default skeleton path used by the User Manager is `/etc/skel`. By default, a set of initialization files for each of the shells are copied from `/etc/skel` into an account's `$HOME` directory and renamed.

The `local.cshrc`, `local.login`, and `local.profile` files, shown in the `/etc/skel` directory listing in the following figure, are copied for normal user accounts.

CODE EXAMPLE 3-1 Contents of the Default `/etc/skel` Directory

```
trusted% cd /etc/skel
trusted% ls -R
local.cshrc local.login local.profile tsol/
tsol:
role.link_files role.profile
```

(The files in the Trusted Solaris-specific `/etc/skel/tsol` directory are explained in “Role Startup Files in `/etc/skel/tsol`” on page 96.)

The `/etc/skel` files are copied into the SLD that corresponds to an account's minimum label.

Role Startup Files in `/etc/skel/tsol`

The `role.link_files` and `role.profile` files in `/etc/skel/tsol` are default startup files that are propagated only to role's home directories.

```
trusted% cd /etc/skel/tsol
trusted% ls
role.link_files role.profile
```

Changing Skeleton Files

The administrator role can do the following:

- Add into the `/etc/skel` or `/etc/skel/tsol` directories other files to be copied
- Modify or rename the default files in `/etc/skel` or `/etc/skel/tsol` pexec(1)

In the Trusted Solaris system, files are automatically copied from the skeleton directory *only* into the SLD at the account's minimum label. Either the user or the user's administrator needs to create either a `.copy_files` or `.link_files` file or both, as described in “Using `.copy_files` and `.link_files`” on page 97, to ensure the desired initialization files are linked or copied into subsequently-created SLDs.

Accessing All Bundled Man Pages

To ensure that the `man(1)` command can find all of the man pages for all the products bundled into the Trusted Solaris product (CDE, X windows, Solaris Management Console), the `MANPATH` environment variable should include all these directories: `/usr/man`, `/usr/openwin/man`, and `/usr/dt/man`.

Users can put the following into their shell initialization files or administrators can put the following into site-wide shell initialization files in `/etc/skel` for all users:

```
setenv MANPATH="/usr/dt/man:/usr/openwin/man:/usr/man:$MANPATH"
```

To find out what your `MANPATH` setting is, enter:

```
$ echo $MANPATH
```

The `MANPATH` should include at least all of the following:

```
/usr/dt/man:/usr/man:/usr/openwin/share/man
```

Using `.copy_files` and `.link_files`

Two new Trusted Solaris copy and link-control files (`.copy_files`, and `.link_files`) can help users or administrators automate the copying or linking of startup files into SLDs created in each account's home directory MLD. These files are created in the first SLD created for the account. The user or the administrator can list in `.copy-files` whatever files should be copied and list in `.link-files` whatever files should be linked from the first SLD into one or more SLDs at other labels. Whether files are copied or linked is at the discretion of whoever is doing the setup.

After any file is copied from the minimum label SLD into a subsequently-created SLD at another label, the file may be edited by the user, so that different versions may appear in different SLDs. Copying would be desirable, for example, if users need to use different mail aliases when they are working at different sensitivity labels. To put a copy of the `.Xdefaults-hostname` in each SLD, you would list `.Xdefaults-hostname` in the `.copy_files` file. See "To Propagate Startup Files to Everyone's Home Directory SLDs" on page 109.

Following the steps in the procedures mentioned above, the administrator role puts into the `/etc/skel` directory generic copies of startup files. The administrator also creates in the skeleton directory a master `.copy_files` containing a list of any startup files to be copied and/or a master `.link_files` containing a list of any

startup files to be linked to all home SLDs. When setting up user accounts, the administrator role specifies the pathname of the modified skeleton directory. All the files from the skeleton directory (including the `.link_files` and/or `.copy_files`) are copied into the user's minimum sensitivity label home SLD, and, based on the `.link_files` and/or `.copy_files`, the specified files are either linked or copied to every subsequently-created SLD.

Whenever a workspace is created at a new label, `dtsession(1)` runs the `updatehome(1M)` command to read the `.copy_files`, and `.link_files` in the account's minimum label SLD and copy or link any listed files into the new workspace.

The `updatehome` command consults the copy and link-control files and performs the actions shown in the following table:

Worksheet for Copy and Link Files

Here are some examples of common files with a worksheet for planning which files to copy or link.

TABLE 3-5 Planning Worksheet for Copying and Linking Startup Files Between SLDs

Common Startup Files	List to be copied (for <code>.copy_files</code>)	List to be copied (for <code>.link_files</code>)
<code>.bugtraqrc</code>		
<code>.cshrc</code>		
<code>.dtprofile</code>		
<code>.login</code>		
<code>.Xdefaults</code>		
<code>.Xdefaults-hostname</code>		
<code>.mailrc</code>		
<code>.newsrc</code>		
<code>.profile</code>		

Administering the Automatic Running of Jobs Using cron, at, and batch

In Trusted Solaris, the SMC Job Scheduler tool, when launched in the Files scope, can be used to manage jobs. This section describes the difference in managing `cron(1M)` and its associated commands in the Trusted Solaris system. See the *System Administration Guide, Volume II* for basic `cron` information. For Trusted Solaris modifications see also the modified man pages for `at(1)`, `atq(1)`, `atrm(1)`, `cron(1M)`, and `crontab(1)`.

In the default `policy.conf(4)` file all users are assigned the Basic Solaris Profile, which assigns the Manage Owned Jobs authorization that is needed in order for any user to manage his or her own `cron` or `at` jobs.

The `crontab` file is generated by a user or role account using the `crontab(1)` command (which, in the Trusted Solaris system, must be in one of the account's execution profiles). The `atjob` file is generated by a user or role account using the `at(1)` or `batch` command (either of which must be in one of the account's execution profiles). In the Trusted Solaris system, the `crontabs` and `atjobs` spool directories are MLDs that hold job files at different sensitivity labels. With MLDs as spool directories, one user can have multiple `crontab` files at different sensitivity labels within the `crontabs` directory, and, similarly, one user can have multiple `atjob` files at different labels within the `atjobs` directory.

Running a Job with a Profile Shell



Caution - If one of the profile shells is specified to execute a job, the security administrator role must ensure that all of the job's commands are also in an execution profile assigned to the invoking user.

`cron_jobs` are executed using one of the profile shells (also called administrator's shells), which are documented on the `pfexec(1)` man page, if either of the following is true:

- The account's login shell is the one of the profile shells or
- The `$SHELL` environment variable is set to

Otherwise, `cron` uses the default Bourne shell, `sh(1)`, for `cron_jobs`.

A user can use `at` with the `-c` (for `csh`), `-k` (for `ksh`), `-s` (for `sh`), option along with the `-P` (for profile shell) option to specify the shell with which the job should be run. Therefore, for `at_jobs` there is a third case in which the profile shell is used. `at_jobs` are executed in the profile shell if either:

- The account's login shell is one of the profile shells or
- The `$SHELL` environment variable is set to `/bin/[p]fsh|[p]ksh|[p]csh` or
- The `at` command is specified with the `-P` option
 - If none of the previously described conditions apply, `at` uses:
- Any shell specified with either the `-c`, `-k`, or `-s` options or
- The default shell, `sh`

Running Privileged Commands in `at` or `cron` Jobs

If a command in an `at` or `cron` job needs to run with privileges, either forced or inheritable privileges may be made available. Allowing a command to run with forced privileges (which apply no matter who executes the command) is not usually consistent with a site's security policy, so the security administrator role usually needs to do the following to make the privileges available by inheritance:

- Specify the command and any privileges it needs in one of the invoking user's profiles using the `Profile Manager` and
- Specify that the job is executed with a profile shell, as described in "Running a Job with a Profile Shell" on page 99

See "Assigning Privileges to Commands and Actions" on page 307 and "To Give Forced Privileges to a Command" on page 329, or "Giving Inheritable Privileges to a Command or Action" on page 309 for more information. See also "To Write a Profile Shell Script that Runs Privileged Commands" on page 333, which uses a `cronjob` in the example.

How the UNIX Domain Socket is Used for Communications

The communication mechanism between `crontab(1)`, `at(1)`, `atrm(1)`, and `cron(1M)` in the Trusted Solaris system is a UNIX domain socket. See the man page for `libt6(3NSL)`.

The `cron(1M)` command is modified to create and bind the UNIX domain socket to `/etc/cron.d/CRON`. The `/etc/cron.d/CRON` file is also used as a lock file to prevent more than one execution of the clock daemon.

An ancillary file is created in the `crontabs` MLD for each `crontab` file and in the `atjobs` MLD for each `atjob` file. Modification of `crontab` or `atjob` file also changes the ancillary file data. The ancillary file is named `username.ad` for a `crontab` file, and `jobname.ad` for an `atjob` file. The ancillary file contains information used by `cron` to set up a job.

Allowing Access to Jobs Owned by Others

The default Trusted Solaris security policy does not allow users to access jobs owned by other users. To allow certain users to access jobs belonging to other users, the security administrator role uses both of the following:

- The `at.admin` and `cron.admin` files in `/etc/cron.d`
- The Manage All Jobs authorization

The Manage All Jobs authorization allows the account to add, modify or delete any user's job and to modify cron policies in the Job Scheduler tool of the SMC.

Conditions for Access to Other's Jobs

An account invoking `at`, `atq`, `atrm`, or `crontab` can look at, edit, or remove jobs belonging to another user only if the following conditions are met.

Conditions for at-related Commands

When using `at`, `atq`, or `atrm`, for an account to create or access an `at_job` owned by another user:

1. The specified `username` or the `username` of the specified `at_job`'s owner is one of the special system account names listed in the `at.admin` file and `3` must be true, or
2. The `username` of the specified `at_job`'s owner is the name of a role account and `3` must be true.
3. The account has the *Manage Owned Jobs* authorization in an execution profile.
4. If neither of 1 or 2 is true, the invoking account must have the *Manage Owned Jobs* authorization in an execution profile

Conditions for the crontab Command

When using `crontab`, for an account to create or access a `crontabs` file owned by another user:

1. The specified `username` is one of the special system account names listed in the `cron.admin` file and `3` must be true, or
2. The specified `username` is one of the role account names and `3` must be true.
3. The invoking account has the *Manage All Jobs* authorization.
4. If neither of 1 or 2 is true, the invoking account must have the *Manage Owned Jobs* authorization in an execution profile.

Miscellaneous

`cron(1M)` is started at `ADMIN_LOW` by the boot profile, and then it is changed to run at `ADMIN_HIGH` after it creates the UNIX domain socket at `ADMIN_LOW`.

Trusted Solaris is delivered with the following `crontab` files:

- At the `ADMIN_HIGH` sensitivity label, pairs of `crontab` and ancillary files for `root`, `uucp`, `adm`, and `sys`.
- At the `ADMIN_HIGH` sensitivity label, pairs of `crontab` and ancillary files for `root`, and `lp`.

The `/var/cron/log` file is created by the clock daemon at `ADMIN_HIGH`. The clock daemon logs its internal messages in this log file .

Specifying an Authorization for a User or Role

To indirectly assign one or more authorizations to an account, the security administrator can use the SMC User Manager to assign an existing rights profile that contains any desired authorization. The security administrator role can alternately follow the procedure “To Directly Specify an Authorization for a User” on page 110, which tells how to use the Admin Editor action to directly assign one or more authorizations. The direct assignment is made by modifying the account’s entry in the `/etc/user_attr` file. As described on the `user_attr(4)` man page, each entry has the form:

```
account_name:qualifier:res1:res2:attr
```

The `auths=` keyword can be entered in the `attr` field followed by one or more comma-separated authorizations; separated from any preceding or following key-value pairs by a semi-colon (;). The following example gives the user `jedgar` the authorizations to allocate devices, to enable logins, to print postscript files, to log in remotely, and to shut down the system.

```
jedgar::::lock=lopen;profiles=All;idletime=5;idlecmd=lock;labelview=external,show  
s1;min_label=0x00000000000000000000000000000000000000000000000000000000000000  
00;clearance=0x7fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff  
ff;min_label=0x00040c00000000000000000000000000000000000000000000000000000000  
00;clearance=0x0006cc0000000000000000000000000000000000000000000000000000000  
00;auths=solaris.device.allocate,solaris.login.enable,solaris.print.ps,solaris.lo  
gin.remote,solaris.system.shutdown;type=normal
```

Assigning the SMC to Normal User Accounts

The Basic Solaris User Profile provides the authorization needed for viewing most information in the Solaris Management Console (SMC) tools. The default `policy.conf(4)` file ships with an entry that assigns the Basic Solaris User Profile to all users.

Note - The responsible administrator at each site may remove or replace the Basic Solaris User rights profile in the default `policy.conf(4)` file or assign this rights profile only to selected users. Therefore, not all users may be able to view the same information in the SMC at all sites.

By default, only administrators are assigned the required authorizations to manage the system using the SMC, but it is possible to assign one or more of the required authorizations to a normal user account. A user with the required authorizations is allowed to use the SMC in a normal user workspace without the trusted path check being performed. The required authorizations are shown in the following table:

Manage Users	Add, modify, or delete user accounts and user templates
Change Password	Change passwords for user and role accounts
Set User Audit Info	Specify audit information for user accounts
Set User Label Info	Specify the minimum label, clearance, label view, and label translation options for a user account

User Setup Procedures

▼ To List All the Roles

Note - The SMC server is not started until the `smc(1M)` application is launched the first time, and the `smrole(1M)` command does not work until the SMC server has been started.

Note - The examples include the required double dash (`--`) either at the end of the command line or before the `list` subcommand's options.

1. **To list the roles in a name service domain, use `smrole list` with the `-D` option to specify the `name_service_type/server_name/domain_name`.**

The following screen example lists the roles defined in NIS+ tables for a domain called `diomedeidae.sun.com` when the name of the NIS+ master server is `albatross`:

```
/usr/sadm/bin/smrole list -D nisplus:/albatross/diomedeidae.sun.com --
```

2. **To list the roles defined in the local Files scope on a computer, use `smrole list` followed by the double dash `--`; you must use the `-h hostname` option if you want to list roles from local files on another computer running the SMC.**

The following screen example shows the `smrole` command used to list all roles defined in local files on the current host:

```
/usr/sadm/bin/smrole list --
```

The following screen example shows the `smrole` command used to list all roles defined in local files on the computer named `trusted`:

```
/usr/sadm/bin/smrole list -- -h trusted
```


▼ To List All Rights Profiles

Note - The SMC server is not started until the `smc(1M)` application is launched the first time, and the `smprofile(1M)` command does not work until the SMC server has been started.

Note - The examples include the required double dash (`--`) either at the end of the command line or before the `list` subcommand's options.

1. To list the rights profiles in a name service domain, use `smprofile list` with the `-D` option to specify the `name_service_type:/server_name/domain_name`.

The following screen example lists the profiles defined in NIS+ tables for a domain called `diomedeidae.sun.com` when the name of the NIS+ master server is `albatross`:

```
/usr/sadm/bin/smprofile list -D nisplus:/albatross/diomedeidae.sun.com --
```

The following screen example lists the security attributes of the `All` profile defined in NIS+ tables for a domain called `diomedeidae.sun.com` when the name of the NIS+ master server is `albatross`:

```
/usr/sadm/bin/smprofile list -D nisplus:/albatross/diomedeidae.sun.com -- -l -n All
```

2. To list the profiles in the Files scope, use `smprofile list` followed by the double dash `--`; the `-h hostname` option is needed only if you want to list profiles from local files on another computer running the SMC.

The following screen example shows the `smprofile` command used to list all roles defined in local files on the current host:

```
/usr/sadm/bin/smprofile list --
```

The following screen example shows the `smprofile` command used to list all roles defined in local files on the computer named `trusted`:

```
/usr/sadm/bin/smprofile list -- -h trusted
```

The following screen example lists the All profile attributes defined in local files on a host called trusted:

```
/usr/sadm/bin/smprofile list -- -h trusted -l -n All -p tsol
Profile name: All
Description: Execute all commands and actions
help: RtAll.html
Command: *;*;*;*
  policy: tsol
  type: act
Command: *
  policy: tsol
  type: cmd
```

▼ To Set Up System-Wide User Attributes

See also “Making Decisions About User Accounts” on page 83 and Chapter 5.

1. **Review the system-wide user attribute settings in the `audit_control(4)`, `label_encodings(4)`, and `policy.conf(4)` files.**
2. **Assume the Security Administrator role.**
3. **Make any modifications needed to the files in Step 1 on page 106 to suit your site’s security policy.**

Note - If these changes are made while the system is being configured on the name service master, then the files can be copied to each client computer as it is configured. If any changes are made after system configuration, these files have to be manually distributed to all computers.

See “Default User Security Attributes” on page 85 for tables showing the default settings.

- a. **To modify the `audit_control` file if needed, assume the use the Audit Control action in an ADMIN_LOW workspace.**
- b. **To modify the `label_encodings` file if needed, use the Edit Label Encodings action in an ADMIN_HIGH workspace.**
- c. **To modify the `policy.conf` file if needed, use the Admin Editor action in an ADMIN_LOW workspace.**

4. Use the SMC User Templates tool to create a user template with an appropriate set of default attributes.
 - a. Click the Users tool, then click the User Templates tool.
 - b. Choose Actions->Add Templates and follow the online help instructions.

5. Use the user template when you create a new account.
 - a. Click the Users tool, then click the User Accounts tool.
 - b. Choose either Add User->From Template or Add Multiple Users->From Template from the Action menu and follow the online help instructions.

6. To change attributes for an individual user account, use the SMC User Accounts tool, select the user, choose Properties from the Action menu.

The system administrator role can change most attributes. Only the security administrator role can assign roles or profiles or modify Trusted Solaris attributes (minimum label, clearance, label view, label translation, account availability options), set up individual auditing, or change the mail server.

▼ To Make .login or .profile Looked at During Login

Note - This procedure changes the default for all users on the host where the change is made.

1. Assume the security administrator role and go to an ADMIN_LOW workspace.

See “To Log In and Assume an Administrative Role” on page 32 if needed.
2. Use the file manager or commands in a terminal emulator to copy `sys.dtpofile` from `/usr/dt/config` to `/etc/dt/config`.

Create the destination directory if it does not already exist.

```
$ cd /usr/dt/config
$ cp sys.dtpofile /etc/dt/config
```

3. Use the Admin Editor action to open the `sys.dtpofile` file for editing.
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
4. Remove the pound sign (#) before the `DTSOURCEPROFILE` variable assignment at the end of the file.
After editing, the line should look like this sample screen.

```
DTSOURCEPROFILE=true
```

5. Save and close the file.

```
:wq
```

▼ To Force dtterm to Launch New Shells as Login Shells

Do this procedure once for each home directory SLD at which the account works, or do it once in the home directory SLD at the account’s minimum label, and then list the `.Xdefaults-hostname` in either `.copy_files` or `.link_files`, as described in “Using `.copy_files` and `.link_files`” on page 97. See also `updatehome(1M)`.

1. Go to your home directory.

```
trusted% cd
```

2. Use a text editor to create or modify the `.Xdefaults-hostname` file.
3. Make the following entry.

```
Dtterm*LoginShell: true
```

4. Write and quit the file.

```
:wq
```

▼ To Separate the Shell Initialization Files for Each Shell

- ◆ **In the System Administrator role, follow the steps under “How to Set Up the User Initialization File” in the *User Accounts, Printers and Main Administration* manual for Solaris.**

The procedure tells you how to create three shell-specific skeleton directory names that are entered into the Skeleton path field in the User Manager. The procedure also tells you to copy the `local.login` file to the `skelC` subdirectory, the `local.profile` file to the `skelK` subdirectory and the `local.login` file to the `skelB` subdirectory.

- ◆ **Assume the administrator role, create a `skelP` subdirectory (for a specialized version of the `.profile` to be installed into the home directories of users whose default shell is one of the profile shells).**
- ◆ **Enter the correct `skelX` subdirectory name into the Skeleton path field in the User Manager, based on the user’s default shell.**

▼ To Propagate Startup Files to Everyone’s Home Directory SLDs

Note - Any user can put a `.copy_files` or `.link_files` into his or her home directory MLD at the SLD that corresponds to the minimum sensitivity label or can modify the files in the minimum label SLD if they are already there. This procedure is for the administrator role to automate the setup for multiple users.

1. **Assume the administrator role and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32 if needed.
2. **Go to `/etc/skel`.**

```
$ cd /etc/skel
```

3. **Put generic copies of startup files into the skeleton directory.**
The following example shows added startup files for the mailer and for `dtterm`. See for the entry you need to make in

Managing Roles

This chapter gives the necessary background and describes how to create and modify roles under the following headings:

- “When to Create a New Administrative Role” on page 111
- “Trusted Path Attribute” on page 112
- “Caveats About Changing the Recommended Roles” on page 112
- “Allowing Remote Logins by Administrative Roles” on page 113
- “Creating a New Role” on page 113
- “Customizing Profiles for the Recommended Roles” on page 113
- “Aliasing vi to adminvi” on page 114
- “Assigning trusted_edit as a Role’s Default Editor” on page 115

This chapter contains the following procedures:

- “To Enable Root or a New Role to Administer NIS+” on page 117
- “To Enable Any Role to Login Remotely” on page 118
- “To Configure a New Role” on page 116
- “To Assign the trusted_edit Editor to a Role” on page 118

When to Create a New Administrative Role

Sites may create a new administrative role to allow a number of individuals to perform a defined set of tasks sharing the same home directories and ownership of

files. A site might create a new administrative role, for one example, to combine the responsibilities of the three default administrative roles into one. See “Caveats About Changing the Recommended Roles” on page 112

Trusted Path Attribute

The trusted path attribute is available during boot or when a program is running in an administrative role workspace.

Many administrative commands and actions require the trusted path attribute. For instance, when the Solaris Management Console (SMC) is launched on behalf of an administrative role, the SMC checks for the trusted path attribute.

Caveats About Changing the Recommended Roles

The security administrator can assign to a new role only the capabilities that the security administrator role has to give away. The primary administrator role must be used to configure roles that have capabilities not available to the security administrator role.

A single combined administrative role can be created to do administrative tasks, if the organization’s security policy does not require the separation of roles.

If the security policy allows the combination of the capabilities of the default security administrator and administrator roles, creating a new role is preferable to expanding the capabilities of the `root` role. See “Creating a New Role” on page 113.



Caution - If site security policy allows, `root`’s capabilities can be extended to allow `root` to do NIS+ administration from a NIS+ client, although this is not recommended. To do so, add the name of all NIS+ clients that are to be used by `root` to the NIS+ `admin` group using the `nisgrpadm(1)` command as described in “To Enable Root or a New Role to Administer NIS+” on page 117.

Allowing Remote Logins by Administrative Roles

In both Trusted Solaris and Solaris, any account can remotely log in without an authorization under certain conditions while other types of conditions require the `Remote Login` authorization. (The conditions and types of logins that require the authorization are described in “Managing Remote Logins” on page 90.) The administrative roles by default have the `Remote Login` authorization, but remote logins by administrative roles require an additional step. The security administrator role needs to change a setting in the `/etc/default/login` file on each computer where administrative roles work, to allow remote log ins from that computer. See “To Enable Any Role to Login Remotely” on page 118.

Creating a New Role

The security administrator role can use the SMC Administrative Roles tool to create a new role. If the new role needs capabilities that the security administrator role does not have to give away, the primary administrator role needs to create the new role. The procedure is described in .

The responsible role may first need to create a new profile for the new role before configuring the new role’s account. Again, if the profile needs capabilities that the security administrator role does not have, the primary administrator role can create the profile.

If a new profile needs new commands or actions, the security administrator role should analyze whether any of the commands or actions need privileges in order to do the tasks the role is assigned to do. See Chapter 13. See the man pages for individual commands for the *required* privileges and *override* privileges a command might need.

Customizing Profiles for the Recommended Roles

Custom role profiles are used when customizing default profiles that are assigned to the recommended roles. Making all changes for each role in its custom profile makes

it much easier to debug problems or characterize the system if you ever need to call for service.

- The root role has the Custom root Role profile assigned by default.
- The Custom secadmin Role profile is nested in the Rights Security Profile, which is assigned to the Security Administrator during configuration, as described in “How to Create Administrative Roles” in *Trusted Solaris Installation and Configuration*.
- Before creating a new role or modifying the Security Administrator, Primary Administrator, or Operator roles, you need to create a Custom *rolename* Profile and then assign it to the role.

See also “To Configure a New Role” on page 116.

- Before modifying the profiles for any role, make sure the appropriate Custom *rolename* Profile is assigned, and make the changes in the custom profile.

Aliasing vi to adminvi

By default, all roles have the restricted editor `adminvi(1M)` assigned instead of the `vi(1)` editor. The `dtterm(1)` terminal is assigned to the roles by default. If a role accidentally types `vi` instead of `adminvi`, the following error displays:

```
vi: command not in profile
```

To allow roles to avoid this error if they type `vi`, the default `profile(4)` file in the home directories for all roles has the following function to alias `vi` to `adminvi`:

```
vi() { adminvi $1 ; }
```

However, the alias does not work unless an entry like the following is also made for each computer the role uses in the `$HOME/.Xdefaults-hostname` file. A copy of the file must be in every SLD at which the role works:

```
Dtterm*LoginShell: true
```

For more information about which startup files are read, see the discussion in Chapter 3 under “Managing Initialization Files” on page 91. For the procedure to make the supporting entry in `.Xdefaults-hostname`, see “To Force dtterm to Launch New Shells as Login Shells ” on page 108.

Assigning trusted_edit as a Role's Default Editor

The `/usr/dt/bin/trusted_edit` script is a wrapper that launches an editing window using the `$EDITOR` environment variable and that audits all changes. To make `trusted_edit` available as an editor for a role, the security administrator can add the `trusted_edit` script to one of the account's effective profiles and assign the `proc_audit_tcb` privilege to the script.

If desired, the security administrator or the affected role can also alias `vi` to `trusted_edit`. See “To Assign the `trusted_edit` Editor to a Role” on page 118 for the procedure.

Procedures for Administering Roles

▼ To Make Changes to a Role

- 1. Decide whether the modified role needs new commands, actions, or authorizations to do its work.**
- 2. Decide whether any of the commands or actions need privileges or other security attributes, and decide whether the role and the command or action can use these security attributes in a trustworthy manner.**
- 3. Decide whether the role needs a new or a modified rights profile.**
- 4. If the role needs a new rights profile, create it.**

Follow the online help for how to use the `Users Rights` tool to create the new rights profile.

See “Adding or Modifying a Rights Profile” on page 144, if needed.
- 5. If a custom role profile does not exist for the role, create it.**

Use the `Rights` tool to create an empty profile and assign it the appropriate name. For example, for a role called `Audit Administrator` with an account name of `auditadmin`, you would create a `Custom Auditadmin Role` profile.

See “Adding or Modifying a Rights Profile” on page 144, if needed.
- 6. Make sure the role has the `Custom rolename Role` profile assigned.**

For example, if you are changing the System Administrator role and the Custom Admin Role profile is not assigned to the role, you would use the Administrative Roles tool to add the Custom Admin Profile to the list of Rights for the role.

7. Make any needed modifications to the Custom *rolename* Role profile.

Use the Rights tool to add any of the following to the Custom *rolename* Role profile

- New commands or actions with security attributes
- New authorizations
- Any new profile created in Step 4 on page 116

8. Use the SMC Administrative Roles tool to modify the properties for the role and add the profile(s) from Step 7 on page 116 and Step 4 on page 116.

See “Adding or Modifying a Role Account” on page 136 and “To Add or Modify a Role” on page 151, if needed.

9. If you are running the NIS+ naming service, make an entry for the new role in the NIS+ admin group.

See “To Enable Root or a New Role to Administer NIS+” on page 117, if needed.

▼ To Configure a New Role

1. Define what the role’s responsibilities are to be, and decide and what commands, actions, security attributes, and authorizations the role needs to do its work.

2. Decide whether any of the commands or actions need privileges or other security attributes to do their work, and decide whether the role and the command or action can use these security attributes in a trustworthy manner.

3. Decide if the role needs to have a new or modified rights profile, and if so, use the SMC Users Rights tool to create or modify the rights profile.

Follow the online help for how to use the Rights tool to create a new rights profile for the role.

4. Create a custom role profile for the role.

For example, for a role whose username is auditadmin, you would create an empty Custom Auditadmin Role profile. In the profile’s description you would enter: “Modify this rights profile to customize the Audit Administrator role.”

5. Use the SMC Administrative Roles tool to create an account for the role and add the profiles from Step 7 on page 116 and Step 4 on page 116.
See “Adding or Modifying a Role Account” on page 136 for more information about specifying role attributes in the SMC Users Administrative Roles tool.
6. If you are running the NIS+ naming service, make an entry for the new role in the NIS+ `admin` group.
See “To Enable Root or a New Role to Administer NIS+” on page 117, if needed.

▼ To Enable Root or a New Role to Administer NIS+

1. Log into the NIS+ master and assume the security administrator role.
2. Log into the NIS+ master and use the `nisgrpadm(1)` command to add an entry in the `admin` NIS+ group.
 - a. To enable root to administer NIS+ from a NIS+ client, enter `nisgrpadmin -a` followed by the name of the `admin` group followed by the fully qualified name(s) of all NIS+ clients from which the root role wishes to remotely administer NIS+.

```
$ nisgrpadm -a admin fully_qualified_hostname . . .
```

For example, the following line adds two computers, `trusted` and `trustworthy` in the `security` domain at `sun.com`.

```
$ nisgrpadm -a admin trusted.security.sun.com. trustworthy.security.sun.com.
```

- b. To enable a new role to administer NIS+ from a NIS+ client, enter `nisgrpadmin -a` followed by the name of the `admin` group followed by the fully qualified name(s) of the new role.

```
$ nisgrpadm -a admin role's_principal_name . . .
```

The example shows the new role in the in the `security` domain at `sun.com`.

```
$ nisgrpadm -a admin newrole.security.sun.com.
```

▼ To Enable Any Role to Login Remotely

Note - See “Managing Remote Logins” on page 90 for a description of other remote login prerequisites.

Do the following on every computer where the root role will work, to enable remote logins from that computer.

1. **Log in, assume the security administrator role, and go to a workspace at the ADMIN_LOW label.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. **Use the Admin Editor action to open the /etc/default/login file for editing.**

3. **Find the line:** `CONSOLE=/dev/console`.

4. **Insert a pound sign (#) to comment out the line.**

```
#CONSOLE=/dev/console
```

▼ To Assign the trusted_edit Editor to a Role

1. **Login and assume the security administrator role.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. **If the trusted_edit command is not in one of the role’s profiles, use the SMC Users Rights tool to add the trusted_edit command to one of the account’s profiles.**

Follow the online help for how to use the Rights tool to modify the profile.

- a. **Add the /usr/dt/bin/trusted_edit script to the desired profile.**

- b. **Give the script the proc_audit_tcb privilege.**

3. **Alias the vi command to trusted_edit by doing the following:**

- a. **Search for the vi function in the .profile file in the role’s home directory:**

```
vi() {adminvi $1;}
```

b. **Replace** `adminvi` **with** `trusted_edit`:

```
vi() {trusted_edit $1;}
```

c. **Make sure the following entry is also made in the** `$HOME/.Xdefaults-hostname` **file in the SLD at each label at which the role works:**

```
Dtterm*LoginShell: true
```

Note - Make a file for each hostname the role uses. For example, when the role works on computers named `trusted` and `trustworthy`, create a `$HOME/.Xdefaults-trusted` and a `$HOME/.Xdefaults-trustworthy` in each SLD.



Caution - Because `trusted_edit` launches a window, it cannot be used for command line editing. Command line editing may be the only option available in a remote login session, so for this reason, do not assign `trusted_edit` as the only editor for a role, unless the role never needs to do remote editing on the command line.

Using the SMC User Manager to Manage User and Role Accounts and Profiles

This chapter details the decisions you need to make before specifying accounts for users and roles:

- “Adding or Modifying a User Account”
- “Adding or Modifying a Role Account” on page 136
- “Adding or Modifying a Rights Profile” on page 144

Adding or Modifying a User Account

This section describes the decisions to make for each user account and where to specify each type of user attribute in the SMC Users tools. The information is organized according to the tabs on the User Accounts Properties and User Templates dialogs shown in the following table. The description for each tab type includes decisions to make before creating an account. The names of the fields are shown with initial capitals, for example: `User Name`.

TABLE 5-1 Users Attributes

User Accounts: Properties Tabs	User Templates Tabs	Section
General	General	“Assigning General Attributes to Users” on page 123
Groups	Groups	“Assigning Groups to Users” on page 125
Home Directory	Home Directory	“Configuring Users’ Home Directories” on page 126
<i>tab above includes sharing options</i>	Home Directory Sharing	“Configuring Users’ Home Directories” on page 126
Passwords	<i>not available</i>	“Assigning Passwords to Users” on page 127
Password Options	<i>not available</i>	“Configuring Password Options for Users” on page 128
Mail	Mail	“Configuring Mail Options for Users” on page 129

TABLE 5-1 Users Attributes *(continued)*

User Accounts: Properties Tabs	User Templates Tabs	Section
Rights	<i>not available</i>	“Assigning Rights to Users” on page 130
Roles	<i>not available</i>	“Assigning Roles to Users” on page 131
Trusted Solaris Attributes	<i>not available</i>	“Assigning Trusted Solaris Attributes to Users” on page 132
Audit	<i>not available</i>	“Assigning Audit Classes to Users” on page 134

Assigning General Attributes to Users

For each user account you plan to create, decide on the following:

- Decide on a unique `User Name` (login name) and associated `UID` (User Id).

User names and UIDs must be unique to ensure traceability of activities back to a single identified user, so each user name and UID:

- Should not be duplicated anywhere on the network
 - Should not be reused during the life of the system
- Enter the user’s full name in the `Full Name` field.
- Decide on a `Description`.

The `Description` is stored in the `GCOS` field of the `passwd(1)` database. It usually contains the first and last name of the user and perhaps the job title and work phone number. The text in the `Description` field appears in the `From:` line when the users sends email, and is part of the information sent about the user if someone enters `finger(1)`, among other uses. Here is an example description:

```
From: Roseanne Sullivan -- Manual Laborer
```

- **Decide on a Login Shell.**

The choices in the shell menu for user accounts are Bourne Shell, C Shell, Korn Shell, BASH, T Shell, Z Shell or other. You can assign a profile shell to users by choosing other and then typing in either /bin/pfsh, /bin/pfcsh, or /bin/pfksh. (See the pfexec(1) man page for descriptions of the profile shells, which are also called Administrator's shells.) The Bourne, Korn, and C shells allow the account to execute all available commands that do not need to inherit privilege. In contrast, while working in a profile shell, an account can execute only those commands that are in the account's set of profiles. See the following man pages for more information about the listed shells: csh(1), ksh(1), bash(1), tsh(1), zsh(1).

- **Decide on Account Availability.**

Decide whether to specify that the account is to expire on a certain date. If the account is to expire, decide on the date.

The following table shows the general user attributes and where they are entered.

TABLE 5-2 General User Attributes and Where They are Entered

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, General Tab	Click User Templates Tool Choose Action->Add Template
Wizard Step 1. Enter a user name. User Name	User Information User Name <i>(not modifiable; set at account creation)</i>	User Template Name: <i>(User Name field not available; set at account creation)</i>
Full Name	Full Name <i>(not modifiable; set at account creation using either the Add User or Add Multiple Users options from the Actions menu)</i>	<i>(Full Name field not available; set at account creation)</i>
<i>(User Type field not available; automatically set)</i>	User Type (automatically set)	<i>(User Type field not available; automatically set)</i>
Description:	Description:	<i>(Description Field not available; set at account creation)</i>

TABLE 5-2 General User Attributes and Where They are Entered *(continued)*

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, General Tab	Click User Templates Tool Choose Action->Add Template
Step 2. Enter a user identification number. User Id Number	<i>(User Id shown but not modifiable; set at account creation)</i>	<i>(User Id field not available; set at account creation)</i>
<i>(Login Shell field not available; Bourne shell assigned automatically)</i>	Login Shell:	Login Shell:
<i>(Account Availability fields not available; by default, account is always available)</i>	Account Availability Account is Always Available Account is Available Until mm/dd/yyyy Account is Locked <i>(Account Availability fields not available for System Administrator role. Available for Security Administrator role.)</i>	Account Availability Account is Always Available Account is Available Until mm/dd/yyyy Account is Locked

Assigning Groups to Users

- Decide the Primary Group for the user and if the user should belong to any Additional Group(s).

In order to specify additional groups, either create a template with additional groups specified and use that template when creating a new account, or modify the user's properties after creating the account.

The default groups that can be assigned as either primary or additional (also called secondary) groups are adm, bin, daemon, lp, mail, noaccess, nuucp, other, root, staff, sys, sysadmin, tty, and uucp.

The following table shows where groups are specified for a user account.

TABLE 5-3 Where Groups are Assigned to User Accounts

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, Groups Tab	Click User Templates Tool Choose Action->Add Template
Wizard Step 4. Select the user's primary group. Primary Group:	Primary Group Primary Group:	Primary Group
<i>(Additional Groups not available)</i>	Additional Groups Available Groups: Member Of:	Additional Groups Available Groups: Member Of:

Configuring Users' Home Directories

- Decide on the account's home directory path.
- Decide on the account's home directory server.
- Decide whether to use `/etc/skel` or another skeleton directory and what to put in it.

Creating a user template is the only way you can specify an alternate skeleton directory for initial files. You enter the pathname to the alternate skeleton directory in the Copy Initial Files From: field.

- Decide whether to let the users do their own modifications to shell initialization files or whether to provide your own administratively-controlled versions.

If the latter, you must also do the set up described in "To Separate the Shell Initialization Files for Each Shell" on page 109 in Chapter 3. If a shell-specific subdirectory has been created for each of the shells, you need to enter the correct `skelX` subdirectory name into the Skeleton path field in the Copy Initial File From: field in a template. For more details about managing initialization files, see "To Make .login or .profile Looked at During Login" on page 107 and see "Managing Initialization Files" on page 91 in Chapter 3.

- Decide how the home directory is to be shared. The values supplied in the User Properties or Template sets the default permission bits for files.
- Decide whether to have the home directory automounted.

If you use the wizard, an entry is automatically created in the auto_home map and other setup is done to enable the automounting of the user's home directory.

Note - The server for the account's home directory must be configured before you create the account.

The following table shows where home directories are specified for a user account.

TABLE 5-4 Home Directory Properties and Where They are Specified for Users

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, Home Directory and Home Directory Sharing Tab	Click User Templates Tool Choose Action->Add Template
Wizard Step 5. Create the user's home directory. Home Directory Server: Home Directory Path: <i>(Home directory is automounted by default)</i>	Home Directory Information Home Directory Path: Home Directory Server: Automatically Mount Home Directory	Home Directory Information Home Directory Path: Home Directory Server: Automatically Mount Home Directory
<i>(Neither: Home Directory Sharing nor Copy Initial Files From: are available)</i>	Home Directory Sharing Shared With: Group Members Read-only Access Full Access All Users Read-only Access Full Access	Copy Initial Files From:

Assigning Passwords to Users

- Decide whether the user can set the password at first or next login or must use a password you create.

The security administrator role usually creates a password for a new account. The security administrator role gives the initial password to the new user to be used at first login. The security administrator role can use the Properties option to create a

new password for the user, or can force the user to choose a new password at next login.

Note - Neither users nor roles use either the `passwd(1)`, the `yppasswd(1)`, or the `nispasswd(1)` commands to change passwords. Both users and roles use the TP menu Change Password option to change their own passwords.

- If you plan to create the password for the user, decide between typing a password of your choice or choosing from an automatically-generated list (Type In or Choose From List).
- Decide whether the account can pick its own password (Type In) when changing its password or if it must choose one from an automatically-generated list (Choose from List).

The following table shows where passwords are entered for users.

TABLE 5-5 Password Properties and Where They are Specified

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, Password Tab	Click User Templates Tool Choose Action->Add Template
Wizard Step 3. Enter the user's password. User set password at first login Set Password By: Type In Choose From List (Update Password By field is not available.)	<i>(available to secadmin; not available to admin)</i> User Password: Set At Next Login Set Password By: Type in Choose from list Update Password By: Type in Choose from list	<i>(Password fields are not available)</i>

Configuring Password Options for Users

The password aging options limit how long any intruder who is able to guess or steal a password could potentially access the system. Establishing a minimum length of time to elapse before change also prevents a user with a new password from reverting immediately to the old password.

Note - The passwords for users allowed to assume roles should not be subject to any password aging restraints.

- Decide if a certain number of days must elapse before the user can change the password
- Decide if the user must change the password after a certain number of days elapses
- Decide how many days before the password expires to send a warning
- Decide if the password expires if not used, and how many days should elapse before the expiration

The following table shows where passwords are entered for users.

TABLE 5-6 Password Options and Where They are Specified

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, Password Options Tab	Click User Templates Tool Choose Action->Add Template
<i>(Password Options in Days not available)</i>	<i>(available to secadmin; not available to admin)</i> Password Options in Days User Must Keep For: Before Change, Alert User: User Must Change By:	<i>(Password Options in Days not available)</i>
Expires If Not Used By:		

Configuring Mail Options for Users

- Decide which computer is to be the mail server for the account.

The following table shows the mail options for users and where they are specified.

TABLE 5-7 Mail Options and Where They are Specified

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, Mail Tab	Click User Templates Tool Choose Action->Add Template
Wizard Step 6. Specify the Mail Server. Mail Server: MailBox: <i>(field is read-only)</i>	Mail Information Mail Server: MailBox: <i>(field is read-only)</i>	Mail Information Mail Server: MailBox: <i>(field is read-only)</i>

Assigning Rights to Users

- The Available Rights are displayed in a scrolling list in the Rights dialog box, along with brief descriptions.

You can assign no rights profiles, one rights profile, or multiple rights profiles to a single user if it is consistent with your site's security policy.

- Decide on which rights profile to assign to the account.

Note - Do not assign any role profiles to a normal user account. Doing so would introduce some measure of risk and a good deal of confusion. To begin with, role profiles you assigned to the user account would be in effect for the user when the user has not assumed a role, which in most cases should not be allowed to happen. What's more, some things that seem like they should work would not: many of the administrative role's commands and applications do not work outside of the administrative role workspace because they require the trusted path attribute.

- Decide the order in which profiles should be listed

The order of profiles is important because when the account invokes a command or action, the profile mechanism uses the command or action the first time its name is found in any of the profiles in the account's profile set—with whatever attributes have been defined for the command or action in the profile where it is found.

Here is a way you can use the sorting order of profiles to your advantage. If you want a command to run with different privileges from those defined for it in an existing profile, create a new profile with the desired privilege assignments for the command and insert that new profile so that the profile mechanism finds the new one first.

The following table shows where rights are assigned to users.

TABLE 5-8 Rights Profiles and Where They are Specified for Users

<p>Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu</p>	<p>Click User Accounts Tool Select User, Choose Action-->Properties, Rights Tab</p>	<p>Click User Templates Tool Choose Action->Add Template</p>
<p><i>(not available)</i></p>	<p>Available Rights Granted Rights <i>(Rights tab available to secadmin; not available to admin)</i></p>	<p><i>(not available)</i></p>

Assigning Roles to Users

The Available Roles display in a scrolling list in the Roles dialog box, along with brief descriptions.

You can assign no roles, one role, or multiple roles to a single user if doing so is consistent with your site's security policy.

- Decide which roles to assign to the account.

Note - In the Administrative Roles tool, the Users tab allows the security administrator to assign roles to users.

The following table shows where roles are assigned to users.

TABLE 5-9 Roles and Where They are Assigned to User Accounts

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, Roles Tab	Click User Templates Tool Choose Action->Add Template
<i>(not available)</i>	<i>(available to secadmin; not available to admin)</i> Available Roles Assigned Roles	<i>(not available)</i>

Assigning Trusted Solaris Attributes to Users

- Decide on a clearance for the account that dominates the minimum clearance set in the `label_encodings` file and that also dominates the minimum label defined in the Label field.

The clearance defines top of the range of labels at which the account can work. Administrative role accounts have a clearance of `ADMIN_HIGH`.

- Decide the minimum label at which the account is permitted to work.

The account's Label: defines the lowest label at which the account can work.

Note - Both the account's clearance and minimum label must be dominated by the highest label and must dominate the minimum clearance that are defined in the user ACCREDITATION RANGE section in the `label_encodings(4)` file. See *Trusted Solaris Label Administration* for more about the label encodings.

- Decide whether the account is allowed to view administrative labels or should be shown alternate labels within the user accreditation range instead.

At some sites the names of administrative labels are considered to be classified information. When each user or role's account is being set up, the security administrator role chooses one of the following for the account:

- Internal

The `INTERNAL` view allows the account to see the names of the administrative labels.

- External

If the label view for an account is set to `EXTERNAL`, the user will see the minimum valid label in the User Accreditation Range instead of the `ADMIN_LOW` label and see the maximum valid label in the User Accreditation Range instead of the `ADMIN_HIGH` label.

Note - It is important to realize that the binary label always remains the same when the `EXTERNAL` view is set. The only difference is that the label is given an alternative name when displayed to hide its real name.

- `System Default`

If the `System Default` option is selected for an account, whatever value is specified in the `label_encodings(4)` file for the `DEFAULT LABEL VIEW` key word applies to the account.

- Decide whether the user can view labels at all.
- Decide what action, if any, should be taken if the user leaves the workstation idle for a specifiable amount of time, whether the screen should be locked or the user's session should be terminated and the user logged out.
- Decide on the maximum length of time that the workstation can potentially sit idle before the specified idle action will be taken.

Choose from the `Idle Time` menu either 1, 2, 3, 4, 5, 10, 15, 30, 60, or 120 minutes or `Forever`. The `Forever` menu option allows the workstation to be idle forever without the idle action occurring.

Note - The `Idle Time` sets the maximum amount of time before the screen lock or logout. Any user can use the `CDE Style Manager Screen` tool to change the amount of time before the start of the screen lock by moving the `Start Lock` slider up to the limit set in this tab. If a number between 1 to 120 is selected, the user cannot turn screen locking off.

The following table shows how Trusted Solaris attributes are specified for users.

TABLE 5-10 Trusted Solaris Attributes and Where They are Specified for Users

<p>Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu</p>	<p>Click User Accounts Tool Select User, Choose Action-->Properties, Trusted Solaris Attributes Tab</p>	<p>Click User Templates Tool Choose Action->Add Template</p>
<p><i>(Trusted Solaris Attributes not available)</i></p>	<p>Labels Minimum Label Clearance View: External Internal Label: Show Hide <i>(available to secadmin; not available to admin)</i></p>	<p><i>(Trusted Solaris Attributes not available)</i></p>
<p><i>(Account Usage not available)</i></p>	<p>Account Usage Lock account after maximum failed logins: No Yes Idle Time: 1, 2, 3, 4, 5, 10, 15, 30, 120 minutes Forever Idle Action: Lock Screen Logout <i>(Trusted Solaris Attributes not available to secadmin.)</i></p>	<p><i>(Account Usage not available)</i></p>

Assigning Audit Classes to Users

To assign audit classes to a user, the security administrator role needs to modify the user's properties after the user account is created.

When a user logs in, any system-wide audit flags from the `audit_control` file are combined with any audit classes assigned to the account in the `audit_user` file to form an audit preselection mask:

`(flags in audit_control + user's always-audit flags) - user's never-audit flags`

See the `audit_control(4)` and `audit_user(4)` man pages for more about how flags that are assigned to individual user or role accounts combine with any flags in the `audit_control(4)` file to specify which actions are audited. Also see *Trusted Solaris Audit Administration* for more about auditing.

The available audit classes are displayed in a scrolling list. The classes are sorted into three categories that determine the circumstances in which they are audited. A class can be audited for the following three types of outcomes:

- Both Success and Failure
- Success only
- Failure only

The following buttons are used to move audit classes into the Included List under three different categories:

- Always Audit
- Never Audit
- Force Except

`Force Except` ensures that the specified class is never audited for the user, and is used to override previously-specified flags.

The following table shows how audit flags are assigned to users.

TABLE 5-11 User Audit Options and Where They are Specified for Users

Click User Accounts Tool Choose Add User -->With Wizard from the Action Menu	Click User Accounts Tool Select User, Choose Action-->Properties, Audit Tab	Click User Templates Tool Choose Action->Add Template
<i>(Audit options not available)</i>	User Audit Classes: Move Classes from the categories Both Success and Failure, Success Only, Failure only from the Excluded list to the Included list using the Add to Always Add to Never Force Except buttons. Excluded Included <i>(Audit tab available to secadmin; not available to admin)</i>	<i>(Audit options not available)</i>

Adding or Modifying a Role Account

This section describes the decisions to make for each role account and where to specify each type of role attribute in the SMC Administrative Role tool. The information is organized as shown in the following table, according to the tabs on the Administrative Roles Properties dialog.

TABLE 5-12 Roles Attributes

Roles Properties Tabs	Section
General	“Assigning General Attributes to Roles” on page 136
Password	“Assigning Passwords to Roles” on page 138
Users	“Assigning Users to Roles” on page 139
Group	“Assigning Groups to Roles” on page 139
Home Directory	“Configuring Roles’ Home Directories” on page 140
Rights	“Assigning Rights to Roles” on page 141
Trusted Solaris Attributes	“Assigning Trusted Solaris Attributes to Roles” on page 141
Audit	“Assigning Audit Classes to Roles” on page 143

Assigning General Attributes to Roles

For each role account you plan to create, decide on the following:

- Decide on a unique Role Name (login name) and associated Role Id.
Role names and role IDs must be unique to ensure traceability of activities back to a single identified user, so each role name and ID:
 - Should not be duplicated anywhere on the network
 - Should not be reused during the life of the system
- Enter the role's full name in the Full Name field.
- Decide on a Description.
The Description is stored in the GCOS field of the passwd(1) database.
- Decide on a Role Shell.
The choices in the shell menu for role accounts are Administrator's Bourne Shell, Administrator's C Shell, or Administrator's Korn Shell. (See the pexec(1) man page for descriptions of the Administrator's shells, which are also called profile shells.) While working in a profile shell, an account can execute only those commands that are in the account's set of profiles.

The following table shows the general role attributes and where they are entered.

TABLE 5-13 General Role Properties and Where They are Entered

Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu	Click Administrative Accounts Tool Select Role, Choose Action-->Properties, General Tab
Wizard Step 1. Enter a Role Name.	
Role Name:	Role Name:
Full Name:	Full Role Name:
Role Id Number:	Role Id: <i>(not modifiable; set at role account creation)</i>
Description:	Description:
Role Shell:	Role Shell
Create a role mailing list.	<i>(not available)</i>

Assigning Passwords to Roles

- Decide whether the role account can set the password at the next login or must use a password you create.

The security administrator role usually creates a password for a new role. The security administrator role gives the initial password to the new role to be used when the role is next assumed. The security administrator role can use the Properties option to create a new password for the role, or can force the role to choose a new password at next login.

Note - Neither users nor roles use either the `passwd(1)`, the `yppasswd(1)`, or the `nispasswd(1)` commands to change passwords. Both users and roles use the TP menu Change Password option to change their own passwords.

- If you plan to create the password for the role, decide between typing a password of your choice or choosing from an automatically-generated list (Type In or Choose From List).
- Decide whether the account can pick its own password (Type In) when changing its password or if it must choose one from an automatically-generated list (Choose from List).

The default is manual (Type In).

The following table shows where passwords are entered for roles.

TABLE 5-14 Password Properties and Where They are Specified for Roles

Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu	Click Administrative Accounts Tool Select Role, Choose Action-->Properties, Password Tab
Wizard Step 2. Enter a role password. Set Password By: Type in Choose from list (Update Password By option not available.)	Role Password: Set Password By: Type in Choose from list Update Password By: Type in Choose from list

Assigning Users to Roles

- Decide which users can assume the role.

The following table shows where users are assigned to roles.

TABLE 5-15 Where Users are Assigned to Roles

Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu	Click Administrative Accounts Tool Select Role, Choose Action-->Properties, Users Tab
Wizard Step 5. Assign users to this role. Add Delete	Users Assigned to This Role: Add Delete

Assigning Groups to Roles

Every role is assigned the sysadmin group 14 as its primary group by default when the role is created. The default groups that can be assigned as additional (also called secondary) groups are adm, bin, daemon, lp, mail, noaccess, nuucp, nobody, other, root, staff, sys, tty, and uucp.

- Decide if the role should belong to any Additional Group(s).

To specify additional groups, you modify the role's properties after creating the role account.

The following table shows where groups are specified for a role account.

TABLE 5-16 Where Groups are Assigned to Roles Accounts

Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu	Click Administrative Accounts Tool Select Role, Choose Action-->Properties, Groups Tab
<i>(Groups not available; sysadmin (GID 14) assigned as primary group by default)</i>	<i>(Primary Group cannot be modified: sysadmin (GID 14) assigned by default)</i>
	Available Groups: Member Of:

Configuring Roles' Home Directories

- Decide the role account's home directory path.
- Decide the role account's home directory server.
- Decide how the home directory is to be shared. The values supplied in the Properties are used to set the default permission bits for files.
- Decide whether to have the home directory automounted.

When you add a new role, an entry is automatically created in the auto_home map and other setup is done to enable the automounting of the role's home directory.

Note - The server for the role account's home directory must be configured before you create the account.

The following table shows where home directories are specified for a role account.

TABLE 5-17 Home Directory Properties and Where They are Specified

Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu	Click Administrative Accounts Tool Select Role, Choose Action-->Properties, Roles Tab
Wizard Step 4. Select a home directory. Home Directory Server: Home Directory Path: (Home directory is automounted by default)	Home Directory Server: (not available, set at account creation) Home Directory Path: Automatically Mount Home Directory
(Home Directory sharing options not available)	Home Directory Shared With: Group Members Read-only Access Full Access All Users Read-only Access Full Access

Assigning Rights to Roles

Except for the root role, which is shipped with a set of rights profile already assigned, each of the recommended roles has a predefined rights profile. Creating the roles by assigning the appropriate rights profiles is described in the “How to Create Administrative Roles” in *Trusted Solaris Installation and Configuration* manual.

- Decide on which rights profile to assign to the role account.
- Decide the order in which profiles should be listed
- The order of profiles is important because when the account invokes a command or action, the profile mechanism uses the command or action the first time its name is found in any of the profiles in the account’s profile set—with whatever attributes have been defined for the command or action in the profile where it is found.

Here is a way you can use the sorting order of profiles to your advantage. If you want a command to run with different privileges from those defined for it in an existing profile, create a new profile with the desired privilege assignments for the command and insert that new profile so that the profile mechanism finds the new one first.

The following table shows where rights profiles are specified for a role account.

TABLE 5-18 Rights and Where They are Specified for Role Accounts

Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu	Click Administrative Accounts Tool Select Role, Choose Action-->Properties, Rights Tab
Step 3. Select role rights. Available Rights Granted Rights	Available Rights Granted Rights

Assigning Trusted Solaris Attributes to Roles

- Decide a clearance for the account that dominates the minimum clearance set in the `label_encodings` file and that dominates the minimum label defined in the Label field.

The clearance defines top of the range of labels at which the account can work. Administrative role accounts have a clearance of `ADMIN_HIGH`.

- Decide the minimum label at which the account is permitted to work.

The account’s Label: defines the lowest label at which the account can work.

Note - Both the account's clearance and minimum label must be dominated by the highest label and must dominate the minimum clearance that are defined in the user ACCREDITATION RANGE section in the `label_encodings(4)` file. See *Trusted Solaris Label Administration* for more about the label encodings.

- Decide whether the account is allowed to view administrative labels or should be shown alternate labels within the user accreditation range instead.

At some sites the names of administrative labels are considered to be classified information. When each user or role's account is being set up, the security administrator role chooses one of the following for the account:

- Internal

The `INTERNAL` view allows the account to see the names of the administrative labels.

- External

If the label view for an account is set to `EXTERNAL`, the user will see the minimum valid label in the User Accreditation Range instead of the `ADMIN_LOW` label and see the maximum valid label in the User Accreditation Range instead of the `ADMIN_HIGH` label.

Note - It is important to realize that the binary label always remains the same when the `EXTERNAL` view is set. The only difference is that the label is given an alternative name when displayed to hide its real name.

- System Default

If the `System Default` option is selected for an account, whatever value is specified in the `label_encodings(4)` file for the `DEFAULT LABEL VIEW` key word applies to the account.

- Decide whether the role can view labels at all.
- Decide whether the account should be locked after the maximum allowable number of failed logins with a bad password is exceeded.

The default number of bad password entries allowed at a single session is 5. The default behavior is not to lock role accounts if they exceed the number.

The following table shows the Trusted Solaris attributes and where they are assigned to role accounts.

TABLE 5-19 Trusted Solaris Attributes and Where They are Specified for Roles

<p>Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu</p>	<p>Click Administrative Accounts Tool Select Role, Choose Action-->Properties, Trusted Solaris Attributes Tab</p>
<p><i>(Trusted Solaris Attributes not available; Minimum Label of ADMIN_LOW, Clearance of ADMIN_HIGH, Internal Label View and Show Labels are assigned by default)</i></p>	<p>Labels Minimum Label Clearance View: External Internal Label: Show Hide <i>(Trusted Solaris Attributes tab is available to secadmin; not available to admin)</i></p>
<p><i>(Account Usage is not available)</i></p>	<p>Account Usage Lock account after maximum failed logins: No Yes</p>

Assigning Audit Classes to Roles

To assign audit classes to a role, the security administrator role needs to modify the role's properties after creating the role.

When a user assumes a role, any system-wide audit flags from the `audit_control` file are combined with any audit classes assigned to the account in the `audit_user` file to form an audit preselection mask:

`(flags in audit_control + user's always-audit flags) - user's never-audit flags`

See the `audit_control(4)` and `audit_user(4)` man pages for more about how flags that are assigned to individual accounts combine with any system-wide flags in the `audit_control(4)` file to specify which actions are audited. Also see *Trusted Solaris Audit Administration* for more about auditing.

The available audit classes are displayed in a scrolling list. The classes are sorted into three categories that determine the circumstances in which they are audited. A class can be audited for the following three types of outcomes:

- Both Success and Failure
- Success only
- Failure only

The following buttons are used to move audit classes into the Included List under three different categories:

- Always Audit
- Never Audit
- Force Except

`Force Except` ensures that the specified class is never audited for the user, and is used to override preceding settings of previously-specified flags.

The following table shows the audit options and where they are assigned to roles.

TABLE 5-20 Audit Options for Roles and Where They are Specified

Click Administrative Roles Tool Choose Add Administrative Role from the Action Menu	Click Administrative Accounts Tool Select Role, Choose Action-->Properties, Audit Tab
<i>(Audit options not available)</i>	User Audit Classes: Move Classes from the categories Both Success and Failure, Success Only, Failure Only from the Excluded list to the Included list using the Add to Always Add to Never Force Except buttons. <i>(Audit tab is available to secadmin; not available to admin role)</i>

Adding or Modifying a Rights Profile

This section describes how to use the Rights tool in the SMC Users collection of tools to create a new rights profile with one or more authorizations, commands, and actions and with optional security attributes for any commands and actions. This section also describes how to modify an existing profile. The following table shows

the name of the tabs in the Rights manager with links to the sections where the attributes are described.

TABLE 5-21 Rights Profiles

Creating a New Right	Section
General	“Specifying General Attributes of Rights” on page 145
Commands	“Assigning Commands to Rights” on page 146
Actions	“Assigning Actions to Rights” on page 147
Authorizations	“Assigning Authorizations to Rights” on page 149
Supplementary Rights	“Assigning Supplementary Rights to Rights” on page 149

From the SMC Users tools, click the Rights tool and click Action->Add Right

Specifying General Attributes of Rights

The following table shows the general rights attributes and where they are entered.

TABLE 5-22 Attributes Set in the Rights Manager General Tab

Name :

(not modifiable for an existing action)

Description:

Help File Name:

For each rights profile you plan to create, decide on the following:

- Decide on a Name.
- Decide on a Description.
- Decide on a Help File Name.

Enter the name of the help file that describes the rights profile. This description displays when the administrator clicks on the name of the rights profile in one of

the SMC Users tools. Create the help file and put it in `/usr/lib/help/profiles/locale/C/`. See “To Create a Help File for a Rights Profile” on page 152.

Assigning Commands to Rights

The following table shows the fields on the Commands Tab.

TABLE 5-23 Attributes Set in the Rights Manager Commands Tab

Commands Denied:	Commands Permitted:	
Add Directory:	Set Security Attributes	
	Ownership	Extended Attributes
	User: Effective Real Group: Effective Real	Label: Clearance: Privileges:

- Decide which commands are needed.

If the directory where a command resides is not already in the list of available commands (in the `Commands Denied` column), enter the pathname to the directory in the `Add Directory:` field.

To move or remove individual commands or directories of commands to or from the `Commands Permitted` list, select the command or directory and click `Add` or `Remove`. Click `Add All` or `Remove All` to move all commands from one column to the other.

- Decide which security attributes are needed for the command.

See the Trusted Solaris man pages for individual commands for the security attributes needed by the command or any of its options to succeed.

If you do not specify any security attributes, a command in a rights profile runs normally, with the real UID/GID, the effective UID/GID, the label, and the clearance of the process that is executing the command and with no inherited privileges.

Click the `Set Security Attributes` button and enter the information requested in the help for the Ownership and Extended Attributes areas.

- **Read and Effective User or Group**

The user name specified in this dialog determines the UID and the group name determines the GID.

The command runs with the real or effective user or group ID associated with the user or group you specify here, instead of running with the real and effective user and group IDs of the user executing the command. Similar to the `setuid` and `setgid` feature of UNIX commands, assigning a real or effective UID or GID to a command in a rights profile allows the command to succeed when it requires a different real or effective UID or GID from the UID or GID of the person who launched the command. The UID most often required is 0, the UID of root (superuser). For example, most installation programs check that they are being run with a real UID of 0.

By adding the name of an installation program to a rights profile, assigning to the program a real UID of 0, and then assigning the profile to a role, the Security Administrator can enable an installation program to succeed when run by a role that has another UID, such as the System Administrator role, which typically has UID 100.

- **Label**

Clicking the `Label Edit` button brings up a label builder. The command is executed with the specified label as the process's label.

- **Clearance**

Clicking the `Clearance Edit` button brings up a Clearance Builder . The command is executed with the specified clearance as the process clearance.

- **Privileges**

Clicking the `Privileges Edit` button brings up a Privilege Chooser. The process running the command is executed with the specified inherited privileges.

Assigning Actions to Rights

The following table shows the fields on the Actions Tab

TABLE 5-24 Attributes Set in the Rights Manager Actions Tab

Actions Denied:	Actions Permitted:	
	Set Security Attributes button	
	Ownership	Extended Attributes
	User: Effective Real Group: Effective Real	Label: Clearance: Privileges:

- Decide which actions are needed.

To move or remove individual actions to or from the Actions Permitted list, select the action and click Add or Remove. Click Add All or Remove All to move all actions from one column to the other.

- Decide which security attributes are needed for the action.

Click the Set Security Attributes button and enter the information requested in the online help for the Ownership and Extended Attributes areas.

If you do not specify any security attributes, an action in a rights profile runs normally, with the real UID/GID, the effective UID/GID, the label, and the clearance of the process that is executing the action and with no inherited privileges.

- Read and Effective User or Group

The action runs with the real or effective group ID associated with the group you specify here, instead of running with the real and effective group IDs of the process executing the action. Similar to the `setuid` and `setgid` feature of UNIX commands, assigning a real or effective UID or GID to an action in a rights profile allows the command to succeed when it requires a real or effective UID or GID that is different from the UID or GID of the person who launched the action. The user name specified in this dialog determines the UID, and the group name determines the GID.

- Label

Clicking the label `Edit` button brings up a label builder. The action is executed with the specified label as the process's label.

- Clearance

Clicking the clearance `Edit` button brings up a Clearance Builder . The action is executed with the specified clearance as the process's clearance.

- Privileges

Clicking the privileges `Edit` button brings up a Privilege Chooser. The process running the action is executed with inherited privileges from the privilege(s) specified here.

Assigning Authorizations to Rights

The following table shows the fields on the Authorizations Tab

TABLE 5-25 Attributes Set in the Rights Manager Authorizations Tab

Authorizations Excluded:	Authorizations Included:
--------------------------	--------------------------

- Decide which authorizations are needed.

To move or remove individual authorizations to or from the `Authorizations Included` list, select the authorization and click `Add` or `Remove`. Click `Add All` or `Remove All` to move all authorizations from one column to the other.

Choose between `View As Names` or `Descriptions`.

Assigning Supplementary Rights to Rights

The following table shows the fields on the Supplementary Rights Tab

TABLE 5-26 Attributes Set in the Rights Manager Supplementary Rights Tab

Rights Excluded:	Rights Included:
------------------	------------------

- Decide whether supplementary rights profiles are needed.

To move or remove individual rights to or from the `Rights Included` list, select the right and click `Add` or `Remove`. Click `Add All` or `Remove All` to move all rights from one column to the other.

Procedures

▼ To Create a User Template

1. **Bring up the SMC Users tool.**
2. **Click the User Templates tool.**
3. **Choose Add User Template from the Action menu and follow .**
Refer to the SMC help for how to create the template. Also see “Adding or Modifying a User Account” on page 121 and its subsections for guidance on what to specify in each field.
4. **Click OK when finished to save the template and exit from the tool.**

▼ To Add or Modify a User Account

1. **Bring up the SMC in the desired scope and double-click the Users tool**
2. **Click the User Accounts tool.**
All configured users are displayed as icons labeled with their usernames.
3. **If adding a new user, choose one of the following from the Action menu:**
 - Add User->With Wizard
 - Add User->From Template
 - Add Multiple Users->With Wizard
 - Add Multiple Users->From Template

Note - To use the From Templates options, you need to first create a template, as described in “To Create a User Template” on page 150.

Refer to the SMC help for how to create the new user. Also see “Adding or Modifying a User Account” on page 121 and its subsections for guidance on what to specify in each field.

4. **To modify an existing user, click to highlight the username, and choose Properties from the Action menu.**

Refer to the SMC help for how to modify the user's properties. Also see "Adding or Modifying a User Account" on page 121 and its subsections for guidance on what to specify in each field.

5. **When you are done, click OK to save your work and exit the tool.**

▼ To Add or Modify a Role

1. **Bring up the SMC in the desired scope and double-click the Users tool**

2. **Click the Administrative Roles tool.**

All configured roles are displayed as icons labeled with their rolenames.

3. **If adding a new role, choose Add Administrative Role from the Action menu:**

Refer to the SMC online help for how to create the new role. Also see "Adding or Modifying a Role Account" on page 136 and its subsections for guidance on what to specify in each field.

4. **To modify an existing role, click to highlight the rolename, and choose Properties from the Action menu.**

Refer to the SMC online help for how to modify the role's properties. Also see "Adding or Modifying a Role Account" on page 136 and its subsections for guidance on what to specify in each field.

5. **When you are done, click OK to save your work and exit the tool.**

▼ To Add a Rights Profile

1. **Bring up the SMC in the desired scope and double-click the Users tool.**

2. **Click the Rights tool.**

Refer to the SMC online help for how to create the new role. Also see "Adding or Modifying a Role Account" on page 136 and its subsections for guidance on what to specify in each field.

3. **Click Action menu Add Right option.**

4. **Click OK to save the new rights profile.**

▼ To Modify a Rights Profile

1. **Bring up the SMC in the desired scope and double-click the `Users` tool.**
2. **Click the `Rights` tool.**

Refer to the SMC online help for how to create the new role. Also see “Adding or Modifying a Role Account” on page 136 and its subsections for guidance on what to specify in each field.
3. **Click the name of an existing right and choose `Properties` from the `Action` menu.**
4. **Enter the information in the tabs.**

See “Adding or Modifying a Rights Profile” on page 144 for tips about the information you need to provide in the tabs, if needed.
5. **Click OK to save the changed rights profile.**

To Create a Help File for a Rights Profile

1. **Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.**
2. **Using the Admin Editor, open a new help file in the directory `/usr/lib/help/profiles/locale/C/`.**
 - a. **Make sure the filename ends with `.html`.**

For example, `FilePriv.html`.
 - b. **In the help file, describe the rights profile you have added.**
 - c. **Make sure the file starts with `<HTML>`, brackets the main text with `</BODY>`, and ends with `</HTML>`—including the angle brackets.**
 - d. **Separate paragraphs with `<P>`.**
 - e. **The text just before `</BODY>` and `</HTML>` should read: "If `right_name` is grayed, then you are not entitled to add or remove it."**

See the following screen example.

```
<HTML>
<HEAD>
Copyright (c) 2000 by Sun Microsystems, Inc. All rights reserved.
<!-- SCCS keyword #pragma ident "%Z%M% %I% %E% SMI; TSOL 2.x" -->
<!-- FilePriv.html -->
</HEAD>
```

(continued)


```
<BODY>
Allows a user to specify the allowed and forced privileges to be associated
with the execution of a program file.
<P>
If the name of the file privileges rights profile is grayed, you are not
allowed to add or remove it.
</BODY>
</HTML>
```

3. Save and quit the new help file.

```
:wq
```

4. Enter the name of the help file in the Help File Name: field on the General tab when creating the right.

See the “Specifying General Attributes of Rights” on page 145 and “To Add a Rights Profile” on page 151, if needed.

Managing Mail

Managing mail is essentially the same in the Trusted Solaris environment as it is in the Solaris environment. The administrator role sets up and administers mail servers according to instructions in the *Solaris System Administration Guide, Volume 2* and *System Administration Guide, Volume 3*, with some differences specific to Trusted Solaris requirements. The System Administrator role administers mail using the Mail Management rights profile. The Trusted Solaris differences are described on the `sendmail(1M)` man page and in this chapter.

This chapter covers the following topics:

- “Managing Trusted Solaris Mail Features” on page 156
- “Allowing Users to List the Entire Mail Queue” on page 157
- “To Allow Listing of the Mail Queue” on page 157
- “To Trace sendmail for Trusted Solaris Information” on page 160
- “To Check for a Properly Configured Network Connection for Sending Mail” on page 161
- “Configuring Trusted Solaris Mail Delivery Options for Mail Below Users’ Minimum Labels” on page 165
- “To Configure Mail Delivery Options for Mail Below Users’ Minimum Labels” on page 166
- “Substituting an Alternate Mail Application” on page 166
- “To Substitute an Alternate Mail Application in the Front Panel for All Users ” on page 168
- “To Create a Multilevel Action for the Alternate Mail Application ” on page 171
- “To Install an Alternate Mailer in the Front Panel” on page 175

Managing Trusted Solaris Mail Features

The following table shows the aspects of managing mail that are different in the Trusted Solaris environment and how to change them.

TABLE 6-1 Features Affecting Mail and How to Change Defaults

Trusted Solaris Feature	Implementation	How to Change
Users' home directories are MLDs. Users' .mailrc files are stored by default only in the SLD at the user's minimum label.	Any .mailrc file created at the user's minimum label needs to be copied or linked into all other SLDs at all the labels at which the user works. Either the security administrator or the individual user can do the set up by listing the file in either <code>copy_files</code> or <code>link_files</code> . See <code>updatehome(1M)</code> for a description of <code>copy_files</code> and <code>link_files</code> .	See "Managing Initialization Files" on page 91. Also see the Mail Aliases section in "Introduction to Mail Services" in the Solaris <i>System Administration Guide, Volume 3</i> for background about mail aliases, if needed.
System wide mail aliases are managed using the Solaris Management Console (SMC) Mailing Lists tool.	Depending on the scope of the selected SMC toolbox, either the local <code>/etc/aliases</code> file, the NIS map <code>mail.aliases</code> , or the NIS+ <code>mail_aliases</code> table is updated.	See the help in the SMC Mailing Lists tool for how to modify existing aliases or create new ones.
A user should not be able to list queued mail sent by other users.	The <code>restrictmailq</code> privacy option is set by default in the <code>sendmail.cf</code> file, and therefore only users in the same group as the mail queue may list jobs in the mail queue.	See "To Allow Listing of the Mail Queue" on page 157 for how to change the default.
Administrators may need to use the <code>sendmail -d</code> option to debug Trusted Solaris-specific problems.	Category 75 in the Trusted Solaris version of <code>sendmail</code> can be specified to select Trusted Solaris debugging information.	See "Tracing sendmail's Activities" on page 158 and "To Trace sendmail for Trusted Solaris Information" on page 160.

TABLE 6-1 Features Affecting Mail and How to Change Defaults (continued)

Trusted Solaris Feature	Implementation	How to Change
A user cannot read email sent at a label below the user's minimum label.	The <code>sendmail.cf</code> file has been extended with <code>tsol</code> options. By default, <code>ADMIN_LOW</code> -labeled mail is upgraded to the user's minimum label and mail at the other labels below the user's minimum label is returned.	The security administrator role must make sure that the Trusted Solaris options in the <code>sendmail</code> configuration file <code>sendmail.cf</code> have values consistent with the site's security policy. See "Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels" on page 165 and "To Configure Mail Delivery Options for Mail Below Users' Minimum Labels" on page 166 for how to change the options.
<code>dtmail(1)</code> is the default mail reader	The <code>dtmail</code> action is in the Mail subpanel of the Front Panel.	See "Substituting an Alternate Mail Application" on page 166 for how to substitute an alternate mail reader.

Allowing Users to List the Entire Mail Queue

▼ To Allow Listing of the Mail Queue

Note - Listing of the mail queue is done either by entering the `mailq` command or the equivalent command, `sendmail` with the `-bp` option. Even after an administrator performs one of the steps described below to allow listing of the mail queue, users cannot list the mail queue unless they have the `mailq` or `sendmail` command in one of their profiles. These commands show mail only at labels dominated by the user's process. These commands are in the Mail Management profile.

1. Use the Set Mail Options action to open the `sendmail.cf` file for editing. See “Accessing the Administration Tools” on page 29, if needed.
2. Search for the `restrictmailq` option in the file.

```
# Privacy flags
O PrivacyOptions=authwarnings,restrictmailq
```

3. Remove the `restrictmailq` option.

```
# Privacy flags
O PrivacyOptions=authwarnings
```

4. Save and quit the file.

Troubleshooting Mail Problems

Tracing `sendmail`'s Activities

Multiple instances of `sendmail` are involved in local and remote mail delivery. To aid in debugging any problems with `sendmail`, Figure 6-1 shows how data flows through the `sendmail` processes.

Any mailer that is used to send mail (the default is `dtmail`) starts an instance of `sendmail`. This instance of `sendmail` attempts to deliver any mail that originates on the host, storing it in the local `/var/spool/mqueue` MLD until it is delivered—in case the system crashes or anything else causes the mail message not to be delivered [1 in Figure 6-1 shows this instance of `sendmail`]. Normally the message is delivered right away so its stay in the queue is only a matter of seconds. However, if the remote host is down, mail can stay in the queue indefinitely.

An instance of the `sendmail` program also starts when the workstation or server is booted, and this instance of `sendmail` listens at port 25 and attempts to deliver any mail that it receives from a remote host, also storing each message in the mail queue MLD until it is delivered [3 and 5 in the example].

Yet another instance of `sendmail` periodically scans the mail queue and attempts to deliver any mail in the queue [2 and 4 in the example]. The following figure shows some of the `sendmail` processes on three hosts: `cascade`, `trustworthy`, and `juggle`. Host `trustworthy` is the mail relay host for `juggle`.

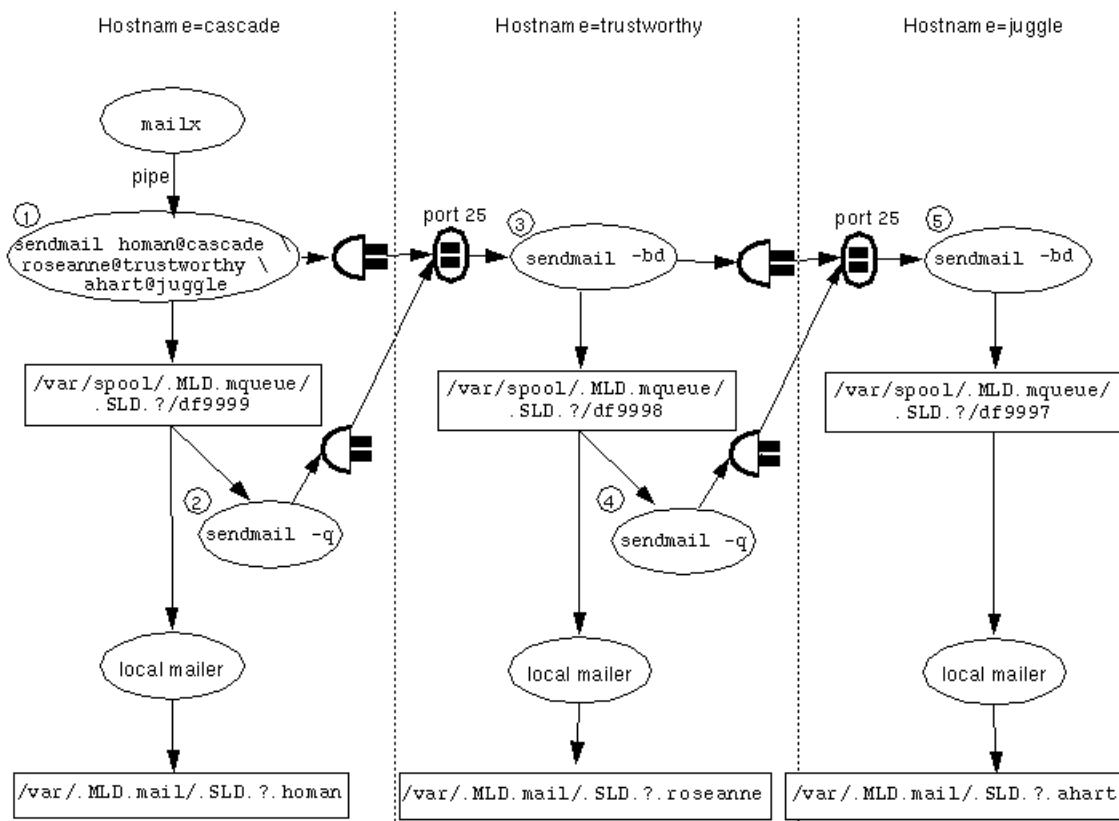


Figure 6-1 Sendmail Data Flow Example

When mail is sent to `username@hostname` and `hostname` is a remote host, `sendmail` forwards the message to port 25 of `hostname`. In the example, when mail addressed to `homan@cascade` is sent from another account on host `cascade`, `sendmail #1` puts the mail into an SLD within the `/var/spool/.MLD.mqueue` on `cascade`, where it is delivered by a local mailer. `sendmail #2` on `cascade` periodically polls the queue and delivers mail that could not get delivered right away. `sendmail #3` and `#5` on hosts `trustworthy` and `juggle` listen on port 25 for incoming mail. The messages originating on `cascade` that are addressed to hosts `trustworthy` and `juggle` are both put into the local `/var/spool/.MLD.mqueue` and sent to port 25 of `trusted`, which is acting as a mail relay host in this example. The `sendmail #3` on `trusted` also puts both messages into an SLD within the local `/var/spool/mqueue`, where the message to

roseanne@trustworthy is delivered by the local mailer and the message to ahart@juggle is forwarded to sendmail #5, which is listening at port 25 of juggle.

Debugging sendmail using the `-d` option is described in the *sendmail Nutshell Handbook* published by O'Reilly & Associates, Inc. To review briefly, you can get debugging information by specifying sendmail with the `-d` option followed by *X*. To limit the output of sendmail `-d` to a specific aspect of sendmail's behavior, you can specify a *category* optionally followed by a *dot* (.) followed by a *level* from 0-9, with 9 meaning the most information. Category 75, which is unique to the Trusted Solaris version of sendmail, selects Trusted Solaris debugging information.

▼ To Trace sendmail for Trusted Solaris Information

1. **Assume the system administrator role and go to an ADMIN_LOW workspace.**

See "To Log In and Assume an Administrative Role" on page 32 if needed.

2. **In a profile shell, go to the `/etc/init.d` directory and stop sendmail.**

```
$ cd /etc/rc2.d
$ sendmail stop
```

3. **Enter sendmail with the `-d` option followed by the category 75 optionally followed by a dot (.) and a level, followed by a space and the address, followed by a message.**

A message can be included either by redirecting the contents of a file to the address, as shown below, or by entering return at the end of the line. In the latter case, a `Subject:` prompt comes up; after entering the subject, you can create a message from the command line, using the syntax of `mail(1)`.

```
$ /usr/lib/sendmail -d75.9 roseanne@trusted < /etc/motd
```

4. **Review the error messages.**
5. **In a profile shell at ADMIN_LOW, go to the `/etc/init.d` directory and restart sendmail.**


```
$ cd /etc/init.d
$ sendmail start
```

Tracing Mail Delivery Difficulties

Remember that `sendmail` makes a number of checks about the destined recipient and about the destined receiving host before sending or forwarding mail. Mail can be received by an account only if the mail is between the account's clearance and minimum label. The account's label range is specified as described in Chapter 5. Mail can be received on a host only if the mail is within the accreditation range of that host, as described in the following list and in Chapter 8.

- Multilevel hosts with an accreditation range between `ADMIN_LOW` and `ADMIN_HIGH` can receive mail at all labels.
- Multilevel hosts that have a restricted accreditation range can receive mail only between their maximum and minimum labels.
- Single-level (unlabeled) hosts can receive mail at only the single label set up for them.

If a user is having trouble sending mail, use the following as guidelines for where to look:

- ◆ **Make sure that there is a properly configured network connection between the sending and receiving hosts.**

Do the procedure described in “To Check for a Properly Configured Network Connection for Sending Mail” on page 161.

- ◆ **Check the mail aliases.**

Remember that the local `/aliases` file, the NIS map `mail.aliases`, or the NIS+ `mail_aliases` table are consulted by `sendmail` in determining where to deliver mail, depending on the `nsswitch.conf(4)` entry for `aliases`. For example, mail to `fred` from a process on his Trusted Solaris workstation `xxx` would not go to `fred@xxx` if `sendmail` consults the `mail_aliases` table and finds an alias of `fred@yyy` in that table for user `fred`.

▼ To Check for a Properly Configured Network Connection for Sending Mail

1. **Send mail using `mailx`.**

```
# mailx -v somebody@somehost
Subject: test1
test1
.
```

Review the messages from mailx.

2. **Log into the sending host or, if the mail server is not the same as the sending host, log into the mail server at the label at which the user needs to send mail.**
3. **Use the telnet(1) command to connect to port 25 of the receiving host.**

```
trustworthy% telnet hostname 25
```

If the connection is properly set up with the correct labels in the trusted networking databases for the sending and the receiving hosts, the sendmail on the destination host prints a message like:

```
220 hostname Sendmail version ready at date
```

Enter quit to end the connection.

```
quit
```

If the connection seems to be set up properly, go to the following step.

If you get an error message from telnet, the connection is not properly set up; go to the step shown in the following table that applies to the type of host you are trying to debug.

Trusted Solaris	Step 6 on page 163 and Step 7 on page 164
label-cognizant non-Trusted Solaris operating environment	Step 8 on page 164
unlabeled operating environment (such as Solaris)	Step 9 on page 165

4. **At the label of the outgoing mail, list the mail queue on the sending host or, if the mail server is not the same as the sending host, list the mail queue on the mail server.**

```
# mailq | more
```

Check the list to see if the mail is stuck on the mail server.

5. **Try the procedure under “To Trace sendmail for Trusted Solaris Information” on page 160.**
6. **If the destination host is running Trusted Solaris 2.x or later, do these steps to make sure the destined user is able to receive mail within Trusted Solaris security policy:**
 - a. **Make sure the destined recipient has a valid user account (if needed, in Trusted Solaris 8 use the `SMC User Accounts` tool, in previous releases use the `Solstice User Manager`).**
 - b. **Note the account’s minimum label and clearance in the Trusted Solaris Attributes tab in the `User Properties` dialog in the `User Accounts` tool.**
 - c. **Make sure that the label of the mail is within both the User Accreditation Range and the System Accreditation range of the destination host as defined by the `label_encodings(4)` file.**

`sendmail` does not deliver mail if the label of the mail is outside the System Accreditation Range.

If the label of the mail is inside the System Accreditation Range but outside the User Accreditation Range, such as mail sent at `ADMIN_LOW` and `ADMIN_HIGH`, remember that a normal user, by default, cannot receive mail sent outside of the User Accreditation Range, and go on to Step 7 on page 164.
 - d. **Make sure that the label of the mail being sent is dominated by the recipient’s maximum label and dominates the recipient’s minimum label.**
 - If the label of the mail being sent is not in the recipient’s account label range, if you can find a mutually-acceptable label for the sender and the recipient, change the label to one within the destined recipient’s label range and try again.
 - If the mail goes through, instruct the sender to send mail to that recipient at the mutually-acceptable label.
 - e. **If the mail is below the minimum label of the destined user, change the default `tsol` options in the `sendmail.cf` file, if doing so is consistent with your site’s security policy.**

See “Configuring Trusted Solaris Mail Delivery Options for Mail Below Users’ Minimum Labels” on page 165 and “To Configure Mail Delivery Options for Mail Below Users’ Minimum Labels” on page 166.
 - f. **To enable anyone to receive mail from system processes outside the User Accreditation Range if the `tsoladminlowaccept` or `tsolotherlowreturn` option are used, use the `Rights Manager` in**

Trusted Solaris 8 or the Profile Manager in previous releases to make sure that the user has the `solaris.label.range` authorization.

The default administrative roles have the needed authorization in their profiles.

7. **For a destination host running the Trusted Solaris operating environment, on the sending host make sure that the `tnrhdb(4)/tnrhttp(4)` entries for the receiving host are configured properly.**

Note - You can use the `tninfo(1M)` command to check which template has been assigned which host and which host type and other attributes are assigned in the template. The `-h hostname` option lists the name of the template assigned to the specified host, while the `-t template_name` option lists the entries specified in the template, including the host type.

- a. **Check that the destination host has the correct template name assigned to it in the `tnrhdb(4)` database and that the template in the `tnrhttp(4)` file correctly defines the host's type as `sun_tsol`.**
 - b. **Check that the minimum and maximum label set in the assigned template in `tnrhttp` allow communications at the label of the mail that is not being delivered.**
 - c. **Once these checks are passed, the network connection ought to work. Go back and do Step 3 on page 162 and run `telnet(1)` to make sure.**
8. **For a destination host running any labeled operating system that is not Trusted Solaris, on the sending host make sure that the `tnrhdb/tnrhttp` entries for the receiving host are configured properly so that the Trusted Solaris system can communicate with that host over the network.**
 - a. **Read the `tnrhttp(4)` man page if necessary to find out the correct host type and other options to specify in the template assigned to the host.**

For example, CIPSO type hosts require certain options, and RIPSO type hosts require other options. See also Chapter 8.
 - b. **Create a template or copy an applicable one in the `tnrhttp` and make sure that the correct template is assigned to the host in the `tnrhdb` database to identify it with the appropriate host type.**
 - c. **Check that the minimum and maximum label set in the assigned template in the `tnrhttp` allow communications at the label of the mail that is not being delivered.**

- d. Once these checks are passed, the network connection ought to work. Go back and do Step 3 on page 162 and run `telnet` to be sure.
9. If the destination host is not running a label-cognizant operating system, on the sending host make sure that the `tnrhdb/tnrhtp` entries for the receiving host are configured properly.
 - a. Check that the destination host has the correct template name assigned to it in the `tnrhdb` database that the template in the `tnrhtp` file correctly defines the host's type as `unlabeled`.
 - b. Check that the default label for the unlabeled host in the assigned template in the `tnrhtp` allows communications at the label of the mail that is not being delivered.
 - c. Once these checks are passed, the network connection ought to work. Go back and do Step 3 on page 162 and run `telnet` to be sure.

Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels

The security administrator role must make sure that the Trusted Solaris-specific privacy options in the `sendmail(1M)` configuration file `sendmail.cf` have values consistent with the site's security policy. If no option is specified, the default is to automatically upgrade mail sent at `ADMIN_LOW` and return to the sender any mail sent at other labels below the user's minimum label. The behavior is the same as the commented-out lines in the `sendmail.cf` file.

```
#0 LabelTooLow=return
#0 LabelAdminLow=upgrade
```

`ADMIN_LOW` mail is treated differently from other mail because `ADMIN_LOW` mail is always sent by a system process to an account (usually an administrative role account) that should see the mail, while a user cleared to a particular label in the user accreditation range, such as `CONFIDENTIAL` or `INTERNAL USE ONLY`, should

probably not be able to send mail to a user whose minimum label dominates the first user's label, such as `SECRET` or `NEED TO KNOW`.

`upgrade` means to deliver the message at the recipient's minimum label. `accept` means to deliver the message at the message's label. `return` means to return the message to the sender.

▼ To Configure Mail Delivery Options for Mail Below Users' Minimum Labels

- 1. Assume the security administrator role and go to an `ADMIN_LOW` workspace.**
See "To Log In and Assume an Administrative Role" on page 32, if needed.
- 2. Use the `Set Mail Options` action to open the `sendmail.cf` file for editing.**
See "To Launch Administrative Actions from a Local Application Manager" on page 43, if needed.
- 3. Search for the lines that begin `-Optsol`, and change either of the two existing default settings.**
See "Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels" on page 165 for names and descriptions of the `tsol` privacy options.

```
# TSOL: Incoming mail below recipient's minimum label
# Possible values are return, upgrade, or accept
#0 LabelTooLow=return
# Special case for mail labeled admin_low
#0 LabelAdminLow=upgrade
```

Substituting an Alternate Mail Application

By default, `dtmail` is the mail application that is launched from the `Mail` panel on the Trusted Solaris `Front Panel`. The Trusted Solaris system allows the substitution of an alternate mail application. Only the system administrator role can do the set up needed so that the mailer provides the full multilevel mail capabilities.

Without administrative intervention, any user can drag and drop an action for an alternate mail application into the Front Panel and then access the newly-installed mailer at the label of the current workspace. However, since mail monitoring at multiple labels does not occur when an action is installed this way, dragging and dropping by individual accounts of alternate mail actions into the Front Panel is only appropriate at a site using a single label.

The system administrator role can either

- Modify the Front Panel control file so that an alternate mail action is available to all users (see “To Substitute an Alternate Mail Application in the Front Panel for All Users ” on page 168).
- Make an alternate mail action control file available with instructions for individual users on how to drag and drop the alternate control file into their Front Panel mail subpanel (see “To Create a Multilevel Action for the Alternate Mail Application ” on page 171).

Before an alternate mail action can be installed in the front panel, an application must first be defined for the mail application. The example in “To Substitute an Alternate Mail Application in the Front Panel for All Users ” on page 168 shows the substitution of the OpenWindows mailtool for Dtmmail, even though it is unlikely that this substitution would be made. The example uses the OpenWindows mailtool action is defined in the /usr/dt/appconfig/types/C/sunOW.dt file as shown in the following example.

CODE EXAMPLE 6-1 OpenWindow’s mailtool Action Definition from sunOW.dt

```
ACTION OWmailtool
{
    LABEL           OW Mail Tool
    ICON            OWmailtool
    TYPE            COMMAND
    WINDOW_TYPE     NO_STDIO
    EXEC_STRING     /usr/openwin/bin/mailtool
}
```

Follow the example to substitute an alternate mailtool action of your own creation.

Precaution When Modifying the Default Mail Icon

If mail arrives while you are installing an alternate mail icon or deleting the default one, problems can arise. For that reason, it is a good idea to stop sendmail before you begin and start sendmail again after you are done.

If all the mail icons disappear from the Front Panel, investigate the account’s .dt/ fp.dynamics directory in the home directory. During the operation of the system, all changes to the Front Panel are stored in each account’s \$HOME/.dt/

fp.dynamics directory at the session clearance. Copy the contents of fp.dynamics to a backup directory and restore the file one by one until the Front Panel configuration is restored.

▼ To Substitute an Alternate Mail Application in the Front Panel for All Users



Caution - Do this procedure before accounts start getting mail on the system. If you do it later, you will need to clean up the contents of directories created by the window system in every .dt/fp.dynamics directory in every SLD in every home directory MLD.

1. **Assume the system administrator role, and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **In a terminal, go to the /etc/init.d directory and stop sendmail.**

```
$ cd /etc/init.d
$ sendmail stop
```

3. **Assume the security administrator role, and go to an ADMIN_LOW workspace.**
4. **Make sure an action is defined for the alternate mail action.**
See “Substituting an Alternate Mail Application” on page 166, if needed.
5. **Use the Admin Editor action from the System_Admin folder in the Application Manager to open the /usr/dt/appconfig/types/C/dtwm.fp for editing.**
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
6. **Find the CONTROL Mail section shown below.**

```
CONTROL Mail
{
  TYPE          icon
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  POSITION_HINTS      5
```

(continued)


```
ICON          Dtmail
LABEL         Mail
ALTERNATE_ICON      DtMnew
MONITOR_TYPE      mail
DROP_ACTION       Compose
PUSH_ACTION       DTWmail
PUSH_RECALL       true
CLIENT_NAME       dtmail
HELP_TOPIC        FPOnItemMail
HELP_VOLUME       FPanel
}
```

- a. Leave TYPE, CONTAINER_NAME, CONTAINER_TYPE, and POSITION_HINTS as shown below.**

```
TYPE          icon
CONTAINER_NAME      Top
CONTAINER_TYPE      BOX
POSITION_HINTS      5
```

- b. Change the ICON field to identify the icon of the replacement application.**

```
ICON          OWmailtool
```

- c. Change the LABEL field to change the icon label that appears with the icon of the replacement application in the mail subpanel.**

```
LABEL         OW Mail Tool
```

- d. Leave ALTERNATE_ICON and MONITOR_TYPE as shown below.**

```
ALTERNATE_ICON      DtMnew
MONITOR_TYPE         mail
```

- e. **Change DROP_ACTION or leave as shown below.**

DROP_ACTION	Compose
-------------	---------

Other mailers may or may not have a `Compose` action. OpenWindows mailtool does not. If you leave the `DROP_ACTION` as `Compose`, if someone drags mail to the mail icon, a `dtmail` `Compose` window will come up. If you remove the `DROP_ACTION`, nothing happens if mail is dragged to the mail icon.

- f. **Change the PUSH_ACTION field to identify the replacement action to be run when the user clicks on the new mail icon.**

PUSH_ACTION	OWmailtool
-------------	------------

The action name supplied here must be defined in the one of the application search paths. The `OWmailtool` action shown is defined in `sunOW.dt` in the `/usr/dt/appconfig/types/C` directory.

- g. **Leave the PUSH_RECALL action as shown.**

PUSH_RECALL	true
-------------	------

When `true`, if an application is launched for a second time, a new application is not launched if the icon for the application window is concealed on the workspace. Instead the application window is brought forward.

- h. **Change the CLIENT_NAME field to identify the executable for the replacement application.**

CLIENT_NAME	mailtool
-------------	----------

The path for `CLIENT_NAME` must be defined by an `EXEC_STRING` in the action's definition. For example, the `OWmailtool` action has the `EXEC_STRING` defined as `/usr/openwin/bin/mailtool`.

- i. **Leave the HELP_* entries as is.**

HELP_TOPIC	FOnItemMail
HELP_VOLUME	FPanel

7. Save the changes and close the file.

:wq

Note - The next step is only necessary if you do this procedure after the system is running.

8. **Remove all contents of the `$HOME/.dt/fp.dynamics` directory.**
9. **Restart the Workspace Manager from the workspace menu to see the changes to the `dtwm.fp` go into effect in the front panel.**
10. **Assume the system administrator role and go to an `ADMIN_LOW` workspace.**
11. **In a terminal emulator such as `dtterm(1)`, go to the `/etc/init.d` directory and restart `sendmail`.**

```
$ cd /etc/init.d
$ sendmail start
```

▼ To Create a Multilevel Action for the Alternate Mail Application

1. **Assume the security administrator role, and go to an `ADMIN_LOW` workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Use the Admin Editor action to bring up the `/usr/dt/appconfig/types/C/dtwm.fp` file to edit.**
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
3. **Find the control section for mail shown below.**

```
CONTROL Mail
{
  TYPE          icon
  CONTAINER_NAME      Top
  CONTAINER_TYPE      BOX
  POSITION_HINTS      5
  ICON             DTmail
  LABEL            Mail
```

(continued)

```

ALTERNATE_ICON      DtMnew
MONITOR_TYPE        mail
DROP_ACTION         Compose
PUSH_ACTION         DTWmail
PUSH_RECALL         true
CLIENT_NAME         dtmail
HELP_TOPIC          FPonItemMail
HELP_VOLUME         FPanel
}

```

4. Copy the control text to a file whose name has the .fp extension, for example, mail.fp, and quit the dtwm.fp file.

Note - Create the new file mail.fp file in a directory such as /etc or /usr/bin that is in every user's path.

```

:wm

```

5. Bring up the Admin Editor action from the System_Admin folder and open the new mail.fp file for editing.

6. Edit the CONTROL OW_Mail section shown below.

```

CONTROL OW_Mail
{
  TYPE          icon
  CONTAINER_NAME      Top
  CONTAINER_TYPE     BOX
  POSITION_HINTS      5
  ICON             DTmail
  LABEL            Mail
  ALTERNATE_ICON     DtMnew
  MONITOR_TYPE       mail
  DROP_ACTION        Compose
  PUSH_ACTION        DTWmail
  PUSH_RECALL        true
  CLIENT_NAME        dtmail
  HELP_TOPIC         FPonItemMail
  HELP_VOLUME        FPanel
}

```

(continued)

```
}
```

- a. Leave TYPE, CONTAINER_NAME, CONTAINER_TYPE, and POSITION_HINTS as shown below.**

```
TYPE          icon
CONTAINER_NAME      Top
CONTAINER_TYPE      BOX
POSITION_HINTS      5
```

- b. Change the ICON field to identify the icon of the replacement application.**

```
ICON          OWmailtool
```

- c. Change the LABEL field to change the icon label that appears with the icon of the replacement application in the mail subpanel.**

```
LABEL          OW Mail Tool
```

- d. Leave ALTERNATE_ICON and MONITOR_TYPE as shown below.**

```
ALTERNATE_ICON      DtMnew
MONITOR_TYPE         mail
```

- e. Change DROP_ACTION or leave as shown below.**

```
DROP_ACTION          Compose
```

Other mailers may or may not have a Compose action. For example, OpenWindows mailtool does not. If you leave the DROP_ACTION as Compose, if someone drags mail to the mail icon, a dtmail Compose window will come up. If you remove the DROP_ACTION, nothing happens if mail is dragged to the mail icon.

- f. Change the PUSH_ACTION field to identify the replacement action to be run when the user clicks on the new mail icon.**

PUSH_ACTION	OWmailtool
-------------	------------

The action name supplied here must be defined in the one of the application search paths. The OWmailtool action shown is defined in sunOW.dt in the /usr/dt/appconfig/types/C directory.

g. Leave the PUSH_RECALL action as shown.

PUSH_RECALL	true
-------------	------

When true, if an application is launched for a second time, a new application is not launched if the icon for the application window is concealed on the workspace. Instead the application window is brought forward.

h. Change the CLIENT_NAME field to identify the executable for the replacement application.

CLIENT_NAME	mailtool
-------------	----------

The path for CLIENT_NAME must be defined by an EXEC_STRING in the action's definition. For example, the OWmailtool action has the EXEC_STRING defined as /usr/openwin/bin/mailtool.

i. Leave the HELP_* entries as is.

HELP_TOPIC	FPOnItemMail
HELP_VOLUME	FPanel

7. Save the changes and quit the file.

:wq

8. Give users the procedure “To Install an Alternate Mailer in the Front Panel” on page 175, after testing the procedure yourself to see if the mailer shows up and works properly.

▼ To Install an Alternate Mailer in the Front Panel



Caution - Unless you have the approval of your site's security administrator, do not install an alternate mailer if either of the following apply:

- If the mailer is from any of the application manager folders
 - If the file does not end with a suffix of `.fp`
-

1. **Obtain the pathname of the correct alternate mail application's control file from the `admin`.**

The security administrator and system administrator roles must have completed the procedure in "To Create a Multilevel Action for the Alternate Mail Application" on page 171 before you start.

2. **Ask the system administrator role to stop `sendmail`.**
3. **Using the `File Manager`, change to the directory where the alternate mail application's control file resides.**
4. **Click the `Mailer` subpanel access button to bring up the subpanel.**
5. **Drag the icon for the alternate mailer's front panel control file onto the `Install Icon` dropsite in the `Mail` subpanel.**

The icon for the alternate mail application should appear in the `Mail` slider.
6. **Click the right mouse button while the pointer is over the alternate mail and select `Copy to Main Panel`.**
7. **For each of the old mail icons in the subpanel, click the right mouse button while the pointer is over any of the mail icons for the old application and select `Delete`.**

Repeat this until all of the old icons have been removed. You cannot have a mixture of mail applications running at the same time.
8. **Select `Restart Workspace Manager` from the `Workspace Menu`.**

The size of the subpanel does not adjust correctly until the `Window Manager` is restarted.
9. **Ask the system administrator role to restart `sendmail`.**

Managing Computers and Networks

This chapter provides the necessary concepts and background for administering computers and networks. See the following sections.

- “Managing Trusted Network Communications” on page 177
- “SMC Tools for Administering Computers and Networks” on page 178
- “Meeting the Goals of Trusted Networking ” on page 180
- “Understanding Security Attributes Assigned to Computers” on page 180
- “Default Templates” on page 187
- “Administering Routing” on page 197
- “MAC Enforcement on Outgoing Messages” on page 195
- “MAC Checks on Messages Being Forwarded” on page 195
- “MAC Enforcement on Incoming Messages” on page 196
- “Setting Up Trusted Routing ” on page 201
- “Allowing a Single-label Gateway to Forward Packets at Multiple Labels” on page 202

For an overview of trusted networking, see also “Administering Trusted Networking” in *Trusted Solaris Administration Overview*.

Managing Trusted Network Communications

The Trusted Solaris operating environment supports network communications between Trusted Solaris computers and any of the following types of computers:

- Other computers running the Trusted Solaris operating environment.
- Computers running operating environments that do not recognize security attributes but do support TCP/IP (such as Solaris and other UNIX systems, Windows, and MacOS)
- Computers running other trusted operating systems that recognize some of the Trusted Solaris security attributes

Network communications and services are managed by several Trusted Solaris subsystems.

- Trusted NFS is used to manage mounts of file systems from other computers.

Mounts among Trusted Solaris computers and other computers that recognize NFS are supported. See Chapter 9 for how to set up mounts and specify security attributes when mounting with NFS and other mount protocols, such as UFS, PCFS, and HSFS.

- NIS and NIS+ provide centralized management of configuration files defining hosts, networks and users.

See Chapter 10 for NIS and NIS+ differences in the Trusted Solaris environment.

- The Solaris Management Console is used to centrally manage users, computers, and networks.

The Solaris Management Console supports the maintenance of most administrative data in NIS maps or NIS+ tables on a NIS or NIS+ server and also provides the option of updating the corresponding local files on individual hosts without relying on a naming service.

The *Trusted Solaris Installation and Configuration* manual describes how to add new workstations and servers during configuration of a distributed system. See Chapter 8 for additional details about how to set up security attributes for computers.

- Trusted networking and extended routing software supports trusted network communications.

The Security Administrator role uses SMC tools to manage communications among hosts across networks. See "SMC Tools for Administering Computers and Networks."

SMC Tools for Administering Computers and Networks

The SMC Trusted Solaris Configuration toolbox contains the following tools for configuring computers and networks and defining security attributes for computers, networks, and network interfaces :

- The Computers and Networks tool, which includes:
 - The unmodified Computers tool for adding new computers
 - The unmodified Add Network option for specifying netmasks
 - The Trusted Solaris Security Families tool for assigning security attributes to computers

- The Trusted Solaris Interface Manager tool for assigning security attributes to network interfaces (only needed if the default values are not appropriate)

Note - The Interface Manager modifies the local `/etc/security/tnidb` file, and the tool displays only when the scope of the selected toolbox is Files.

The tools are shown in the following figure.

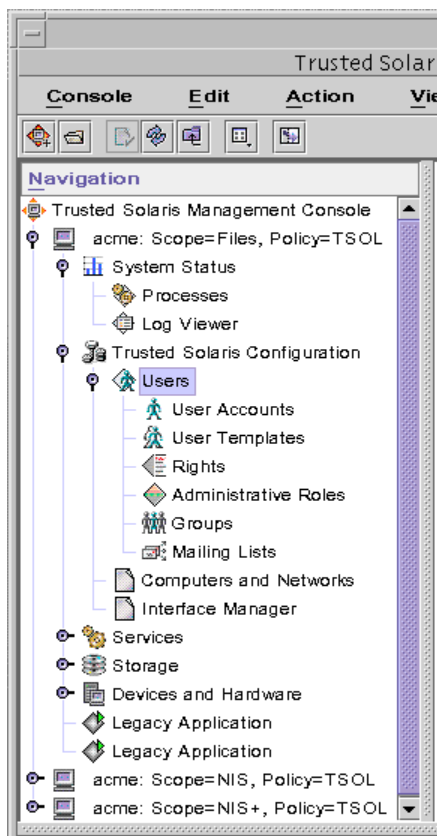


Figure 7-1 SMC Tools

Meeting the Goals of Trusted Networking

The overall goal of trusted networking software is to ensure that the Trusted Solaris security policy is enforced even when the subjects (processes) and objects (data) are located on different hosts.

The security attributes that are assigned to computers and networks and to mounted file systems are used by the trusted network software to help ensure the following:

- Data in network communications is properly labeled.
- Mandatory access control (MAC) rules are enforced when data is sent or received across a local network and when file systems are mounted
- MAC rules are enforced when data is routed to distant networks

Understanding Security Attributes Assigned to Computers

Security attributes are administratively assigned to computers (hosts and routers) by means of templates. The Security Administrator role administers templates and assigns templates to computers using the Security Families tool. If a computer does not have a template assigned, no communications are allowed with that computer. Computers that share the same template are considered to be part of the same security family.

Every template has a `Host Type`, which determines which protocol is used to communicate with the computer that is assigned the template. The protocols tell the kernel which security attributes to look for in the header of an incoming packet or to insert into an outgoing packet. See “Host Types” on page 181.

Every template also has an `Accreditation Range` (consisting of a `Minimum Label` and a `Maximum Label`) and a default `DOI` (Domain of Interpretation). For details about these attributes, see “Computer Accreditation Range” on page 183 and “DOI” on page 183.

Each host type has its own set of additional required and optional security attributes, which are introduced in the following list:

- Templates for the Unlabeled and RIPS0 host types specify a `Default Label` that is used to control communications with computers whose operating systems are not aware of labels, such as Solaris or RIPS0-cognizant operating systems.

Because communications with these computers are essentially limited to the Default Label, they are referred to as single-label computers. See “Default Label” on page 185.

- Templates for single-label computers also specify a Default Clearance. One or more optional privileges can be specified in the template’s Forced Privileges field. See “Default Clearance” on page 185 and “Forced Privileges” on page 185.
- The template for the Trusted Solaris host type has an Allowed Privileges field that can optionally be used to limit the privileges accepted from the remote computer. See “Allowed Privileges” on page 185.
- The template for any host type can be used to specify an IP label to be used in trusted routing of packets. See “Using IP Labels in Trusted Routing” on page 187

For more about specifying an IP label or how to change the default DOI, see “Advanced Security Attributes” on page 186.

Host Types

The following table describes the host types for which entries can be made in the trusted network databases. The first column shows the name used in the Security Families host type menu.

TABLE 7-1 Host Types, Protocols, and Notes

Name in Security Families->Template Manager	Protocols and Notes
Trusted Solaris	<p>The TSOL protocol simplifies passing security attributes between computers running Trusted Solaris 2.x or later compatible releases. TSOL is a derivative of the TSIX(RE) 1.1 - SAMP protocol that passes the security attributes in a similar place in the network protocol stack and uses similar header structures. The TSOL protocol passes security attributes in binary form and thus does not require token mapping. NOTE: For communications between Trusted Solaris computers, either the Trusted Solaris or TSIX host type can be assigned in the templates, depending on whether you want the labels to be transmitted in binary form or in token form. If only the labels' names differ on two computers while the labels' binary representations are the same, the Trusted Solaris host type can be used. If the labels' names are the same but the labels' binary representations are different on both Trusted Solaris computers, the TSIX host type can be assigned.</p>
Unlabeled	<p>This host type is assigned to computers running Solaris or other unlabeled operating systems to specify a default label and default clearance to apply to communications with the unlabeled computer. Also, a minimum and maximum label can be set to allow the sending of packets to an unlabeled gateway for forwarding when the packets' labels do not match the default label and would therefore not be sent to the computer as a destination.</p>
RIPSO	<p>Revised IP Security Option (RIPSO) described in the IETF RFC 1108. It specifies a DoD IP labeling method to incorporate labels into IP packets, which are then used for network mandatory access control checks. A fixed RIPSO label specified in the template is applied to network packets interchanged with the particular host. Though this functionality does not fully meet the RFC specifications, it is expected to supply sufficient functionality where RIPSO labels are needed.</p>

TABLE 7-1 Host Types, Protocols, and Notes *(continued)*

Name in Security Families->Template Manager	Protocols and Notes
CIPSO	Common IP Security Option (CIPSO) protocol TSIX(RE) 1.1 is used to specify security labels that are passed in the IP options field. CIPSO labels are derived automatically from the data's label. Tag type 1 is used to pass the CIPSO security label. This label is then used to make security checks at the IP level and to label the data in the network packet.
TSIX	Trusted Security Information Exchange for Restricted Environments (TSIX/RE) protocol uses token mapping to pass security attributes. Can be used for computers running Trusted Solaris or other TSIX-cognizant operating environments. See the notes for the Trusted Solaris host type in the first line of this column.

Computer Accreditation Range

The Minimum Label and the Maximum Label are used in the following ways:

- To set the range of labels that can be used when communicating with a computer.
 - In order for a packet to be sent to a computer, the label of the packet must be within the label range assigned to the destination computer in its template.
- To set a label range for packets being forwarded through an unlabeled or RIPS0 gateway.

The label range can be specified in the template for an unlabeled or RIPS0 host type to make it possible to forward a packet to that computer for forwarding, even when the packet's label is not the same as the Default Label.

DOI

A default Domain of Interpretation is assigned in the default templates for all host types. Two computers need to have the same DOI in order to communicate. Organizations with the same DOI need to agree among themselves about how labels and other security attributes are to be interpreted. Each host type has a DOI

associated with it. By default each existing or new template has the default DOI specified in the DOI field. You do need to change the default DOI unless you have reasons for wanting to do so.

As mentioned under “Host Types” on page 181, either the Trusted Solaris or TSIX host type can be specified in templates assigned to Trusted Solaris computers, and if the following things are true the Trusted Solaris or TSIX host-type computers can share the same DOI even if they do not have the same label encodings:

- If only the labels’ names differ on two computers while the labels’ binary formats are the same, if the Trusted Solaris host type is assigned then both computers can share the same DOI.
- If the labels’ names are the same but the labels’ binary formats are different on both Trusted Solaris computers, if the TSIX host type is assigned then both computers can share the same DOI.

DOIs in Trusted Solaris IPv4 Packets

In Trusted Solaris IPv4 packets, the DOI is carried in the packet along with the label. In an IPv4 packet, the specified DOI is included both with the IP options (if any are specified) and in the SAMP header.

Headers (Options [IP options including DOI])	SAMP including DOI	Data
--	--------------------	------

DOIs in Trusted Solaris IPv6 Packets

In Trusted Solaris IPv6 packets, label information is carried in multilevel security (MLS) options portion of the packet’s Headers. Because label information is in the Headers portion of the packet, the packet’s label can be used for routing.

Note - With IPv6, trusted routing using IP labels is not supported.

Headers (Options [SAMP MLS options including DOI]	Data
--	------

If you want to specify another DOI than the default, go to the Advanced Security Attributes tabs.

Default Label

Each unlabeled or RIPS0-type computer is assigned a single label in the `Default Label` field. The `Default Label` assigned to an unlabeled or RIPS0-type computer should reflect the level of trust that is appropriate for the computer and its users. For RIPS0 hosts, the `Default Label` should be the same as the RIPS0 Label (which is a combination the RIPS0 Send Class and the RIPS0 Send PAF).

Default Clearance

Each single-label computer (with the Unlabeled or RIPS0 host type) is assigned a clearance in the `Default Clearance` field. The clearance sets the upper limit for write operations performed on the Trusted Solaris computer by someone on the unlabeled host. For example, on an unlabeled computer with a `Default Label` of `CONFIDENTIAL` and `Default Clearance` of `SECRET`, a user who is working on a file system mounted from a Trusted Solaris host can open an upgraded file with a label of `SECRET` and write into it (if the file's name is known to that user).

Forced Privileges

One or more privileges can be specified in the `Forced Privileges` field of a template that has the unlabeled host type. An unlabeled computer does not understand about privileges. Specifying privileges in this field affects only how the Trusted Solaris computer handles requests from a program that is running on the unlabeled computer. Privileges can be specified to allow a client from an unlabeled computer to do something not otherwise permitted, such as reading a file whose label dominates that of the client or communicating with X clients owned by another user.

Allowed Privileges

Remote Trusted Solaris computers can usually be trusted to provide correct privileges. If needed, the privileges that a remote Trusted Solaris computer is allowed to use can be controlled by specifying a restricted set of privileges in the `Allowed Privileges` field of a template with the Trusted Solaris host type.

Processes running on a remote Trusted Solaris system communicate their effective privileges as part of their security attributes. You can locally restrict those privileges to the ones that are specified in the Allowed Privilege set.

Advanced Security Attributes

The `Advanced Security Attributes` tab in the `Security Families Template` dialog is for setting the following options.

- `DOI`

Every type of supported protocol has a domain of interpretation field. The DOI identifies the labeling scheme. Computers need to have the same DOI in order to communicate. Two organizations that use the same DOI need to agree among themselves to interpret label information the same way.

You need to replace the default domain of interpretation (DOI) only if your site needs another number than the default that is assigned to each host type. Replace the DOI, if desired, by entering an integer into the DOI field.

The type of DOI (TSOL, TSIX, or CIPSO) is determined from the type of host and from any IP label specified in a machine's template. For example, on a Trusted Solaris router with an IP label of CIPSO, the DOI is understood to be a CIPSO DOI.

- `IP Options`

If using trusted routing with IPv4 packets, choose either "none," "CIPSO," or "RIPSO" from the IP Label pull-down menu.

When the CIPSO IP label is specified in a host's template, then a CIPSO label is inserted into the IP options portion of any packet outgoing to that host. See "CIPSO Labels in Packets" on page 190 for how CIPSO labels are used.

If you choose RIPSO, you need to choose a RIPSO Send Class, an optional RIPSO Send PAF, and RIPSO Return PAF from the pull-down menus. PAF means Protection Authority Flag. Any Send PAF specified is used like a compartment name along with the classification to make up the RIPSO label (as in `Top_Secret SCI`). The PAF specified in the Return PAF is used in labeling ICMP messages that can be generated as errors in response to incoming RIPSO labeled packets. The Send Class is also sent back with the RIPSO error in an ICMP message. The RIPSO label should have the same name as the Default Label assigned to the host. Make sure to specify the same RIPSO label and RIPSO PAFs for the sending host, all gateways, and the destination host. See "RIPSO Labels in Packets" on page 190 for how RIPSO labels are used.

Using IP Labels in Trusted Routing

If a computer has an IP Label type of RIPS0 or CIPSO specified in its template, the specified type of IP label is put into outgoing packets, and the incoming packets from the specified host must contain an IP label of the specified type. IP labels can be used for trusted routing. Packets with an IP label are only forwarded to routers whose label range allows the specified IP label.

Some organizations have the requirement to label all of their packets with RIPS0 or CIPSO labels, unless the packets are being sent to unlabeled computers directly connected to the network. Others need to use IP labels for trusted routing of packets going to certain destination hosts. In a homogeneous Trusted Solaris security domain, this is accomplished by assigning a template with the Trusted Solaris host type and an IP label of either RIPS0 or CIPSO to all or some Trusted Solaris computers.

Similarly a template with the TSIX host type can also be configured with CIPSO or RIPS0 labels to achieve the same labeling of packets for TSIX hosts.

And, of course, packets to and from a host assigned a template with a CIPSO or RIPS0 host type carry either a CIPSO or RIPS0 IP label. The IP Options supported in the templates for the Unlabeled host type provide a way to label packets coming into a Trusted Solaris security domain from unlabeled computers. Unlabeled packets become labeled when they pass through Trusted Solaris/ripso or Trusted Solaris/cipso gateways on their way to other Trusted Solaris/ripso or Trusted Solaris/cipso computers. The RIPS0 or CIPSO labels are stripped from packets before they are delivered to unlabeled computers, which are typically outside the security domain. To accomplish this, administrators can specify an IP label of RIPS0 or CIPSO in the template for an unlabeled host.

Default Templates

Trusted Solaris ships with a set of templates. Icons for all defined templates appear when the Security Families tool is double-clicked. The Security Families tool enforces the required fields in the templates, based on the host type you select.

- All default templates should be assessed for their applicability and can be used as is or copied, renamed, or modified by the Security Administrator role.
- New templates can be added.

The simplest and safest configuration is to enable communication only among Trusted Solaris computers that share the same `label_encodings(4)` file. To set up such a configuration, the System Administrator role can assign the default `tsol` template or

other similar template with the Trusted Solaris host type to all Trusted Solaris computers. No modifications are needed.

Default Templates for Trusted Solaris Computers

A computer running Trusted Solaris can be assigned any template that has the Trusted Solaris host type. See the online help for a description of the default templates for the Trusted Solaris host type.

Default Templates for Unlabeled or RIPSOComputers

Trusted Solaris supports communications with computers running operating environments that do not recognize labels (such as the Solaris operating environment). A computer that does not recognize labels or that uses RIPSOC labels must be assigned a single label and a clearance that limit communications with that computer. Before assigning a template that has the Unlabeled or RIPSOC host type to an unlabeled host, specify the following:

- An appropriate label in the Default Label field.
- An appropriate clearance in the Default Clearance field.
- The Maximum Label equal to the Default Label, unless the unlabeled host is a gateway that needs to forward packets at labels that are not equal to its default label.



Warning - When creating or modifying a template for an unlabeled or RIPSOC-type computer, do not forget to change the default label to reflect the level of trust that you have determined is appropriate for the unlabeled or RIPSOC-type computer and its users. Customers who report problems with not being able to communicate with remote single-label computers at the expected label have almost always forgotten to specify that label in the Default Label field.

The default unlabeled and ripso host type templates are valid only when either the default `label_encodings(4)` file is used or another `label_encodings` with the same label names and binary representations for the labels. See the online help in the Security Families tool for descriptions of the default unlabeled or RIPSOC templates.

Do not use the `admin_low` template during normal system operations. The `admin_low` template is included because it is needed at the initial boot time before the system is configured. The template assignment is stored in local `tnrhdb/tnrhtp(4)` files in `/etc/security/tsol` because the name server is not yet available. Once the system is installed, the Security Administrator role should either remove the `0.0.0.` entry entirely or change it to assign a template with an appropriate host type and security attributes.

Wildcard Entry and Prefix Length

A wildcard IP address is the IP address of a subnetwork. A subnetwork is defined by its IP address and its netmask. The netmask determines the prefix that has to be common to all the addresses belonging to a subnetwork.

For example, the IP address 129.150.123.0 is a wildcard with a netmask = 255.255.255.0. The subnet is made up of all the IP addresses between 129.150.123.1 and 129.150.123.255. A optional Prefix Length can be specified in the form of an integer. The prefix length determines the size of the subnet and is the number of 1 bits in the netmask.

TABLE 7-2 Wildcard Address, Netmask, and Prefix Length

class A addresses:a.0.0.0, or a	class B addresses: a.b.0.0, or a.b	class C addresses: a.b.c.0, or a.b.c
netmask = 255.0.0.0	netmask = 255.255.0.0	netmask = 255.255.255.0
prefix length = 8	prefix length = 16	prefix length = 24

With variable-length subnetting, the prefix length does not have to be a multiple of 8. For example, you can have the IP address 129.150.123.224, with a netmask = 255.255.255.224, and a prefix length = 27, covering the addresses between 129.150.123.225 and 129.150.123.255. IPv4 network addresses can have a prefix length between 1 and 32. IPv6 network addresses can have a prefix length between 1 and 128.

The trusted network software looks first for an entry that specifically assigns the host to a template, and if it does not find a specific entry, the software looks for the subnetwork entry that best matches the hosts's IP address (a subnetwork with the longest prefix length to which that address belongs).

If a computer's IP address cannot be matched to an entry, communication with that computer is not permitted.

A default 0.0.0.0 entry matches all computers that are not otherwise matched by other entries.

Sites that need to strictly control remote access should remove the 0.0.0.0 entry using the Remote Hosts tool and to carefully assess whether to use any wildcard addresses. (For more information, see the `tnrhdb(4)` man page.)

CIPSO Labels in Packets

The CIPSO label is derived from the actual label of the data on the sending Trusted Solaris computer.

The trusted networking software puts a CIPSO label and a DOI (domain of interpretation) number into the IP option for outgoing packets and also looks for a CIPSO label and DOI in the IP option of incoming packets, if the trusted network template entry assigned to the remote host meets one of these criteria:

- Assigns the host the `CIPSO` host type
- Assigns the host the `Trusted Solaris` host type, setting the IP label type to `CIPSO`
- Assigns the host the `TSIX` host type, setting the IP label type to `CIPSO`

The CIPSO label that is inserted into outgoing packets is derived by the trusted networking software from the actual label associated with the data. Sometimes Trusted Solaris labels match directly to a CIPSO label. For example, the label of `CONFIDENTIAL` matches the CIPSO label of `CONFIDENTIAL`. However, most Trusted Solaris labels do not map directly to CIPSO labels.

Note - At a site that plans to use CIPSO labels for trusted routing or wishes to communicate with a host with a host type of `CIPSO`, the Security Administrator role should plan ahead to configure the site's labels so they map well to CIPSO labels.

A DOI (domain of interpretation) must also be specified, and the same DOI must be:

- Assigned to the sending host
- In a routing table entry for all gateways through which messages travel and understood by routers
- Assigned to the destination host

Ensuring Labels Are Mappable to CIPSO Labels

The Security Administrator role needs to plan ahead to ensure that the labels defined in the `label_encodings(4)` file map well to CIPSO labels. See *Trusted Solaris Label Administration*.

RIPSO Labels in Packets

The RIPSO, Revised IP Security Option, protocol is described in the IETF RFC 1108. The trusted networking software puts a RIPSO label into the IP option for outgoing packets and also looks for a RIPSO label in the IP option of incoming packets from a host, if the trusted network template entry for the host meets one of these criteria:

- Assigns the host the `ripso` host Type

- Assigns the host the `sun_tsol` host type, specifying the IP Label Type as RIPS0
- Assigns the host the `tsix` host Type, specifying the IP Label Type as RIPS0

RIPS0 labels on outgoing packets are administratively defined. The Security Administrator role specifies them in the `tnrhtp` database, putting the classification in the `RIPS0 Send Class` field and the compartment(s), or protection authority flags (PAF) in the `RIPS0 Send PAF` field.

The following table shows the supported RIPS0 Send classifications.

TABLE 7-3 Supported Classifications for RIPS0 Labels

Supported Classifications for RIPS0 Labels
<code>Top_Secret</code>
<code>Secret</code>
<code>Confidential</code>
<code>Unclassified</code>

The `RIPS0 Send PAF` and `Return PAF` fields refer to Protection Authority Flags, which are shown in Table 7-4. PAFs specified in the `Send PAF` field are used like compartment names along with the classification within the RIPS0 labels (as in `Top_Secret SCI`). PAFs specified in the `Return PAF` field are used in labeling ICMP messages that can be generated as errors in response to incoming RIPS0 labeled packets. The classification sent back in an ICMP message is the same as the RIPS0 classification in the packet.

TABLE 7-4 Protection Authority Flags that Can Be Specified in the RIPS0 Send PAF or RIPS0 Return PAF Fields

Protection Authority Flags (may be specified along with supported classifications in RIPS0 labels or specified as RIPS0 errors)
<code>GENSER</code>
<code>SIOP-ESI</code>
<code>SCI</code>
<code>NSA</code>
<code>DOE</code>

Understanding Security Attributes Assigned to Network Interfaces

All interfaces on a computer running Trusted Solaris are automatically detected by the trusted network software and assigned a default set of attributes. The Interface Manager shown below is used only when the Security Administrator role wants to change the defaults for an interface.

The default attributes are shown in the following figure:

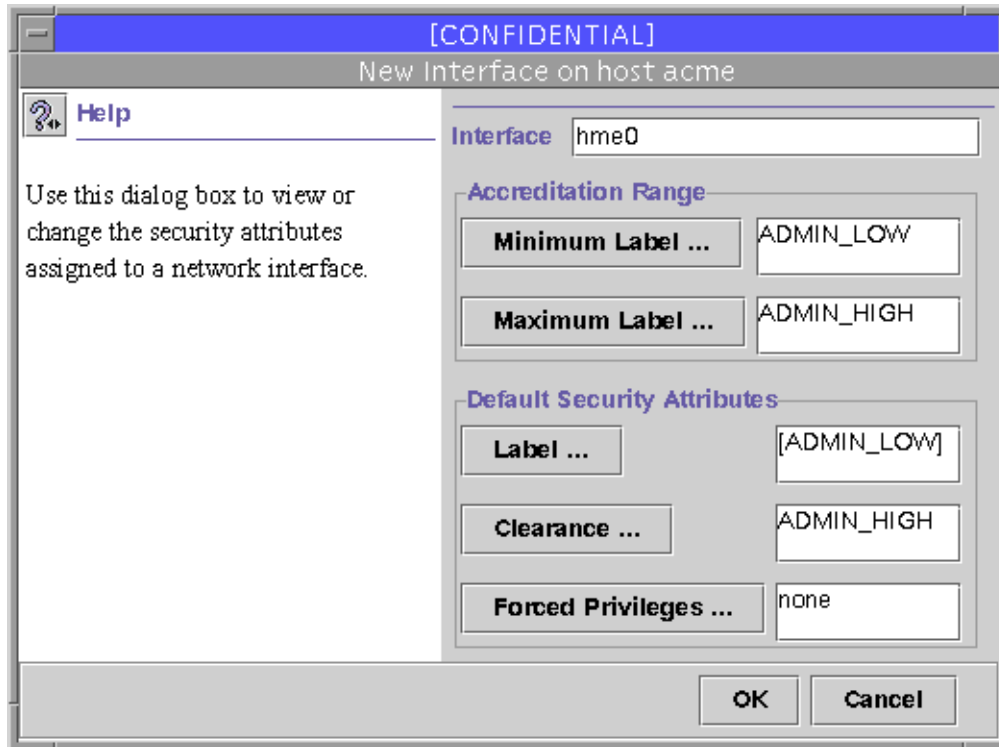


Figure 7-2 Interface Manager with Default Security Attributes

Summary: any values specified for a computer in a template take precedence over any values supplied for the network interface, and if no values are specified, system defaults apply. For an example of precedence, say computer A is assigned a default label of INTERNAL, while the network interface that is connected to the network where computer A resides is assigned a default label of PUBLIC. In that case, data coming from computer A is assigned the INTERNAL label, and the default label assigned by the network interface is not used.

Network Interface Accreditation Range

The `Minimum Label` and the `Maximum Label` are used to set the range of labels for data that can be sent through the interface.

Note - Full communications within a Trusted Solaris domain require an accreditation range of `ADMIN_LOW` to `ADMIN_HIGH`.

To be able to leave certain fields empty in a single template assigned to one computer or to a group of computers that is accessed through the same network interface, the Security Administrator role can specify the values in an entry that applies to that network interface.

The entries assigned to network interfaces are looked at only if certain fields are left empty in the template assigned to a computer. If a value is not found either in the template that covers the host or in an entry that applies to the interface through which the remote computer is accessed, then a set of default values is applied.

Note - Restrict the accreditation range on a network interface with care. Network services fail unless the network interface is configured with an accreditation range that includes the labels upon which those services depend. For example, audit clients cannot write `ADMIN_HIGH` audit data onto the audit server unless the `ADMIN_HIGH` label is in the range. Full communications within a Trusted Solaris domain require an accreditation range of `ADMIN_LOW` to `ADMIN_HIGH`.

Default Security Attributes

The `Default Label`, `Default Clearance`, and optional `Forced Privileges` in the Interface Manager are rarely useful. They would be used when the Trusted Solaris computer is communicating with a computer that is running an operating system that does not recognize labels or privileges, such as the Solaris operating environment, and then only if the same fields have been left empty in the template that applies to the single-label computer. For example, the Security Administrator role might create an entry for a second interface on the local computer that would apply the same label, clearance, and optional forced privileges to all computers running Solaris on the network that is connected to the second interface. These fields could then be left empty in any templates that cover the computers (as specified in the Security Families tool in Computers and Networks).

Default Label

The `Default Label` should reflect the level of trust that is appropriate for the computer and its users.

Default Clearance

The `Default Clearance` sets the upper limit for write operations performed on the Trusted Solaris computer by someone on the unlabeled computer. For example, on an unlabeled computer with a Default Label of `CONFIDENTIAL` and Default Clearance of `SECRET`, a user who is working on a file system mounted from a Trusted Solaris computer can open an upgraded file with a label of `SECRET` and write into it (if the file's name is known to that user).

Forced Privileges

An unlabeled computer does not understand about privileges. Specifying privileges in the `Forced Privileges` field affects only how the Trusted Solaris computer handles requests from a program that is running on the unlabeled computer. Privileges can be specified to allow a client from an unlabeled computer to do something not otherwise permitted, such as reading a file whose label dominates that of the client or communicating with X clients owned by another user. Precedence If the corresponding values are set in a template that covers the computer, the value in the template takes precedence over the values specified for the network interface.

- Is the needed value specified in the template?
 - If yes, the value in the template is used
 - If no, is the needed value specified in an entry for the interface
 - If yes, use the value specified for the interface.
 - If no, use the default value.

Accreditation Checks

The trusted networking software performs accreditation checks to compare the security attributes of the source host, the destination host, and of the routes along the way.

Security attributes for the accreditation range check (accreditation range and any CIPSO or RIPS0 label information that may be specified) are obtained from a host's templates. The security attributes for a route (its SRI) are obtained from the route's emetric in the routing table. If an emetric for a route has not been specified, the security attributes of the first hop gateway host's entries are checked.

On a router, accreditation checks are performed only if the packet to be forwarded has RIPS0 or CIPSO labels and then the labels in the IP options portion of the packet are used. If the packet has a CIPSO label, its label is compared to the label range of the incoming and outgoing interface. Its label is also compared to the label range of the next hop gateway.

MAC Enforcement on Outgoing Messages

The following accreditation checks are performed on the sending host.

- The label of the packet being sent must be:
 - Within the accreditation range of the destination host
 - Within the accreditation range of the network interface of the source host.
- If the packet has a CIPSO label, then its DOI must match the DOI of the destination and of the route's emetric. If no emetric is specified for the route, the DOI must match the DOI of the first hop gateway.
- If the packet has a RIPS0 label, then its RIPS0 label and PAF flag must match the RIPS0 label and PAF flag of the destination and of the route's emetric. If no emetric is specified for the route, the RIPS0 label and PAF flag must match the RIPS0 label and PAF flag of the first hop gateway.
- If the destination is specified as a MSIX host, then the label of the packet being sent must be within the accreditation range of the destination host and the route's emetric must include the MSIX attribute. If no emetric is specified for the route, the host type of the first hop gateway must be specified as MSIX and the label of the packet must be within the accreditation range specified for the first hop gateway.

Note - A first hop check occurs when a message is being sent from a host on one network to a host on another through a gateway.

MAC Checks on Messages Being Forwarded

On a Trusted Solaris gateway, accreditation checks are performed for the next hop and for the network interfaces.

If the packet has CIPSO label information, the following must be true for a packet to be forwarded:

- The route's emetric must include the CIPSO option. If no emetric is specified for the route, the next hop gateway's entry must be defined as either of the following:
 - CIPSO host type
 - sun_tsol host type with a CIPSO IP label
 - tsix host type with a CIPSO IP label
- The CIPSO label of the packet must be within the accreditation range from the emetric of the route. If no emetric is specified for the route, the packet's CIPSO label must be within the accreditation range specified in next hop gateway's entry
- The CIPSO DOI specified in the network database entry for the outgoing interface must equal the packet's DOI.

If the packet has RIPS0 label information, the following must be true for a packet to be forwarded:

- The route's emetric must include the RIPS0 option. If no emetric is specified for the route, the next hop gateway's entry must be defined as either of the following:
 - RIPS0 host type
 - tsol host type with a RIPS0 IP label
 - tsix host type with a RIPS0 IP label
- The RIPS0 label of the packet and PAF must be the same as the RIPS0 label and RIPS0 PAF in the emetric of the route, or if no emetric is specified for the route, the packet's RIPS0 label and RIPS0 PAF must be the same as the RIPS0 label and RIPS0 PAF specified in next hop gateway's entry

If the label of a message is not within the minimum and maximum labels specified in the accreditation range for any of the destination host, gateways, or the network interface, the message is dropped.

MAC Enforcement on Incoming Messages

The following checks are performed on a receiving host.

- The label of the packet being received must be:
 - Within the accreditation range specified in the source host's trusted network database entry
 - Within the accreditation range specified in the trusted network database entry for the network interface receiving the data
- If the packet has a CIPS0 label, then its DOI must match the DOI specified in the receiving host's trusted network database entry
- If the packet has a RIPS0 label, then its RIPS0 label and PAF flag must match the RIPS0 label and PAF flag specified in the trusted network database entry for the receiving host

For incoming communications, the Trusted Solaris networking software obtains labels and other security attributes from the packets themselves whenever possible—which is only completely possible when the messages are sent from systems that support labels and all the other required attributes in a form recognized by the Trusted Solaris system. In many cases, packets arrive from hosts that are not label-cognizant or that do not send recognizable labels, or the packets do not have all of the other required attributes in their packets.

When the needed security attributes are not all available from a packet, those that are lacking are assigned to the message from trusted networking databases. Any attributes not obtainable from the host's entry are supplemented by the attributes

specified in the entry in the trusted network interface database entry the interface through which the message arrives.

Administering Routing

Some sites may restrict communications outside of the local network to a single label, such as PUBLIC. At these locations, assigning a network accreditation range of PUBLIC to the interface that is connected to the external network would provide the desired degree of control. The Trusted Solaris environment supports additional methods for routing communications between networks, so that the Security Administrator role can set up routes that enforce the degree of security required by the site's security policy. See the *TCP/IP and Data Communications Administration Guide* for more details about TCP/IP and routing.

Background

For communications sent to destinations on the same subnet, accreditation checks are performed by Trusted Solaris endpoints only since no routers are involved. (Because gateways and routers route packets, the terms gateway and router are used interchangeably in this discussion.) Accreditation range checks are performed at the source. If the receiving host is running Trusted Solaris, accreditation range checks are also performed at the destination.

When the source and destination hosts are on two different sub-networks, the packet is sent from the source host to a gateway. The accreditation range of the destination and of the first hop gateway is checked at the source when selecting a route. The gateway forwards the packet to the network where the destination host is connected. A packet may go through a number of gateways before reaching the destination.

On Trusted Solaris gateways, accreditation range checks are performed in certain cases. A Trusted Solaris computer routing a packet between two unlabeled hosts compares any IP label in the IP options portion of the packet against the accreditation range of the network interface. If no IP option is specified, the default label assigned to the sending host is compared to the accreditation range on the network interface through which the packet is going. Because the "write up read-down" MAC rule is enforced even on communications between unlabeled hosts, the default label of the sending host must be dominated by the default label of the destination host. In practice, two way communications would be impossible unless both unlabeled hosts shared a default label.

Each gateway maintains a list of routes to all destinations. Standard Solaris routing metrics allow routes to be chosen based on the shortest path to the destination. Extensions in Trusted Solaris 2.5.1 and later compatible releases enable trusted

routing based on the shortest path to the destination that also satisfies security requirements. IP security options in a packet allow IP labels to be available for accreditation range checks on intermediate routers.

Trusted routing depends on all gateways recognizing extended RIP, the Routing Information Protocol. Therefore, trusted routing is only possible in an Intranet whose gateways are all known to use RIP, because routing in the Internet is done using other protocols.

Some sites using trusted routing need to enable communications with Trusted Solaris hosts that are on the other side of a cloud of unlabeled hosts when communications must go through one or more routers that do not understand labels. At these sites, the Security Administrator role needs to set up tunneling. (The terms *clusters* and *tunneling* are defined under “Setting Up Tunneling” on page 207.)

Choosing Routers

Because routes must be carefully chosen in the Trusted Solaris environment, the Security Administrator role needs to understand the security characteristics of all routers through which sensitive information is passing.

For the highest degree of trust, routes should be set up with Trusted Solaris computers as routers. If other types of routers are used, keep in mind that the Trusted Solaris security features are not always available on those routers, and without administrative action packets can be routed through routers without MAC security protection.

CIPSO and RIPS0 routers drop packets when they do not find the right type of information in the IP options section of the packet. For example, a CIPSO router drops a packet if it does not find a matching CIPSO label or a matching DOI in the packet's IP options section. Other types of routers not running Trusted Solaris do not drop packets when they find labels they do not understand in the IP options section; they just pass the packets along. Be aware of these considerations when setting up communications between hosts, and make sure that packets are routed through the appropriate types of routers.

To support trusted routing, the Trusted Solaris routing tables are extended to include security information along with the metric for the number of hops to the destination. (See Specifying the SRI and Emetrics, which follow this section.)

Specifying the SRI

The set of security attributes necessary for trusted routing is called the *SRI* (for *security routing information*). The SRI always includes both of the following to establish the route's accreditation range:

- Minimum label

- Maximum label

As described on the `route(1M)` man page, the SRI can also incorporate other security attributes. The SRI is obtained from one of two possible sources:

- The dynamic routing software initially derives the SRI from the `tnrhdb/tnrhtp` entries for the gateway on the router.
- The Security Administrator role can enter the SRI manually in a static routing table.

Emetric

The *emetric* (Extended Metric) consists of both the standard routing metric and the SRI. The emetric is stored in each route's entry in the routing table. The routing software selects the shortest path that satisfies the security requirements by comparing emetrics. Alternately, the emetric can be entered manually for static routes using the `route(1M)`. (See "Routing Table" on page 199 for how routes are manually defined.)

If dynamic routing is used, the routing daemon, `in.routed` broadcasts a special type of security-enhanced response packet advertising the known routes.

Several routes through multiple gateways may exist between a sending and receiving host, and the emetric for each route may be different.

Routing Table

The routing table in the kernel of each host contains routes. Each entry in the routing table provides a route to a particular destination:

destination (a specific host or network)	first hop gateway (first gateway in the route)	interface associated with gateway
--	--	-----------------------------------

The routing software tries to find a route to the destination host in the route tables. When the host is not explicitly named, the routing software looks for an entry for the (sub)network where the host resides. When neither the host nor the network where the host resides is defined, the host sends the packet to a default gateway, if one has been defined. Multiple default gateways can be defined, and each is treated equally. A pointer keeps track of which default gateway has been used most recently, and the next one in the list is used for the next routing.

Routing table entries are created either of the following two ways:

- Dynamically
 - The `routed(1M)` routing daemon dynamically creates the route entries including the emetric
- Statically

The administrator role creates static routes manually in one of two routing files.
The administrator role chooses whether to supply an emetric with the route entry.

With a small network, it is feasible for the administrator role to manually set up routes, and to manually make changes to the routing table when conditions change. For example, many sites have a single gateway through which all communications go to the outside world. In these cases, the single gateway can be statically defined as the *default* on each host on the network.

Note - Routers using static routing do not advertise their availability, and so they are invisible to almost all other systems, which almost always use dynamic routing.

With large networks, manually configuring and maintaining static routes is impossible. In large networks, dynamic routing is almost always used.

Extended RIP

Xerox Routing Information Protocol (RIP) version is extended in the Trusted Solaris environment to supply security attributes along with a route's metric when the router advertises the route. The extended RIP is compatible only within an Intranet whose gateways all recognize RIP, because routing in the Internet is done using other protocols.

Determining Dynamic or Static Routing

The following figure shows how the presence or absence of certain files and programs on a Trusted Solaris host that is not a gateway determines whether static or dynamic routing is done.

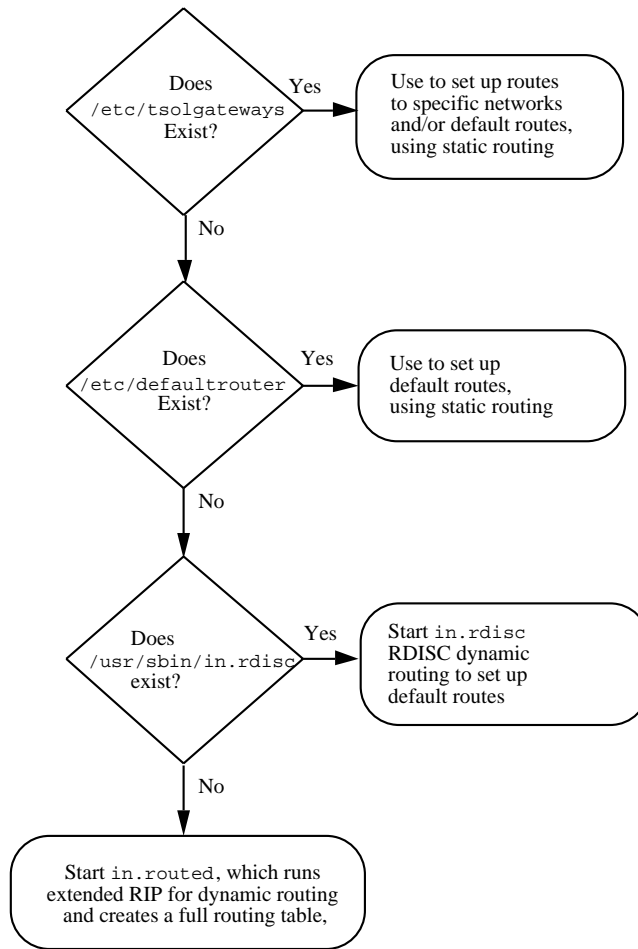


Figure 7-3 How a Host Determines Which Type of Routing to Do

Setting Up Trusted Routing

This section describes what you need to know to ensure routing of outgoing communications so that they go only through gateways that are configured with a security level matching the sensitivity of the data being sent out. This section assumes you have read and understood “Administering Routing” on page 197.

Trusted routing requires that the Security Administrator role is able to configure or to coordinate configuration of trusted network database entries on all hosts and gateways.

On a sending host, when a packet is outgoing to a destination whose template defines a CIPSO IP label and CIPSO DOI, the DOI assigned to the destination must match the DOI assigned to the sending host. The template for the first hop gateway must define a CIPSO DOI that matches that of the destination and must have an accreditation range that includes the CIPSO label of the packet. When a packet is outgoing to a destination whose template defines a RIPS0 IP label (classification) and RIPS0 PAF (compartments), the combined RIPS0 classification and compartments assigned to the destination must match the RIPS0 classification and compartments assigned to the sending host. The template for the first hop gateway must define a RIPS0 label and PAF that match the destination. When an IP label is specified by the Security Administrator role in a destination host's template, the trusted networking software adds whichever type of label was specified to the IP portion of packets that go out. If the IP label type is CIPSO, the trusted network software derives the CIPSO label from the label of the packet. When the IP label type is RIPS0, the trusted network software uses the RIPS0 label administratively defined in the destination host's template.

On a receiving host, when a packet is incoming, the CIPSO label and CIPSO DOI specified in the template for the source host must match the CIPSO label and DOI specified for the receiving host.

Allowing a Single-label Gateway to Forward Packets at Multiple Labels

A single-label host (specified with a host type of `unlabeled` or `rips0`) must be assigned a default label in its template. A minimum and a maximum label in the unlabeled host's template define an accreditation range that can be used for routing. Specifying the accreditation range enables a single-label gateway to be able to forward packets that it would not otherwise be allowed to receive based on its default label alone.

The accreditation range specified for a single-label gateway is used by the trusted network software in deciding which packets can be sent through that gateway. The packet being forwarded by the unlabeled gateway must be within the gateway's accreditation range.

Specifying Security Attributes for Remote Computers and Setting Up Routing

This chapter provides procedures for the Security Administrator role to follow when specifying which security attributes apply to communications with remote computers.

- “Assigning Security Attributes to Remote Computers and Network Interfaces” on page 204
- “Before Assigning Templates to Computers” on page 205
- “Making Decisions About Templates” on page 205
- “What You Need to Know About Trusted Network Databases” on page 206
- “Modifying the Boot-time Trusted Network Databases” on page 206
- “Setting Up Tunneling” on page 207

This chapter includes the following procedures.

- “To Use the Security Families Tool” on page 208
- “To Specify Templates for Security Families” on page 208
- “To Assign Templates to Computers” on page 209
- “To Create a Wildcard Entry for All Computers Not Otherwise Specified” on page 211
- “To Create a Wildcard Entry for All Computers Not Otherwise Specified” on page 211
- “To Configure a Network Interface” on page 212
- “To Set Up Static Routes with Optional Emetrics for Specific Hosts or Networks ” on page 213

- “To Set Up Tunneling” on page 215

Assigning Security Attributes to Remote Computers and Network Interfaces

Each site's Security Administrator decides which computers should be allowed to communicate with the Trusted Solaris system and which security attributes each computer needs to have assigned. Security attributes are assigned to computers by means of templates.

Templates can be assigned directly to a computer or indirectly through a wildcard entry that assigns a template to a network address that includes the computer. If a computer does not have a template assigned either directly or indirectly, no communications are allowed with that computer.

Computers (hosts or routers) that share the same template are considered to be part of the same security family. The SMC Security Families tool is used to:

- Assign a template
- Change the assignment of a template
- Modify a template
- Create a new template
- Delete a template

Optionally, the SMC Interface Manager tool can be used to assign security attributes to network interfaces, but doing so is useful only in limited circumstances when the defaults are not acceptable:

- To limit the range of labels at which communications are allowed through a network interface, the Security Administrator role can set a restricted label range. The default label range is ADMIN_LOW to ADMIN_HIGH.
- If it is desirable to be able to leave certain fields empty in a single template assigned to one computer or to a group of computers that is accessed through the same network interface, he or she can specify the values in an entry that applies to that network interface.

The entries assigned to network interfaces are looked at only if certain fields are left empty in the template assigned to a computer. If a value is not found either in the template that covers the host or in an entry that applies to the interface through which the remote computer is accessed, then a set of default values is applied.

See precedence and Using the Interface Manager for more details.

Before assigning templates, the Security Administrator role should do the following:

- Review the existing templates.
 - Choose View->Details from the Security Families tool, which displays some of the values specified for each template.
 - Use the Security Families tool to bring up the Template Manager dialog, select each template in turn and view its contents.
- Decide which templates should be used for each host and network
- Modify existing templates or create any new templates needed for the site

Setting Up Templates

Follow the online help in the Solaris Management Console for how to use the SMC tool and enter values in the fields.

Before Assigning Templates to Computers

Have the following information available:

- A list of the available templates
- A list of all the computers and networks with which the computers in the Trusted Solaris system are allowed to communicate.

Making Decisions About Templates

Make the following decisions before starting because they affect how you configure templates and assign the templates to computers.

- Decide which security attributes to apply to each computer.
- Decide whether you can use existing templates or to modify them.

What You Need to Know About Trusted Network Databases

The Security Families tool stores template definitions in the `tnrhtp(4)` database and stores template to host assignments in the `tnrhdb(4)` database. The Interface Manager stores network interface definitions in the `tnidb(4)` file.

The Trusted Solaris version of the name service switch file, `nsswitch.conf(4)`, includes entries for `tnrhtp` and `tnrhdb`, which should be modified to suit each site's configuration. The default is shown below.

```
# TSOL
tnrhtp: files nisplus
tnrhdb: files nisplus
```

To modify these entries, the System Administrator role uses the Name Service Switch action. See “To Launch Administrative Actions from a Local Application Manager” on page 43, if needed, for how to access the Name Service Switch action. To preserve the required file attributes (owner, group, mode and label), the role should not edit the `nsswitch.conf` file directly.

Modifying the Boot-time Trusted Network Databases

Boot-time-only local versions of the `tnidb(4)` and `tnrhdb(4)` files reside by default in the `/etc/security/tsol` directory on every Trusted Solaris computer. The `admin_low` template from the local `tnrhtp` file is assigned to the wildcard address `0.0.0.0` in the local `tnrhdb` file. The files are local because the entry is needed at the initial boot time before the system is configured and before the naming server is available.



Caution - The `admin_low` template should not be used at any other time. Once the system is installed, the Security Administrator role should either remove the `0.0.0.0` entry or assign another template to it that has the appropriate host type and security attributes.

After first boot of each NIS+ client, we recommend that the install team does one of the following, depending on the level of security required:

- Remove the wildcard entry and create entries for all the hosts to be contacted during boot, adding templates as needed

See “To Change the Default Entry in the Boot-Time Files” on page 210.



Caution - Organizations that specifically define each host and network with which communications are allowed need to change the default wildcard entry.

- The following entries are needed.
- An entry for the NIS or NIS+ master
- An entry for every local IP address

For example, if a router called `trusted` has two interfaces with IP addresses: `137.150.113.111` for `trusted` and `137.150.113.112` for `trusted-gw`, you need to make one or more entries that covers both addresses.

- An entry for the local hosts' loopback address:

```
127.0.0.1:tsol
```

- One or more router entries

If the NIS+ client is not a router, do one of the following:

- Enter the IP address of the router on the local network, such as `137.150.113.120`.
- Create a fall back network entry, such as `137.150.113.0`.
- If the NIS+ client is a router, list all the routers with which it needs to communicate during boot.

Setting Up Tunneling

Tunneling enables the sharing of emetrics for routes on an Intranet even when there is a non-Trusted Solaris cloud of hosts and gateways between two Trusted Solaris gateways. All hosts must be in the same Intranet with gateways using Trusted Solaris extended RIP. Without tunneling, the security response packets generated by extended RIP on one gateway cannot be received on the remote Trusted Solaris gateway to pass along the emetrics of its known routes.

To set up tunneling, the Security Administrator role creates a `tunnel` file on a Trusted Solaris gateway. The tunnel file contains the IP addresses of remote networks connected to Trusted Solaris gateways. Unlabeled broadcast packets containing security information are sent directly to the networks listed in the `tunnel` file, where they are picked by Trusted Solaris gateways. See “To Set Up Tunneling” on page 215.

Note - The term tunneling as used here has nothing to do with the IP-in-IP tunneling feature in Solaris.

Procedures

▼ To Use the Security Families Tool

1. **Assume the Security Administrator role, go to an ADMIN_LOW workspace, bring up the SMC, and load a Trusted Solaris toolbox with the desired scope.**
2. **Double-click the Computers and Networks tool.**
3. **Double-click the Security Families tool.**
All currently-defined templates display in the right hand pane.

▼ To Specify Templates for Security Families

1. **Review the security attribute settings in the default templates and decide whether to modify them to suit your site's security policy.**
2. **As Security Administrator, bring up the Security Families tool.**
All currently-defined templates display in the right hand pane.
3. **To modify an existing template, double-click the name of a template, and choose Properties from the Action menu.**
The Modify Template dialog displays with the name of the currently-selected template at its top.
4. **To add a new template, choose Add Template from the Action menu.**
The New Template dialog displays.
5. **Supply the desired values in the tabs in the Template Manager.**
6. **When done, click OK.**

▼ To Assign Templates to Computers

- 1. As Security Administrator, double-click on the Security Families tool.**
All currently-defined templates display in the right hand pane.
- 2. To change the assignment of a computer or network to a template, double-click the name of the ALL template.**
All computers and networks that are currently in the ALL family display in the right hand pane.
- 3. Double-click the icon for the computer or network.**
- 4. Chooses Action->Properties.**
The Modify Remote Host Entry dialog displays with the IP address of the network or computer at its top.
- 5. Supply the desired values in the fields in the Template Manager.**
- 6. Click OK**
- 7. To assign an existing template to a computer or network, double-click the name of a template.**
All computers currently defined in the same Security Family display in the right hand pane.
- 8. Choose Action->Add Host.**
The New Remote Host Entry dialog displays.
- 9. Type in either the Hostname or the IP Address for any computer or network to which the template should be assigned.**
If a Hostname is entered, the IP address is looked up. If an IP Address is entered, then the hostname is looked up. The IP Address field accepts any valid IPv4 or IPv6 address for the computer or network.
- 10. Type in an optional Prefix Length that indicates the length of the network portion of the address.**
- 11. Choose the name of a template from the Template pull-down menu.**
- 12. Click OK.**

▼ To Change the Default Entry in the Boot-Time Files

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Use the Admin Editor action to open the `/etc/security/tsol/tnrhdb` file for editing.**
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
The following example shows the default entry.

```
0.0.0.0:admin_low
```

3. **Remove the wildcard entry or redefine it.**
If you remove the entry, go to the next step. If you redefined the entry, go to Step 5 on page 211.
4. **To replace the wildcard entry, make an entry for every host with which the current host needs to communicate during boot.**
 - a. **For a NIS+ client, make an entry for the NIS+ master.**
 - b. **For a NIS client, make an entry for the NIS master.**
 - c. **Make an entry for every IP address of the host being configured.**
The example shows entries for a host that is a router with two interfaces.

```
127.150.113.111:tsol 127.150.113.112:tsol
```

- d. **Make an entry for the loopback address of the host being configured.**

```
127.0.0.1:tsol
```

- e. **If the host being configured is not a router, make one or more entries so the host can find its router.**
Do one of the following substeps.
 - i. **Make an entry for a router on the local network.**

- ii. **Make a fallback entry for the network so the NIS+ client can find its local router.**

The example below shows entries for a NIS+ client: the host's IP address 129.96.22.29, the NIS+ master's IP address 129.96.22.40, the loopback IP address 127.0.0.1, and the network's IP address 129.96.22.0 (which is used for locating the router).

```
129.96.22.29:tsol
129.96.22.40:tsol
127.0.0.1:tsol
129.96.22.0:tsol
```

5. **Write and quit the file.**

```
:wq
```

6. **Add additional templates that are needed for the entries.**

Use the SMC Security Families tool to create any needed templates.

For example, if one of the hosts identified in the `tnrhdb(4)` has a host type of `unlabeled`, add a template that has the `host_type=unlabeled` as shown below.



Caution - When creating a new template with the unlabeled host type, do not forget to change the Default Label. The Default Label in the default template is `ADMIN_LOW`, and normal users cannot work at that label.

▼ To Create a Wildcard Entry for All Computers Not Otherwise Specified

1. **Assume the Security Administrator role, go to an `ADMIN_LOW` workspace, and access the SMC Security Families tool in the desired scope.**

2. Use the **Security Families Add Template** option to create a new template, if needed.
3. Use the **Security Families Add Host** option to assign the desired template to the wildcard IP address of **0.0.0.0**.
4. When you are done making changes, click the **OK** button.



Caution - Using a wildcard entry allows any host to communicate with the system.

▼ To Configure a Network Interface

1. If adding a new interface, insert the network interface card, following the hardware and software installation steps in the manuals shipped with the interface.

The interface installation program installs a new device file called `hostname.device_abbreviation` in `/etc`.

2. For a host with more than one network interface, do the configuration either for a router or multihomed host, as described in the base Solaris *TCP/IP and Data Communications Administration Guide*.
3. If the site security policy requires other than default settings for any interfaces, change the entries.

As described in “Understanding Security Attributes Assigned to Network Interfaces” on page 192, interfaces on a computer running Trusted Solaris are automatically detected by the trusted network software and assigned a default set of attributes. The Interface Manager shown below is used only when the security administrator role wants to change the defaults for an interface.

The default attributes are shown in the following screen shot.

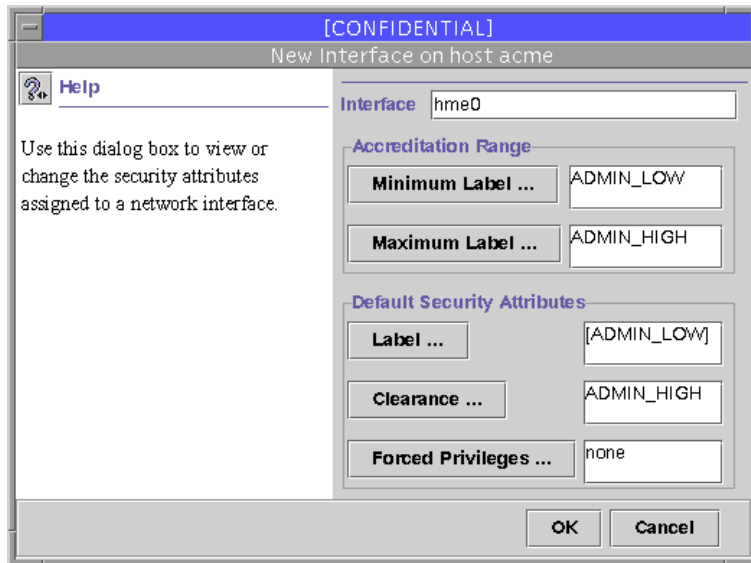


Figure 8-1 Interface Manager with Default Security Attributes

▼ To Set Up Static Routes with Optional Emetrics for Specific Hosts or Networks

1. **Assume the System Administrator role and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Use the Set TSOL Gateways action to open the /etc/tsolgateways file for editing.**
See “To Launch Administrative Actions from a Local Application Manager” on page 43, if needed. See also the `tsolgateways(4)` man page for more about the syntax and use of `/etc/tsolgateways`. The syntax of the emetric in `tsolgateways` is the same as for the `route` command.
3. **Set up one or more default entries, if desired.**
The first entry sets up a default route, using a specific gateway’s address 129.150.113.36 and a metric of 1 to be used when there is no specific route defined for either the host or destination of a packet.

```
default 129.150.113.36 1
```

4. Set up one or more network entries, if desired.

The second line below shows a network entry set up with a standard metric. The third line shows a network entry set up with an emetric, setting a label range of PUBLIC to INTERNAL.

```
default 129.150.113.36 1
net 129.150.102.0 gateway-101 1
net 129.150.101.0 gateway-102 -m metric=2,min_sl='`PUBLIC`', max_sl='`INTERNAL`'
```

5. Set up one or more host entries, if desired.

The new fourth line shows a host entry set up for a gateway host named trusted with an emetric setting a label range of PUBLIC to PUBLIC.

```
default 129.150.113.36 1
net 129.150.102.0 gateway-101 1
net 129.150.101.0 gateway-102 -m metric=2,min_sl="PUBLIC",max_sl="INTERNAL"
host 129.150.101.3 trusted -m metric=2,min_sl="PUBLIC",max_sl="PUBLIC"
```

6. Make sure there is an entry for any destination host(s) and gateway(s) in the local /etc/hosts file, or NIS+ hosts.org_dir table.

```
129.150.113.36 merlot
```

7. Make sure there is an entry for all destination hosts, network(s) and gateway(s) in the local /etc/security/tsol/tnrhdb file.

```
29.150.113.36:tsol1
```

8. Write and quit the file.

```
:wq
```

▼ To Set Up Tunneling

A forwarding host is any Trusted Solaris 8, 7, or 2.5.1 gateway being set up to tunnel through one or more gateway(s) not running Trusted Solaris 8, 7, or 2.5.1, to advertise the emetrics of its routes to Trusted Solaris 8, 7, or 2.5.1 gateways on the other side.

1. **Assume the Security Administrator role on the forwarding host and go to an ADMIN_LOW workspace.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. **Use the Admin Editor action to create or open the `/etc/security/tsol/tunnel` file for editing.**

See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.

3. **Enter one IP address of a target (sub)network on per line.**

See the following example.

```
129.299.36.0
```

4. **Write and quit the file.**

```
:wq
```

5. **To set up two way routing using emetrics, repeat the previous steps on the remote gateway(s), specifying the IP address for the local network.**

Managing Files and File Systems

This chapter gives the background needed to understand how to manage files, directories, and file systems and how to share (export) and mount files in the Trusted Solaris system. This chapter covers the following topics:

- “Security Attributes on Files and Directories” on page 219
- “Specifying Security Attributes on Files and Directories” on page 220
- “Security Attributes on File Systems” on page 222
- “Specifying Security Attributes on Variable File Systems” on page 224
- “Specifying Security Attributes on Fixed File Systems” on page 225
- “Mounting Various Types of File Systems in the Trusted Solaris System” on page 226
- “Trusted Solaris Attribute Precedence Rules” on page 231
- “Trusted Solaris and NFS” on page 232
- “Exporting Directories” on page 233
- “Troubleshooting Mount Failures” on page 233

This chapter provides the following procedures.

- “To Back Up Files” on page 234
- “To Restore Files” on page 234
- “To Change Labels and Privileges on Files and Directories Using the File Manager” on page 234
- “To Specify Alternative Security Attributes While Creating a Local File System” on page 236
- “To Set Security Attributes on a File System ” on page 237
- “To Specify Mount-time Security Attributes on the Command Line” on page 238
- “To Specify Mount-time Security Attributes in the `vfstab_adjunct` File” on page 239

- “To Share a Directory for Mounting by Other Computers ” on page 240
- “To Mount a TMPFS-type File System Using the Command Line ” on page 242
- “To Mount a CD-ROM with a HSF5-type File System” on page 242
- “To Automatically Launch a CD Player for an Audio CD-ROM” on page 242
- “To Listen to an Audio CD as any User or Role” on page 243
- “To Troubleshoot Mount Failures ” on page 243

The Trusted Solaris system supports the same files and directories, most of the file system types, and all of the file system management commands that are supported by the base Solaris system. Whenever a file or directory is accessed, security attributes obtained from various sources are used in making access control decisions. New commands have been added and existing commands have been modified to work with the extended Trusted Solaris security attributes.

This chapter describes what the security administrator role needs to know about:

- How security attributes are obtained
- How to set security attributes and change their settings
- How to specify security attributes at mount time for file systems that do not have them

Additional facts related to administering files, directories, and file systems in the Trusted Solaris environment are below:

- No order requirements are imposed on the labels of directories in a pathname.
- Files and directories can be created only at the same label as the containing directory.
- Privileged subjects can create files and directories and relabel existing files and directories at any valid label to create *upgraded* or *downgraded objects*.

See “To Change Labels and Privileges on Files and Directories Using the File Manager” on page 234.

- The system can be configured so the names of upgraded files and directories are not visible. By default, their names are visible.

To change the default so the names are visible, the security administrator role can change the setting of the `tsol_hide_upgraded_names` switch in the `system` file as described in “Changing Configurable Trusted Solaris Kernel Switches” on page 66 and reboot.

- Directory names are cleared when a directory is removed, to meet the object reuse requirement that the names of removed directories should no longer be accessible.
- Trusted Solaris symbolic links have labels.
- Multilevel directories (MLDs) appear in the file system as ordinary directories with a flag identifying them as MLDs.
- MLDs require no privilege to create, delete, or use.

- Read-down access to SLDs within an MLD permits an unprivileged process to combine information from SLDs at its own and lower labels.
- If an MLD is mounted by a single-label computer, the SLD that corresponds to the label administratively assigned to the computer in the trusted networking databases is mounted instead of the MLD.

If, for example, a user's home directory is automounted on an unlabeled computer, only the SLD with the default label assigned to the computer in the Security Families template is mounted. For example, if the default label for the computer is `INTERNAL_USE_ONLY`, then only the SLD at `INTERNAL_USE_ONLY` is mounted on the unlabeled computer.

Specifying Security Attributes on Files and File Systems

Security attributes can be specified:

- At the level of an the individual file or directory within a file system
See “Security Attributes on Files and Directories” on page 219 and “Specifying Security Attributes on Files and Directories” on page 220.
- At the level of the file system
See “Security Attributes on File Systems” on page 222, “Specifying Security Attributes on Variable File Systems” on page 224, and “Specifying Security Attributes on Fixed File Systems” on page 225)

If a needed attribute is not obtained elsewhere, a set of defaults is used. For rules about how attributes are obtained, see “Trusted Solaris Attribute Precedence Rules” on page 231.

Security Attributes on Files and Directories

These attributes are present on objects in Solaris file systems: User Id, Group Id, Permission Mode, Access ACL (optional), Default ACL (optional). Along with the Solaris attributes, Trusted Solaris files and directories have extended security attributes. The following table gives the extended security attributes required by Trusted Solaris security policy.

TABLE 9-1 File and Directory Attributes From Trusted Solaris Operating System

Extended Attributes	Description of Extended Trusted Solaris Attributes
Label	The label of the file or directory.
Forced Privileges	Optional. The set of privileges that an executable file is guaranteed to have available at start of execution. Must be a subset of the allowed privileges.
Allowed Privileges	Optional. The maximum set of privileges that an executable file is allowed to use during its execution. (Editing executable files causes them to lose all their privileges. Therefore, limiting the privileges that an executable can use to those in its allowed set provides a protection against Trojan Horses, since programs cannot use inheritable privileges if the programs have been edited.) Must be a superset of the forced privileges.
File Attribute Flag	Optional. The only supported file attribute flag is <code>public</code> . If the <code>public</code> flag is set, audit records are not generated when certain read operations are performed, even when these read operations are part of a preselected audit class, with one exception. If the audit pseudo event for use of privilege (<code>AUE_UPRIV</code>) is included in a preselected audit class and if the operation involves the use of privilege, then an audit record is always generated. With the previous exception, the following read operations are not audited when the <code>public</code> flag is set: <code>access(2)</code> , <code>fgetcmwlabel(2)</code> , <code>fgetslname(2)</code> , <code>fstatvfs(2)</code> , <code>getcmwfsrange(2)</code> , <code>getcmwlabel(2)</code> , <code>getfpriv(2)</code> , <code>getmldadorn(2)</code> , <code>getslname(2)</code> , <code>lgetcmwlabel(2)</code> , <code>lstat(2)</code> , <code>mldlstat(3TSOL)</code> , <code>mldstat(3TSOL)</code> , <code>open(2)</code> read only, <code>pathconf(2)</code> , <code>preadl(2)</code> , <code>readl(2)</code> , <code>readlink(2)</code> , <code>stat(2)</code> , <code>statvfs(2)</code>
Directory Attribute Flag	Optional. Flag indicating that a directory is an MLD

Specifying Security Attributes on Files and Directories

The Trusted Solaris File Manager lets users and administrators change permissions and lets authorized users and administrators set privileges and labels on files and directories. Authorizations are required to change privileges and labels. Additional authorizations are required when the change is outside DAC or MAC policy.

Changing Labels and Privileges

The `File Manager Selected` menu has a `Change Labels` option to set the label, which can also be done on the command line by any account that has the `setlabel(1)` command in one of its profiles. The `File Manager Selected` menu also has a `Change Privileges` option to set forced and allowed privileges on executable files. Changing forced and allowed privileges can also be done on the command line by any account that has the `setfpriv(1)` command in one of its profiles.

The following authorizations are required in order to set privileges and labels through the `File Manager Selected` menu options:

- Setting privileges requires the `Set File Privileges` authorization.
- Upgrading file and directory labels requires the `Upgrade File Label` authorization.
- Downgrading file and directory labels requires the `Downgrade File Label` authorization.

The following figure shows the `File Manager Selected` menu when the account has the required authorizations. See “To Change Labels and Privileges on Files and Directories Using the File Manager” on page 234 for how to change labels and privileges.

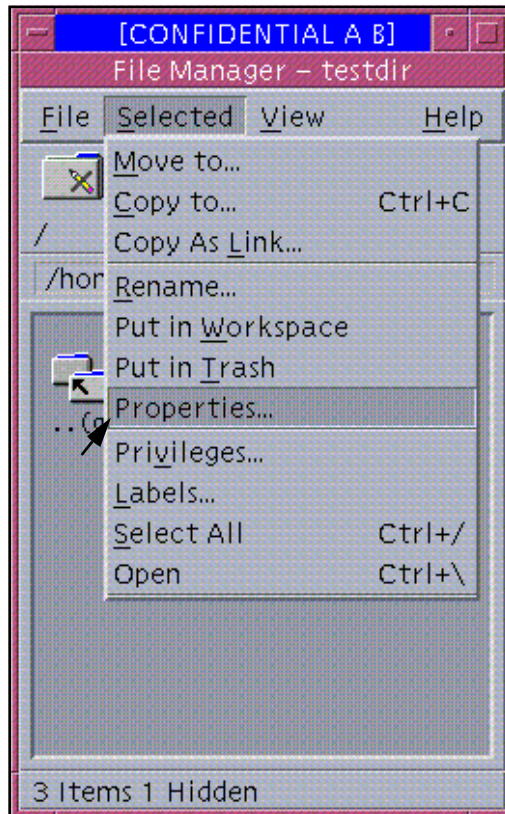


Figure 9-1 File Manager Selected Menu for an Authorized User

Changing File and Directory Attribute Flags

The `getfattrflag(1)` command gets the security attribute flags of a file or directory and the `setfattrflag(1)` command sets the public object flag on a file and sets the MLD flag on a directory.

Security Attributes on File Systems

File systems supported by the Trusted Solaris system are characterized by whether their attributes can be changed or not. When the attributes can be changed, they are called *variable attribute* or *variable* file systems. File systems that do not support the Trusted Solaris extended security attributes are called *fixed* because any attributes assigned to them (either at mount time or by default) cannot be altered.

Following are more details relevant for understanding and managing the various types of variable and fixed file system types:

- All `ufs`-type file systems are variable and therefore, all file systems installed with the Trusted Solaris system are variable.

For example, if you connect a hard disk containing an unlabeled file system directly to a Trusted Solaris computer, when the file system is `ufs`-mounted the unlabeled file system becomes a variable file system, with a default set of attributes shown in Table 9-2.

- An `nfs`-type file system mounted from a Trusted Solaris or TSIX NFS server is variable.
- An `nfs`-type file system mounted from an NFS server running another operating environment is fixed.
- `tmpfs` file systems are variable.
- These file system types are always fixed: `fdfs`, `hsfs`, `pcfs`.
- The `lofs`-type file system's attributes are those of the underlying file system. See "Mounting Various Types of File Systems in the Trusted Solaris System" on page 226 for more information.

The following table shows the security attributes for variable-attribute file systems, with the default values that are used when none are specified.

TABLE 9-2 Variable File System Security Attributes with Defined Settings

Attribute	Description	Defaults
MLD prefix	The characters to use for the MLD prefix for MLDs on this file system	.MLD.
Label Range	The minimum and maximum sensitivity level for files and directories created on this file system	ADMIN_LOW to ADMIN_HIGH
Label	Label to infer for all files and directories on this file system that do not have an explicit label	none. NOTE: Files and directories in a fixed-attribute file system get a label assigned at mount time when they are UFS-mounted. For that reason, it is essential to assign a label to an unlabeled file system when it is UFS-mounted .

TABLE 9-2 Variable File System Security Attributes with Defined Settings (continued)

Attribute	Description	Defaults
Forced Privilege Set	Set of forced privileges to infer for all executable files on this file system that do not have explicit forced privileges	none
Allowed Privilege Set	Set of allowed privileges to infer for all executable files on this file system that do not have explicit allowed privileges	none

The Label Attribute

In variable file systems the label of each object is set when it is created and can be changed by an authorized user. In fixed file systems, a single label is assigned when the file system is mounted. The label can be changed only if an object is moved from the fixed file system. Because they are configured to have a single label when mounted on Trusted Solaris hosts, fixed attribute file systems are also referred to as *single-label file systems*.

The label is obtained differently when a fixed-attribute file system is NFS-mounted than when it is PCFS-mounted from a floppy disk or HSFS-mounted from a CDROM.

- An NFS-mounted file system is assigned the label that is specified in the `DefaultLabel` setting in the Security Families template assigned to the remote computer from which the file system is NFS-mounted.
- For a PCFS- or HSFS-mounted fixed-attribute file system, the label is specified at mount time, either on the mount command line or in an entry in the `vfstab_adjunct(4)` file.

Specifying Security Attributes on Variable File Systems

When site security policy requires, the security administrator role can:

- Use the `getfsattr(1M)` command to get the security attributes of a file system.
- Use the `setfsattr(1M)` command to tune the attributes set on an already-existing file system (See “To Set Security Attributes on a File System ” on page 237).



Caution - Do not change or explicitly set the security attributes of the `/`, `/usr`, or `/var` file systems on a Trusted Solaris host. The results are unpredictable.

Specifying Security Attributes on Fixed File Systems

When mounting a fixed-attribute file system, the security administrator role can specify security attributes in the following ways:

- On the command line using the `mount` command with the `-S` or `-o` options
See the `mount(1M)` man page for the security attributes and other security-related options that can be applied

Note - Most of the keyword=value pairs used to specify security attributes with the `-S` can be specified with the `-o` option. If a keyword is followed by multiple values separated by commas, the keyword must be specified with the `-S` option because comma-separated values are not allowed after `-o`. Use of the `-o` option is preferable because it allows more. For more about the security-related mount options that can be specified with the `-o` option, see “Mount Options Used for Protection ” on page 228.

- In the `vfstab_adjunct(4)` file (Note that `vfstab_adjunct` is protected at `ADMIN_HIGH`. See “To Specify Mount-time Security Attributes in the `vfstab_adjunct` File” on page 239.
- In `/etc/auto_master` file other `autoFs` maps (see `automount(1M)`).

Any attributes specified at mount time are applied to all the files and directories in the mounted file system, if the files or directories themselves do not have the attribute. Any attributes on the file or directory are used. If the file or directory does not have an attribute and none is specified at mount-time, the defaults shown in Table 9-3 apply.

In fixed attribute file systems, the security attributes cannot change on an object as long as the object resides in the file system.

If, for example, the mounted file system `/spare` contains a file called `test`, no one can change the label of `/spare/test`. However, if `/spare/test` is copied into another directory such as `/tmp` or `/export/home/secadmin`, its label can be changed.

Table 9-3 shows the attributes that can be specified for a fixed attribute file system when the file system does not support the attribute, and the default values that apply if no value for the attribute is supplied.

TABLE 9-3 Attributes Assignable to Fixed File Systems

Attribute	-S or -o Option Keyword to Use When Mounting	Default Values
MLD prefix	mld_prefix	.MLD.
Label Range	low_range, high_range	ADMIN_LOW to ADMIN_HIGH
Label	slabel=	If a fixed file system is being mounted from a CD-ROM or floppy disk: the label of the mount point (which is changed to be the mounting process's label) If a fixed file system is being mounted from a NFS server: the default label administratively assigned to the server in its trusted network entries
Forced Privilege Set	forced=	None
Allowed Privilege Set	allowed=	None

The following example shows a command line to NFS-mount a fixed attribute file system called /spare from an NFS server running the Solaris operating environment. The server is called outside. /spare is mounted with a label of INTERNAL_USE_ONLY using mount with the -S option on the command line as shown here:

```
$ mount -F nfs -S "slabel=INTERNAL_USE_ONLY; outside:/spare /spare
```

Mounting Various Types of File Systems in the Trusted Solaris System

Trusted Solaris mount(1M) can be used to mount the types of file systems shown in the following table.

The table includes cross-references to mount_* mount man pages, when they are available for the named filesystem type, such as mount_nfs(1M) and mount_ufs(1M). The mount man page describes security attributes that can be set for any file system type that supports using the -S option at mount time and describes the privileges, UID and GID that mount needs in order to succeed. The

mount_* man pages give the subcommands that can be entered with the -o option for each filesystem type. See also “Security Attributes on File Systems” on page 222 and following for more about security attributes.

TABLE 9-4 Mount Types, Examples, and Notes

Type	When Used	Notes
FDFS	A pseudo file system type that allows a program to access its own file descriptors through the file name space	MAC and DAC isolation are assured because each process can access/see only its own file descriptors. The mode (0666), group (root), and owner (root) are fabricated by the kernel and are not used in any DAC decisions. The label is of the backing file or directory. This is a fixed attribute file system.
HSFS	Mounts a file system from a CD device.	See <code>mount_hfs(1M)</code> . In the Trusted Solaris environment, the file system can be given fixed attributes at mount time.
LOFS	A pseudo file system type that allows virtual file systems to be created that provide access to existing files using alternate pathnames	See <code>lofs(7FS)</code> . In the Trusted Solaris system, the security attributes are identical to those of the underlying file system.
NFS	Mounts a file system from a remote NFS server.	See <code>mount_nfs(1M)</code> . NFS mounts can be performed on fixed and variable attribute file systems.
PCFS	Mounts DOS file systems from a diskette.	See <code>mount_pcfs(1M)</code> and <code>pcfs(7FS)</code> . No extended attributes can be set on this file system type.
PROCFS	A pseudo file system provides access to the image of each process in the system. The name of each entry in the <code>/proc</code> directory is a decimal number corresponding to a process-ID. The owner of each “file” is determined by the process’s real user-ID.	In a Trusted Solaris system, PROCFS is a variable attribute file system in which all the Trusted Solaris attributes are supported. Process access decisions are based on the DAC and MAC attributes of the <code>/proc</code> file, which are imputed from the underlying process’s DAC and MAC attributes. If the calling process has the <code>proc_owner</code> privilege, then the process can get information at the same label about processes not owned by the caller. If the calling process has <code>proc_mac_read</code> privilege, the process can get information about a process that is owned by the caller when the process’s label dominates that of the caller or is disjoint. The restrictions for modifying are more granular than the ones for reading. See the <code>proc(4)</code> man page.

TABLE 9-4 Mount Types, Examples, and Notes *(continued)*

Type	When Used	Notes
TMPFS	Mounts in memory a temporary file system that uses swap pages, either in primary memory or on swap storage. The contents disappear at reboot.	Often <code>/tmp</code> is mounted as a tmpfs. The advantage is a huge increase in speed of access to whatever the temporary file system contains, since the information is retrieved from memory instead of from a disk. See <code>mount_tmpfs(1M)</code> .
UFS	Mounts a file system from a local disk	See <code>mount_ufs(1M)</code> . UFS file systems can have fixed mount time attributes assigned or variable attributes assigned at creation or later. See “Specifying Security Attributes on Variable File Systems” on page 224.
AUTOFS	Automounting mounts file systems with the AUTOFS type .	See <code>automount(1M)</code> .

Note - The CACHEFS file system type is not supported.

The `vfstab_adjunct(4)` man page describes the `/etc/security/tsol/vfstab_adjunct` and the `automount(1M)` man page describes the automounting-related files, such as `/etc/auto_direct` files, where mount-time security options can be entered.

Mount Options Used for Protection

The `mount` command can be used with the `-o` option followed by one of four protection options, either on the command line or in the `vfstab(4)` file. Some options can be used to protect the data on the file system being mounted, while others prevent a Trojan Horse attack initiated from the mounted file system. The `mount` restrictions shown in the following table are supported on all file system types. The Default Values column shows the values used when no option is specified.

TABLE 9-5 Mount Restrictions, Default Values

Description	Default Value	Alternate Value
Disallow write operations	rw	ro
Ignore set user id bits on executables	suid	nosuid
Ignore forced privilege sets on executables	priv	nopriv
Disallow opens on device special files, preventing the use of devices from non-standard directory locations	devices	nodevices

Note - The ro and suid options to disallow writes and ignore set user ID bits are from the base Solaris version of mount.

Summary of Attributes on Various File System Types

The following table indicates how different file systems support the various file system attributes. See the key in Table 9-7.

TABLE 9-6 Attributes Supported by the Supported File System Types

Attribute	TNFS	UFS/TMPFS/SLNFS	PCFS/HSFS
Allowed privileges	FS	MT	MT
Forced privileges	FS	MT	MT
CMW label	FS	MT (sensitivity label only)	MT (sensitivity label only; from host's template)
MLD prefix	FS	MT	MT
Label range	FS	MT	MT
File system attribute flags	FS	none	none
Object attribute flags	FS	MT	MT

TABLE 9-6 Attributes Supported by the Supported File System Types *(continued)*

Attribute	TNFS	UFS/TMPFS/SLNFS	PCFS/HSFS
Mount flags	MT	MT	MT
Access ACL	OBJ	OBJ	none
File mode	OBJ	OBJ	*
File owner	OBJ	OBJ	*
File group	OBJ	OBJ	*
	Type	Where Attribute Obtained	
	FS	From the file system	
	MT	From attributes specified at mount time	
	*	For HSFS with Rock Ridge extensions: same as the object	

TABLE 9-7 KEY to Table 9-6

UFS	A UFS file system on a Trusted Solaris host
TNFS	A TNFS file system from a Trusted Solaris or TSIX server
TMPFS	A TMPFS file system
SLNFS	A NFSv2 file system or a NFSv3 file system from a single-label/unlabeled server
PCFS	A PCFS file system
HSFS	A HSFS file system

MLDs are supported only by the following file system types:

- `ufs` (always variable)
- `nfs-variable`
(NFS file systems mounted from Trusted Solaris servers)
- `lofs`, and
- `tmpfs`

Trusted Solaris Attribute Precedence Rules

A file or directory's attributes take precedence over the attributes on the containing file system. Attributes specified at mount-time take precedence over filesystem attributes already in effect for a filesystem. Any attributes not obtainable at mount time or from the file system are assigned from the defaults.

The following figure illustrates the rules.

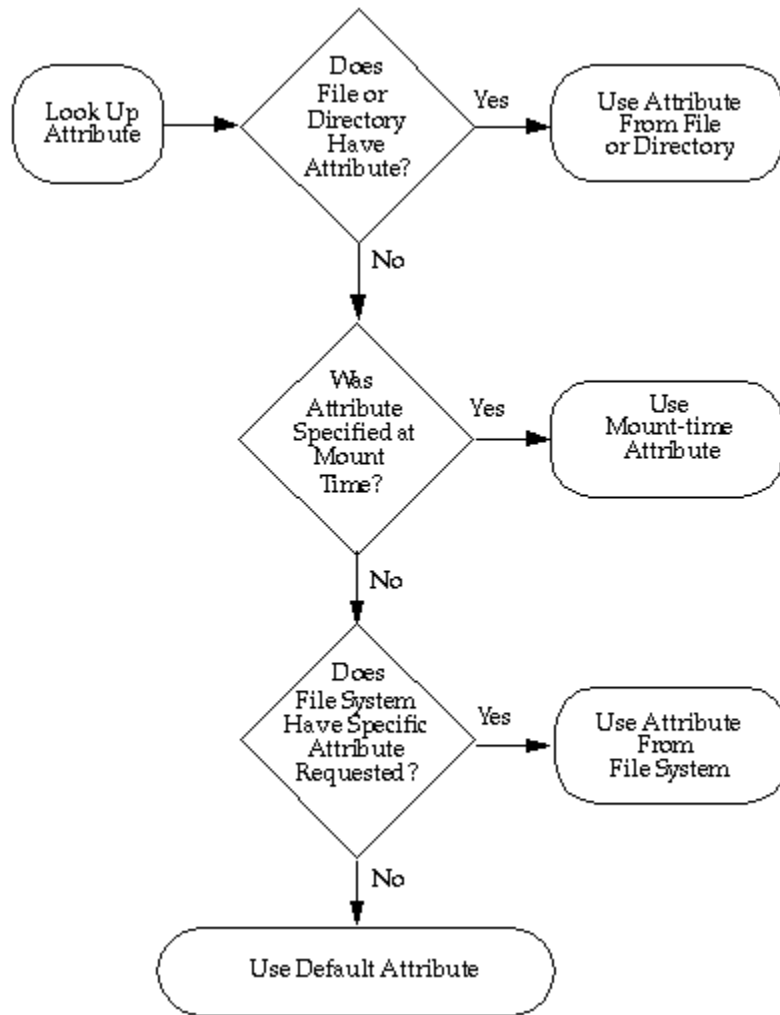


Figure 9-2 Trusted Solaris Attribute Precedence Rules

Trusted Solaris and NFS

Trusted Solaris supports both of the Network File System (NFS) protocols supported in the base Solaris operating environment and the Trusted Solaris 1.x release:

- NFS Version 2 (V2) (from the Solaris 1.x environment)
- NFS Version 3 (V3) (from the Solaris 2.5 and later compatible environment)

When a Solaris computer exports a file system using one of the NFS protocols above, the administrator of a computer running Trusted Solaris 7, 2.5.1, or 2.5 can specify the corresponding NFS protocol version to access the file system at a single label.

A Trusted Solaris computer can also specify the appropriate NFS protocol to export its own file systems to unlabeled client computers. A file or directory exported to an unlabeled client is *writable* if its label equals the label associated with the client computer in its trusted networking database entries. A file or directory exported to an unlabeled client is *readable* only if its label is dominated by the label associated with the client computer.

Communications with computers running Trusted Solaris 1.1 and 1.2 computers is possible only at a single label. Both systems must assign each other a template with the unlabeled host type specified with the same single label.

Any file system being mounted from a NFS server running Trusted Solaris 1.x must be mounted with *vers=2* and *proto=udp* mount options.

The NFS protocol used (whether it is NFS V2/V3, TNFS, TSIG/TNFS) is independent of the type of the local file system; rather, it depends on the type of the exporting computer's operating system. The file system type specified to the `mount` command or in the `vfstab` for remote file systems is always *nfs*.

Exporting Directories

Exporting directories (sharing) for mounting by other computers is done the same way it is done in the base Solaris system. Two new Trusted Solaris mount options `nodevices` and `nopriv` can also be used when sharing file systems. See "To Share a Directory for Mounting by Other Computers " on page 240.

Troubleshooting Mount Failures

If an attempted mount fails, and if all the standard setup has been done as required in the base Solaris system (as described in the *Solaris System Administration Guide, Volume II*), do the steps in "To Troubleshoot Mount Failures " on page 243.

File and File System-related Procedures

▼ To Back Up Files

1. **Assume the oper role or any other role with the Media Backup rights profile.**
2. **Use one of the following backup methods:**
 - `/usr/lib/fs/ufs/ufsdump` for major backups
 - `/usr/sbin/tar -T` with other options for small backups
 - a script calling either of the above commands



Caution - Only these commands will preserve security attributes and can read multilevel and single-level directories correctly.

To Restore Files

1. **Assume the system administrator role or any other role with the Media Restore rights profile.**
2. **Use one of the following methods:**
 - `/usr/lib/fs/ufs/ufsrestore` for major restores
 - `/usr/sbin/tar -T` with other options for small restores
 - a script calling either of the above commands



Caution - Only these commands will preserve security attributes and can read multilevel and single-level directories correctly.

▼ To Change Labels and Privileges on Files and Directories Using the File Manager

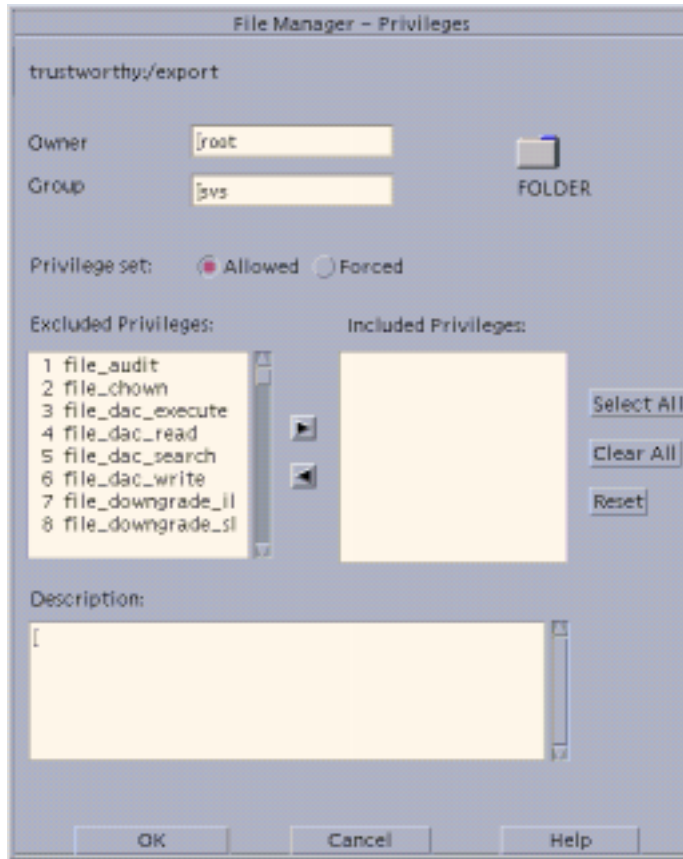
1. **Assume the security administrator role.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. **Bring up the File Manager and highlight the file whose privileges or label you wish to change.**

3. **To change privileges, choose Privileges from the Selected menu.**

The File Manager Privileges dialog box displays as shown below.



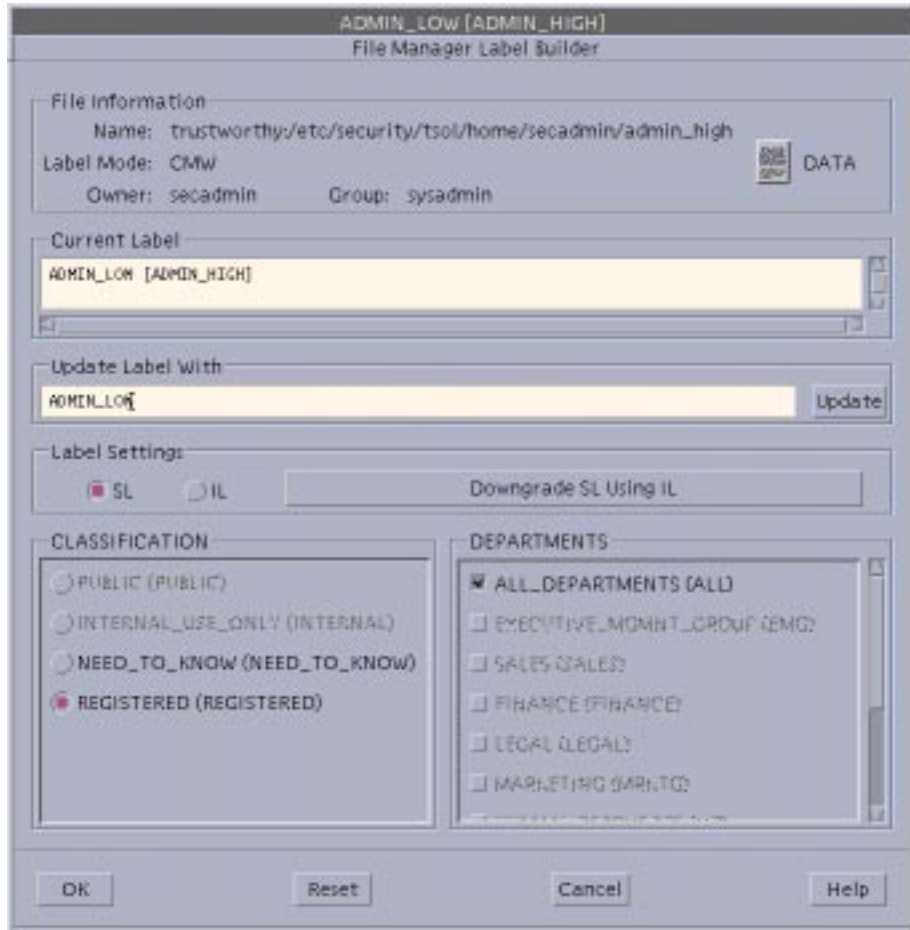
a. **On the File Manager Privileges dialog box, check the button for Allowed or Forced.**

b. **Move the desired privileges from the Excluded to the Included list.**

c. **Click OK.**

4. **To change labels, choose Labels from the Selected menu.**

The File Manager Label Builder Displays



- a. **On the File Manager Label Builder dialog box, enter a label.**
Do either of the following steps:
 - Type in the text entry field under `Update With`.
 - Click the desired classification, compartments or markings, as appropriate.
- b. **Click OK.**

▼ To Specify Alternative Security Attributes While Creating a Local File System

1. **Assume the administrator role and go to an ADMIN_HIGH workspace.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. **Using the File Manager or the `mkdir(1)` command in a `pfsh(1M)` shell, make the mount point directory.**

```
$ mkdir /newpublic
```

3. **Use the Set Mount Points Action to open the `vfstab(4)` file for editing.**
See “To Launch Administrative Actions from a Local Application Manager” on page 43, if needed.
4. **Make an entry for the file system in the `vfstab(4)` file.**

```
/dev/dsk/c0t3d0s3 /dev/rdisk/c0t3d0s3 /newpublic ufs 2 yes -
```

5. **Write and quit the file.**

```
:wq
```

6. **Assume the security administrator role and go to an `ADMIN_LOW` workspace.**
7. **In a profile shell [`pfsh(1M)`], execute the `newsecfs(1M)` command with the options that specify the desired alternative security attributes, then mount the file system.**

The following example sets a label range of `SECRET` to `SECRET`.

```
$ newsecfs -l "Secret;Secret" /newpublic  
$ mount /spublic
```

▼ To Set Security Attributes on a File System

1. **Assume the administrator role and go to an `ADMIN_HIGH` workspace.**
See “To Log In and Assume an Administrative Role” on page 32.
2. **In a profile shell, enter `umount(1M)` to unmount the file system.**

```
$ umount /spublic
```

3. Use the **Set Mount Points** action to open the `vfstab(4)` file for editing. See “To Launch Administrative Actions from a Local Application Manager” on page 43, if needed.
4. Make sure that an entry exists for the file system in the `vfstab(4)` file.

```
/dev/dsk/c0t3d0s4 /dev/rdisk/c0t3d0s4 /spublic ufs 2 yes -
```

5. Assume the security administrator role and go to an `ADMIN_LOW` workspace.
6. In a profile shell, execute the `setfsattr(1M)` command with the appropriate arguments, then remount the file system. The following example sets a label range of `SECRET` to `SECRET`.

```
$ setfsattr -l "Secret;Secret" /public  
$ mount /spublic
```



Caution - Do not use proprietary names for mounted file systems. The names of mounted file systems are visible to every user.

▼ To Specify Mount-time Security Attributes on the Command Line

1. Assume the administrator role and go to an `ADMIN_LOW` workspace. See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. In a profile shell, enter the `mount` command, using the `-S` option followed by any security attributes that you wish to specify.

```
$ mount -F tmpfs -S "allowed=all;forced=all" swap /mnt
```

The example mounts a `tmpfs`-type file system, `swap`, on `/mnt` with all allowed and all forced privileges.

▼ To Specify Mount-time Security Attributes in the `vfstab_adjunct` File

1. **Assume the administrator role and go to an ADMIN_HIGH workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Use the Set Mount Points action to open the `vfstab(4)` file for editing.**
See “To Launch Administrative Actions from a Local Application Manager” on page 43, if needed.
3. **Specify the mount point as described in the `vfstab` man page and add filesystem-specific security options in the mount options column as desired.**
See the filesystem-specific options in the `mount_*` man page for the file system type.
The example below shows a filesystem type of `ufs` and the Trusted Solaris `nodevices` and `nopriv` mount options, along with `nosuid` from the base Solaris operating environment.

```
/dev/dsk/c0t3d0s4 /dev/rdisk/c0t3d0s4 /spublic ufs 2 yes nodevices,nopriv,nosuid
```

4. **Save and close the file.**

```
:wq
```

5. **Assume the security administrator role and go to an ADMIN_HIGH workspace.**
6. **Use the Set Mount Attributes action to open the `vfstab_adjunct(4)` file for editing.**
7. **Copy and paste the template entry at the top of the file, and modify the copy.**

```
#<mount point>; \  
#slabel=; \  
#forced=; allowed=; \  
#low_range=; hi_range=; \  
#mld_prefix=;
```

The example below gives the following security attributes to /spublic: all files in the file system get an slabel (label) of SECRET A, all allowed privileges, and all the file-related privileges.

```
#      assigns the Secret A label and label range, all file-related
#      forced privileges and all allowed privileges to an unlabeled file system
#
/spublic;\
slabel='Secret A';\
forced=file_audit,file_chown,file_dac_execute,file_dac_read,file_dac_search,\
file_dac_write,file_downgrade_sl,file_lock,file_mac_read,file_mac_search,file\
_mac_write,file_owner,file_setdac,file_setid,file_setpriv,file_upgrade_sl;\
allowed=all;\
low_range='Secret A';\
hi_range='Secret A';
```

8. Save and close the file.

```
⌘:~
```

▼ To Share a Directory for Mounting by Other Computers

1. Assume the administrator role in an ADMIN_LOW workspace.

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. Use the Share Filesystems action to open the /etc/dfs/dfstab file for editing.

See “To Launch Administrative Actions from a Local Application Manager” on page 43 and the `dfstab(4)` man page, if needed.

3. Make an entry for the file system you wish to export.

The following example exports a user’s home directory with the `nodevices`, `nopriv`, `nosuid`, and `rw` options.

```
share -F nfs -o nodevices,nopriv,nosuid,rw -d "My Home Directory" /export/home/roseanne
```

4. Save and close the file.


```
:wq
```

5. In a terminal, run `shareall(1M)` to tell the NFS daemon, `nfsd(1M)`, to reread `dfstab(4)` file.

```
$ shareall
```

6. Make sure that the NFS daemon is running.
 - a. Use `grep(1)` on the output of `ps(1)` to see whether the `nfsd` is running.

```
$ ps -ef | grep nfs
root  303      1  0 10:35:42 ?          0:00 /usr/lib/nfs/nfsd -a 16
```

Note - The NFS daemon does not start automatically unless an entry exists in the `dfstab` file during boot.

- b. If the NFS daemon is not running, go to step Step 7 on page 241 to start the NFS daemon.
 - c. If the NFS daemon is running, go to step Step 8 on page 242.
7. If needed, start the NFS daemon.
 - a. In a profile shell, go to the `/etc/init.d` directory.
 - b. Enter the `nfs.server start` command.

```
$ ./nfs.server start
```

- c. Ensure that the NFS daemon is now running.

```
$ ps -ef | grep nfs
root  303      1  0 10:35:42 ?          0:00 /usr/lib/nfs/nfsd -a 16
```

8. Enter the `share(1M)` command with no options to make sure that the file system is being exported.

```
$ share
- /spare/manuals rw "manuals"
```

▼ To Mount a TMPFS-type File System Using the Command Line

1. Assume the administrator role, and go to an `ADMIN_LOW` workspace.
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. In a profile shell, enter the `mount` command, using the `-S` option followed by any security attributes that you wish to specify.

```
$ mount -F tmpfs -S "allowed=all;forced=all" swap /mnt
```

The example mounts a tmpfs-type file system, `swap`, on `/mnt`.

▼ To Mount a CD-ROM with a HSFS-type File System

As any user or role, use the Device Allocation Manager to allocate the `cd_rom_N` device.

If a CD in an allocated CD-ROM device contains a file system, the user is queried whether or not to mount the file system. Answer yes and the file system is automatically mounted.

▼ To Automatically Launch a CD Player for an Audio CD-ROM

As described in Chapter 12, under “Handling of CD-ROM Devices” on page 287, if an allocated CD-ROM device contains an audio CD and if an audio action is specified in `rmmount.conf`, the audio action executes.

1. Assume the security administrator role in an `ADMIN_LOW` workspace.
See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. Use the `Admin Editor` action to open the `/etc/rmmount.conf` file for editing.

See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.

3. Add an action to automatically launch a CD player.

The following example shows how the security administrator role could make an action in `rmmount.conf` for a CD player called `workman` installed in `/usr/bin`.

```
action cdrom action_workman.so /usr/bin/workman
```

4. Save and close the file.

```
:wq
```

▼ To Listen to an Audio CD as any User or Role

1. Make sure speakers are connected to the CD-ROM device and turned on.
2. Make sure the procedure “To Automatically Launch a CD Player for an Audio CD-ROM” on page 242 has been done.
3. Allocate the audio and the `cd_rom_N` devices at your working label.
4. When prompted, insert the audio CD into the device.

The specified CD player program will automatically be launched.

▼ To Troubleshoot Mount Failures

1. Make sure that the computer sharing the file system has been assigned a template on the Trusted Solaris computer doing the mounting.

Use the SMC Security Families tool to make sure that an appropriate template is assigned to an IP address that covers the NFS server. Look for the entry using the toolbox for the appropriate scope. If the NIS+ naming service is being used, bring up the SMC with the NIS+ scope. If NIS is being used, bring up the SMC with the NIS scope. If no naming service is being used, use the Files scope. See “Assigning Security Attributes to Remote Computers and Network Interfaces” on page 204 for more about how to assign templates to computers.

2. **If the computer is not running the Trusted Solaris operating environment, make sure the computer has a valid label assigned in its template.**
The label at which you access the mounted directory must be the same as the label assigned in the computer's template.
3. **Ensure that the mount is being done by the administrative role with the `mount` command in one of its execution profiles.**
In the default configuration, the security administrator role specifies the security attributes of mounts while the administrator role takes care of the normal Solaris aspects of mounting.
4. **When mounting any file system from a NFS server running Trusted Solaris 1.x, make sure to use the `vers=2` and `proto=udp` options to `mount(1M)`.**

Managing Name Services

To achieve uniformity of user, host, and network attributes within a security domain with multiple Trusted Solaris computers, a naming service is used for distributing most configuration information. If a name service is not used, administrators should ensure that configuration information for users, hosts, and networks is identical in the local files on all hosts and any changes made on one host is made on all. See “Administering Remote Systems” on page 31, if needed.

Administering name services is described in the *Solaris 8 System Administrator Collection* in the following manuals:

- *NIS+ Transition Guide*
- *Solaris Naming Administration Guide*
- *Solaris Naming Setup and Configuration Guide*

Setting up a name service master and clients (either NIS or NIS+) is described in the *Trusted Solaris Installation and Configuration* manual.

This chapter describes the differences in managing a name service in a Trusted Solaris environment. This chapter includes the following major topics:

- “Managing Multiple Trusted Solaris Computers in a Security Domain” on page 246
- “Managing Standalone Trusted Solaris Computers” on page 246
- “Allowing the root Role or a New Role to Administer A Name Server” on page 247
- “Trusted Solaris NIS Maps and NIS+ Tables” on page 247
- “Adding Trusted NIS Maps or NIS+ Tables” on page 248
- “Adding a New Host and Giving It Credentials ” on page 248

This chapter includes the following procedure.

- “To Save NIS+ Tables and Restore Them After Reinstalling the Trusted Solaris Environment” on page 249

Managing Multiple Trusted Solaris Computers in a Security Domain

A Trusted Solaris NIS or NIS+ master can manage data for Trusted Solaris NIS or NIS+ clients or Solaris NIS or NIS+ clients.

A Trusted Solaris NIS+ master can also manage data for NIS clients (such as hosts running the 1.x version of the Trusted Solaris operating environment) if NIS compatibility mode is used. NIS compatibility mode requires slightly different setup procedures than for a standard NIS+ server. NIS compatibility mode has security implications for NIS+ tables. For the differences and security implications, see “Using NIS-Compatibility Mode” in the *NIS+ Transition Guide*.

Trusted Solaris computers cannot be clients of Solaris NIS or NIS+ masters.

A security domain is generally administered as a single domain with a single name service master. Multiple security domains may be administered together in a hierarchy of subdomains with multiple non-root masters under a single root master. There can be only one root master. Replica servers may also be created to provide backup query services; the replica is associated with a particular name service master (root or non-root) and responds to requests in the event that the primary master is unable to respond.

Configuration files that for one reason or another cannot be administered by a name service should be centrally administered and duplicated on individual hosts by other means.

Managing Standalone Trusted Solaris Computers

Trusted Solaris computers may or may not be connected to a network with computers running other operating environments. A standalone Trusted Solaris computer may either be configured as its own name service master server or configured with no name service. If a Trusted Solaris standalone computer is configured without a name service, the configuration information is maintained in the `/etc`, `/etc/security`, and `/etc/security/tsol` directories. The administrative tools in the Trusted Solaris version of the Solaris Management Console allow the administrative role to specify Files scope so that the information is stored locally.

Allowing the root Role or a New Role to Administer A Name Server

If site security policy allows, root's capabilities can be extended to allow the root role to do administration from a client, although this is not recommended.

For root to administer NIS+ from a NIS+ client, the name of the NIS+ client must be added to the NIS+ admin group using the `nisgrpadm(1)` command. If a new administrative role is created to administer NIS+ tables, an entry also must be added to the NIS+ admin group with the role's principal name (for example, `newrole.security.sun.com.`). If desired, see "To Enable Root or a New Role to Administer NIS+" on page 117.

Trusted Solaris NIS Maps and NIS+ Tables

Besides the standard databases listed in the "Information in NIS+ Tables" in *Solaris Naming Administration Guide*, Trusted Solaris includes the following NIS maps/NIS+ tables:

TABLE 10-1 Trusted Solaris Added NIS+ Tables

<code>audit_user(4)</code>
<code>exec_attr(4)</code>
<code>ipnodes(4)</code>
<code>prof_attr(4)</code>
<code>tnrhdb(4)</code>
<code>tnrhtp(4)</code>
<code>user_attr(4)</code>

Adding Trusted NIS Maps or NIS+ Tables

As in the base Solaris, the administrator role can add NIS maps or NIS+ tables with protected data fields. See these manuals:

- *Solaris Naming Administration Guide*
- *Solaris Naming Setup and Configuration Guide*



Caution - Do not add new rows to the default NIS+ tables or modify the access rules defined for existing table fields.

Adding a New Host and Giving It Credentials

The root role does this during initial configuration of the system, as described in “Configuring a NIS or NIS+ Client” in *Trusted Solaris Installation and Configuration*. Following the same procedure, the administrator role adds a new NIS or NIS+ client host using the Create NIS Client action or the Create NIS+ Client action in the System_Admin folder. .

Name Service-Related Procedures

To Save NIS Maps and Restore Them After Reinstalling the Trusted Solaris Environment

- ◆ **Use `yppcat(1)` to dump NIS maps into flat files and then propagate NIS maps from the files.**
See “Administering NIS” in *Solaris Naming Administration Guide* for how to propagate NIS maps from files.

▼ To Save NIS+ Tables and Restore Them After Reinstalling the Trusted Solaris Environment

1. Create a script or use another means to dump the NIS+ tables into ASCII files.

Note - It is a good idea to dump the NIS+ tables into ASCII files routinely, at least every time you make a change to NIS+.

a. To create a script, assume the security administrator role and use the Admin Editor action to create the script file at ADMIN_LOW.

The following example shows a script called `nisscript` that the administrator role can create to do the dumps and to create a list of group members for later re-creation of the groups table.

```
#!/bin/sh
# nisscript
# nisplus tables into ascii files
#

mkdir -p /var/nis-backup
chmod 700 /var/nis-backup
cp /etc/.rootkey /var/nis-backup/dot-rootkey

# standard Solaris and Trusted Solaris tables
# NOTE: Add any tables created at your site

cd /var/nis/data
for i in audit_user auth_attr aliases bootparams ethers \
exec_attr group hosts netgroup netmasks networks passwd \
prof_attr protocols rpc services timezone tnrhdb tnrhnp \
user_attr shadow
do echo $i
/usr/lib/nis/nisaddent -d $i >/var/nis-backup/$i
done

# Use the following if you have any key value tables

for i in sendmailvars tntime
do echo $i
/usr/lib/nis/nisaddent -d -t $i.org_dir key-value >/var/nis-backup/$i
done

# get a list of each group and list each member in each group

mkdir -p /var/nis-backup/groups.list
chmod 700 /var/nis-backup/groups.list
for i in `nisls groups_dir | grep -v ``
do nisgrpadm -l $i >> /var/nis-backup/groups.list/group.members
```

(continued)

```
done
```

- b. Assume the root role and run the `nisscript` created in the previous step at `ADMIN_LOW`.
2. For each group, execute `nisgrpadm -l group_name` to list each of its members and save the output for use in Step 7 on page 251.

```
$ nisgrpadm -l group_name
```

3. Copy the directory containing the ASCII dump files to a partition that you plan not to overwrite during installation or use `tar` to copy the files to tape or floppy.
4. After installation, if you did not save the ASCII dump files in a saved partition, as root at `ADMIN_LOW`, create a staging directory for the ASCII dumps of NIS+ tables and restore the files from tape or floppy.

The screen example illustrates what to do when restoring the ASCII NIS+ files to a `/setup/files` directory from a tape.

```
# cd /setup/files
# tar xv
bootparams
ethers
.
.
.
```

5. At the appropriate point in “Configuring the NIS+ Domain” in *Trusted Solaris Installation and Configuration*, re-create the NIS+ environment.

```
# nisserver -r -d domain-name.
```

Make sure to include the final period (.) in the domain’s name.

6. As the security administrator role, at ADMIN_LOW, after running the `nisserver` command, run the `nispopulate` command in a profile shell with the `-F` and `-p` options followed by the name of the directory where the ASCII dump files reside.

```
# nispopulate -F -p /setup/files
```

7. Re-create the NIS+ groups and add members manually from the list of group members created in the `nisscript` shown in the example in .
There is no easy way to recreate the NIS+ groups automatically.

Managing Printing

Standard Solaris print utilities and databases have been modified to meet Trusted Solaris requirements for:

- Label-based control of access to printers and to information about queued print jobs
- Automatic printing of labels and other handling information on printer output and on mandatory banner and trailer pages

Managing printing is essentially the same in the Trusted Solaris environment as it is in the Solaris environment. This chapter describes the aspects of managing printing that are unique to administering printing in the Trusted Solaris environment.

The System Administrator and the Security Administrator roles share printer administration duties. Both administrators follow basic printer administration procedures described in the *Solaris System Administration Guide, Volume 2* (see especially the “Print Management (Overview)” and “Setting Up Printers (Tasks)”). The responsibility for managing printers is assigned to the System Administrator role. The responsibility for managing printer security is assigned to the Security Administrator role.

This chapter covers the following topics:

- “Configuring Printers (Trusted Solaris Tasks and Roles)” on page 254
- “Managing Network Printers” on page 257
- “Controlling Whether Labels and Other Information are Printed on Print Jobs” on page 257
- “Setting Up Printers Without Support for Page Labels or Labeled Banner/Trailer Pages” on page 256

This chapter also provides the following procedures:

- “To Set Up Printing to a Non-Trusted Solaris Server ” on page 261
- “To Access the Printer Administrator Action” on page 261

- “To Configure an Attached Printer” on page 262
- “To Configure a Network Printer for Labeled Output” on page 263
- “To Configure a Restricted Label Range for a Printer Managed by a Trusted Solaris Print Server” on page 264
- “To Add Access to a Remote Printer” on page 266
- “To Allow Some Users to Print Without Banners and Trailer Pages ” on page 266
- “To Assign Printing-related Authorization(s) to an Account” on page 267
- “To Suppress the Printing of Page Labels on All Print Jobs” on page 267
- “To Allow Some Users to Print Jobs Without Page Labels ” on page 268
- “To Set Up Publicly-Readable Print Jobs from an Unlabeled Print Server” on page 268

Configuring Printers (Trusted Solaris Tasks and Roles)

The following table shows the tasks for configuring printers in Trusted Solaris, the recommended roles and the tools that perform each task, and the table provides links to procedures and other related documentation.

TABLE 11-1 Tasks for Configuring Printers

Role/Rights Profile	Task	Tool	Notes
System Administrator/ Device Management	Configures printers	Printer Administrator action	See “To Configure an Attached Printer” on page 262, “To Configure a Network Printer for Labeled Output” on page 263, and “To Add Access to a Remote Printer” on page 266. See also “Starting Solaris Print Manager” and “Setting Up Printers (Tasks)” in the <i>Solaris 8 System Administration Guide, Volume 2</i> and following for how to do the configuration. Note - Where the instructions tell you to become superuser, do the steps at ADMIN_LOW in the System Administrator role.
Security Administrator/ Printer Security	Specifies a restricted label range for a printer (optional). The default is ADMIN_LOW to ADMIN_HIGH.	The Set Printer Label Range action or the <code>add_allocatable(1M)</code> command	See “To Configure a Restricted Label Range for a Printer Managed by a Trusted Solaris Print Server” on page 264.

Printer clients can only submit print requests at labels that are allowed by the trusted network database entries for the printer client computer and printer server.

Allowing the Printing of PostScript Files

By default, users cannot print PostScript files. This restriction exists because a knowledgeable PostScript programmer could create a PostScript file that modifies the labels on the printer output.

If desired, the Security Administrator role can assign the authorization called `Print PostScript` (`solaris.print.ps`) to trustworthy users and role accounts. The Security Administrator role should do so only if the account can be trusted not to spoof the labels on printer output and if allowing anyone to print PostScript files is consistent with the site’s security policy.

Adding Support for Additional File Types

A filter provided with the Trusted Solaris printing system converts text files to PostScript. Files converted to PostScript by any installed filter programs can be trusted to have authentic labels and banner and trailer page text because the filter's programs are trusted programs that are run by the printer daemon.

A site's System Administrator role can install additional filters, which then can be trusted to have authentic labels and banner and trailer pages. See the "Managing Character Sets, Filters, Forms, and Fonts (Tasks)" in *System Administration Guide, Volume 2* for how to add filters.

Setting Up Printers Without Support for Page Labels or Labeled Banner/Trailer Pages

PostScript printers are the only types of printers with support for labels and other handling information on printer output and on mandatory banner and trailer pages. The following types of printers function correctly, but they do not support page labels or labeled banner and trailer pages.

- Non-PostScript printers
- Printers connected to a print server that is not running Trusted Solaris
- Network printers that have not been configured from a Trusted Solaris computer

Jobs sent to a network printer print without labels and trailer pages if the network is not being managed by a Trusted Solaris print server. The network printer would have been configured in one of the two following ways:

- Using the printer's own software supplied by the printer vendor to be a standalone node on the network
- Using LP printer administration commands on a print server that is not running Trusted Solaris

If desired, the Trusted Solaris computer can be set up to send jobs to a printer connected to or managed by a computer (print server) that is not running Trusted Solaris. Print servers connected to unlabeled servers can print jobs only at the single label that is specified for the print server in the trusted network databases on the

Trusted Solaris computer. Jobs print without labels or trailer pages and without security information on banner pages.

Printing from unlabeled computers to a printer on a Trusted Solaris print server is supported.

Note - A user submitting a job from a single-label computer to a Trusted Solaris print server cannot cancel that job and cannot remove the job from the print queue. When a user sends a job from a labeled computer, the trusted network provides the UID of the user sending the print request. For unlabeled computers, the UID of the sender of the job is not available, so the UID assigned to the print job does not match that of the submitting user.

Managing Network Printers

Network printers print labels on body pages and banner and trailer pages only if the printer is managed by a Trusted Solaris computer. See “To Configure a Network Printer for Labeled Output” on page 263 for how to set this up.

Note - A network printer can print jobs only at the single label specified in the template that is assigned to the network printer’s IP address.

Controlling Whether Labels and Other Information are Printed on Print Jobs

The Security Administrator role can change the default for the printing of labels on body pages in the following ways:

- Give users an authorization on the print server to allow them to print jobs without labels on the body pages or print jobs without banner or trailer pages.
See “To Allow Some Users to Print Without Banners and Trailer Pages ” on page 266
- Redefine fields in the `/usr/lib/lp/postscript/tsol_separator.ps` file on the print server in one of the following ways:
 - Completely disable the printing of labels on body pages for any users

“To Suppress the Printing of Page Labels on All Print Jobs” on page 267

- Specify that another label or other wording or label is printed on body pages for all users

By default, the Protect As classification is printed at the top and bottom of every body page. The “Protect As” classification is the dominant classification when the classification from the job’s label is compared to the minimum protect as classification that is defined in the `label_encodings` file.

The label printed at the top and bottom of banner and trailer pages as shown in the following figure is specified by means of the `/PageLabel` definition.

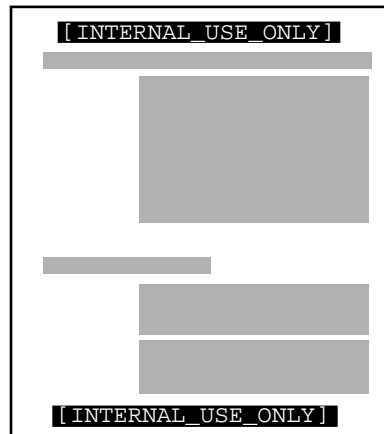


Figure 11-1 Job’s Label Printed on Body Pages

The `/HeadLabel` definition can be changed to put a different value or string at the top and bottom of the banner trailer pages or to print nothing at all.

Labels, Job Numbers, and Handling Information on Banner and Trailer Pages

The following figures show a default banner page and the differences in the default trailer page. The names of the various sections are shown because they are needed when configuring what appears.

All the text and the labels and warnings that appear on print jobs are site-configurable. The text can also be replaced with text in another language for localization.

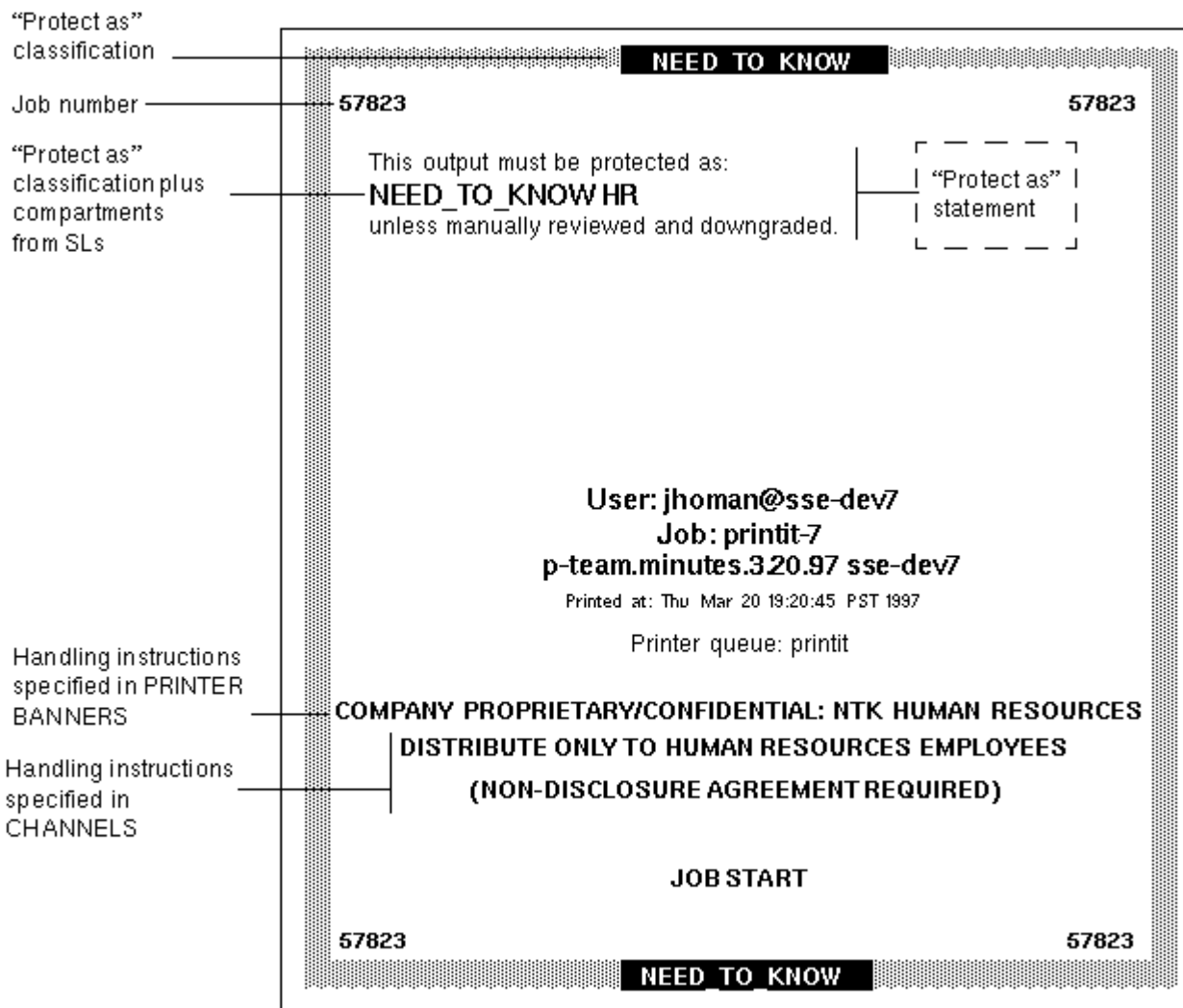


Figure 11-2 Typical Print Job Banner Page

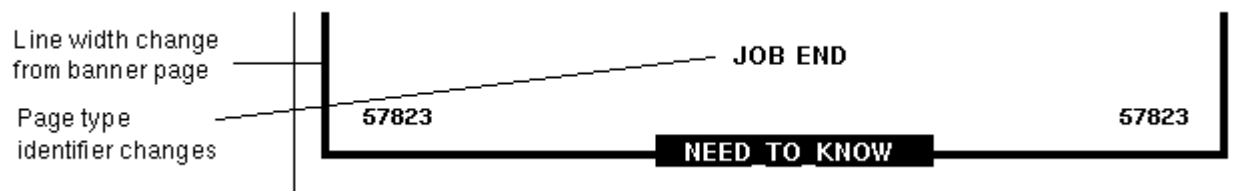


Figure 11-3 Differences on a Trailer Page

The following table shows aspects of trusted printing that the Security Administrator can change by assigning an authorization. For other printing-related authorizations see the Trusted Solaris Administrator's Overview.

TABLE 11-2 Modifiable Printing Features

What Can Be Changed	Authorization Name	How Value is Defined	How to Change
Whether individual users can print jobs without labels on body pages	Print without Label (solaris.print.unlabeled) authorization	SMC User Manager-> Properties-> Rights	Assign a rights profile with the Print without Label authorization to the user.
Whether all users can print jobs without labels on body pages	Print without Label (solaris.print.unlabeled) authorization	/etc/security/policy.conf file	Security administrator enters AUTHS_GRANTED=solaris.print.unlabeled in policy.conf.
Whether individual users can print jobs without banner or trailer pages	Print without Banner (solaris.print.nobanner) authorization	SMC User Manager-> Properties-> Rights	Assign a rights profile with the Print without Banner authorization to the user.
Whether all users can print jobs without banner or trailer pages	Print without Banner (solaris.print.nobanner) authorization	/etc/security/policy.conf file	Security administrator enters AUTHS_GRANTED=solaris.print.nobanner in policy.conf.
Text on banner and trailer pages			

The Security Administrator role can do the following to modify defaults that set labels and handling caveats on printer output:

- Localize or customize the text on the banner and trailer pages.
- Specify alternate labels to be printed in the various fields of the banner and trailer pages or at the top and bottom of body pages
- Change or omit any of the text or labels

Note - For how to do customizations or internationalization, see the comments in the `tsol_separator.ps` file.

Permitting Publicly-readable Jobs to Be Printed by Default Without Labeled Pages

Certain users, such as technical writers, need to produce publicly-readable documents that do not have labels printed on the top and bottom of the pages. If a printer connected to a Solaris print server is available, the Security Administrator role can set up the users' environments so that the publicly-readable jobs go to the printer connected to the Solaris computer while jobs at all other labels go to Trusted Solaris computers. See: "To Set Up Publicly-Readable Print Jobs from an Unlabeled Print Server" on page 268. The procedure requires understanding of how to set up user accounts as described in Chapter 3, and computer network entries as described in Chapter 8.

Printing-related Procedures

▼ To Set Up Printing to a Non-Trusted Solaris Server

1. **The Security Administrator role assigns a template to the print server with the desired label.**

The template is assigned to the IP address of the unlabeled print server using the Security Families tool.

See Chapter 8 for how the Security Administrator assigns a single label to an unlabeled computer.

2. **Users send print jobs to the single-label printer at the same label assigned to the print server.**

▼ To Access the Printer Administrator Action

1. **Assume the System Administrator role and go to an ADMIN_LOW workspace.**
See "To Log In and Assume an Administrative Role" on page 32, if needed.
2. **Access the Printer Administrator action from the System_Admin folder in the Application Manager.**

3. **Choose files to update local files or choose either NIS, NIS+(xfn) or NIS+ for a naming service.**

Go to “To Configure an Attached Printer” on page 262, “To Configure a Network Printer for Labeled Output” on page 263, or “To Add Access to a Remote Printer” on page 266.

▼ To Configure an Attached Printer

1. **Connect the printer to a serial or parallel port on a print server using the appropriate cable, as described in the printer’s installation manual.**
2. **Assume the System Administrator role on the print server, and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
3. **If the printer is connected to a serial port, make sure the correct baud rate is set, using the Serial Port tool from the Solaris Management Console Devices and Hardware manager.**
See the printer documentation for the correct baud rate. See also “Adjusting Printer Port Characteristics” in *System Administration Guide, Volume 2*.
4. **Bring up the Printer Administrator tool.**
See “To Access the Printer Administrator Action” on page 261, if needed.
5. **Choose New Attached Printer from the Printer menu.**
If needed, follow the procedure “How to Add a New Attached Printer With Solaris Print Manager” in the “Setting Up Printers (Tasks)” in *System Administration Guide, Volume 2*.



Warning - Do not change the Printer Type and File Contents settings from the default value of PostScript. If you do, printing will not work.

6. **If the default printer label range of ADMIN_LOW to ADMIN_HIGH is acceptable, you are done. To restrict the label range for the printer, go to “To Configure a Restricted Label Range for a Printer Managed by a Trusted Solaris Print Server” on page 264.**

▼ To Configure a Network Printer for Labeled Output

A network printer must be managed by a Trusted Solaris print server in order to print labeled output. A network printer prints only at a single-label assigned to it in a Security Families template.

1. **Pick a printer name to be used as its host name, and assign the printer an IP address.**
2. **Set up the printer as described in the printer's documentation.**
3. **Assume the System Administrator role on the Trusted Solaris print server, and go to an ADMIN_LOW workspace.**
See "To Log In and Assume an Administrative Role" on page 32, if needed.
4. **Add an entry for the printer using the Computers tool in the Solaris Management Console.**
The scope of the toolbox you have loaded determines whether the entry is made in the local hosts file, NIS map or NIS+ table.
 - a. **Double-click Trusted Solaris Configuration->Computers and Networks->Computers.**
 - b. **Select Action->Add Computer.**
 - c. **On the Add Computer dialog, type the printer name in the Name field, type the printer's IP address in the IP Address field, and click OK.**
5. **Create a new unlabeled template assigning it the ADMIN_HIGH label.**
 - a. **Double-click Trusted Solaris Configuration->Computers and Networks->Security Families.**
 - b. **In the Action menu, select Add->Template.**
 - c. **On the New Template dialog->Basic Information tab, do the following steps**
 - i. **Assign a Name.**
 - ii. **Select Unlabeled from the Host Type menu.**
 - iii. **Specify the Minimum Label and the Maximum Label as ADMIN_HIGH.**
 - iv. **Assign a Label and a Clearance of ADMIN_HIGH.**

- v. **Click OK in the New Template dialog.**

6. **Assign the new template to the host name or IP address of the printer.**
 - a. **Double-click the icon for the new template.**
 - b. **In the Action menu, select Add->Host.**
 - c. **In the New Remote Host Entry dialog, enter the Host Name and IP address.**
 - d. **Click OK.**

7. **Configure the printer on the Trusted Solaris computer using the LP administration commands.**

Complete the setup of the Network printer on the Trusted Solaris computer by following the procedure “How To Add A Network Printer Using LP Commands” in the “Setting Up Printers (Tasks)” in *System Administration Guide, Volume 2*.

▼ To Configure a Restricted Label Range for a Printer Managed by a Trusted Solaris Print Server

Do this procedure only if you need to restrict the label range for the printer. the default printer label range is ADMIN_LOW to ADMIN_HIGH.

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Bring up the Device Allocation Manager.**
Either select the Allocate Device option from the Trusted Path menu or launch the Device Allocation Manager action from the Tools subpanel on the Front Panel.
3. **Click the Device Administration button to display the Device Allocation: Administration dialog box.**
4. **Select the name of the new printer.**

5. Click the **Configure** button to display the **Device Allocation: Configuration** dialog box, as shown in the following figure.

Device Allocation: Configuration

Device Name: floppy_0

Device Type: fd

Min Label... ADMIN_LOW

Max Label... ADMIN_HIGH

Clean Program: /etc/security/lib/disk_clean

For Allocations From:

Trusted Path Non-Trusted Path

Allocatable By: Authorized Users
 No users
 All users
 Some As Trusted Path

Authorizations... solaris.device.allocate

Deallocation Options: Deallocate on Boot
 Deallocate on Logout

OK Reset Cancel

6. Change the label range as desired by clicking the **Min Label** and **Max Label** buttons and using the label builders that display to select the desired label.
7. Click the **OK** button on the **Configuration** dialog box to save the label changes, click the **OK** button on the **Administration** dialog box to close it, and then close the **Device Allocation Manager**.

▼ To Add Access to a Remote Printer


Note - If either NIS+ or NIS was specified as the naming service when the print server is configured, this procedure is not needed on any NIS+ or NIS clients in the domain.

1. **On the local computer, access the Printer Administrator.**
See “To Access the Printer Administrator Action” on page 261, if needed.
2. **See How to Add Printer Access With Solaris Print Manager in “Setting Up Printers (Tasks)” in *System Administration Guide, Volume 2*.**

▼ To Allow Some Users to Print Without Banners and Trailer Pages



Caution - If the `Always Print Banner` check box on the Printer Administrator dialog is checked, banner and trailer pages always print, even if the user has the `solaris.print.nobanner` authorization and uses the `--o nobanner` option to `lp`.

1. **Bring up the Printer Administrator on the print server.**
See “To Access the Printer Administrator Action” on page 261, if needed.
2. **Make sure that the `Always Print Banner` check box on the Printer Administrator is *not* checked.**


Always Print Banner
3. **Exit the Printer Administrator.**
4. **Make sure that the `solaris.print.nobanner` authorization is in one of the profiles assigned to each user or role that is allowed to print without banner and trailer pages.**
See “To Assign Printing-related Authorization(s) to an Account” on page 267, if needed.

5. **Instruct the user or role to submit jobs using the `lp` command with the option `-o nobanner`.**

```
trustworthy% lp -o nobanner staff.mtg.notes
```

▼ To Assign Printing-related Authorization(s) to an Account

1. **Assume the Security Administrator role.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Bring up the `User Accounts` tool.**
3. **Make sure that the desired print-related authorization is contained in one of the user’s rights profiles.**

▼ To Suppress the Printing of Page Labels on All Print Jobs

1. **Assume the Security Administrator role.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Use the Admin Editor action to bring up the `/usr/lib/lp/postscript/tsol_separator.ps` file for editing.**
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
3. **Find the following lines:**

```
%% To eliminate page labels completely, change this line to  
%% set the page label to an empty string: /PageLabel () def  
/PageLabel Job_SL_Internal def
```

Note - The value of `Job_PageLabel` may have been changed at your site.

4. Replace the value of `/PageLabel` with an empty parentheses.

```
/PageLabel () def
```

▼ To Allow Some Users to Print Jobs Without Page Labels

1. Make sure that the **Print Without Label** authorization is in one of the profiles assigned to each user or role that is allowed to print jobs without labels at the top and bottom of each page.

See “To Assign Printing-related Authorization(s) to an Account” on page 267, if needed.

2. Make sure that the user or role submits jobs using `lp` with the option `-o nolabels`.

```
trustworthy% lp -o nolabels staff.mtg.notes
```

Doing this procedure enables an authorized user or role to print jobs without labels when working at any label.

▼ To Set Up Publicly-Readable Print Jobs from an Unlabeled Print Server

1. In the `tnrhdb/tnrhtp` entries that define an unlabeled print server, assign to the print server the label that your site uses to identify files available to the general public.

For example, a site may label files that are available to the general public as `PUBLIC` or `UNCLASSIFIED`.

2. Do the following three steps for each user or role allowed to print publicly-readable files without page labels.
 - a. Make sure that the public label is in each account’s personal label range.
 - b. Instruct each user to define the `PRINTER` variable in the appropriate shell initialization file in the user’s publicly-labeled home directory SLD.
 - i. Go to the publicly-labeled home directory SLD.
 - ii. Open the `.login` or `.profile` file (as appropriate) for editing.

iii. Define the `PRINTER` variable to be the name of the printer connected to the unlabeled print server.

When a printer named `nolabels` is connected to a single-label print server whose label is `PUBLIC`, the `.login` or `.profile` file in the `PUBLIC SLD` directory would have the following environment variable defined.

```
setenv PRINTER nolabels
```

iv. Write and quit the file.

- a. Have each affected account log out and log in again to put the changed printer definitions in effect.**
- b. Have each affected account create and print jobs that need to be printed without labels from within the publicly-labeled SLD.**

Managing Devices

This chapter describes how to meet an organization's goals for protection of information on devices. This chapter includes the following topics.

- “Managing Allocation of Devices” on page 272
- “Setting a Label Range on a Workstation” on page 273
- “Setting a Label Range on a Local Printer” on page 273
- “Managing Device Access Policies Set in the device_policy File” on page 273
- “Managing Device Configuration and Setting Device Label Ranges” on page 274
- “Understanding the Device Allocation Manager ” on page 275
- “Training Authorized Users, Defining, and Enforcing Security Procedures” on page 283
- “Device-related Authorizations” on page 284
- “Tools for Device Management” on page 284
- “Device-Clean Scripts” on page 286
- “Handling of Allocated Devices at Boot” on page 288
- “Device-related Commands and Databases” on page 289

This chapter provides the following device-related procedures:

- “To Allocate a Tape Device and Use tar to Save Security Attributes on Exported Information” on page 289
- “To Set Device Policy on a New Device or Modify Policy on an Existing Device” on page 291
- “To Use the Device Allocation Administration Dialog Box” on page 292
- “To Add or Configure a Device” on page 293
- “To Configure a Serial Line for Logins” on page 295

- “To Assign Device-related Authorization(s) to an Account” on page 296
- “To Prevent Automatic Display of File Manager After Device Allocation” on page 298
- “To Change or Add a Device Clean Script ” on page 299

Managing Allocation of Devices

The Security Administrator role controls whether individual users are allowed to access certain devices by making the device nonallocatable or by granting or withholding the authorization that is needed to allocate the device.

The Security Administrator role can also control the labels at which a device can be accessed by specifying a restricted label range.

Here are some highlights of device management under Trusted Solaris:

- An unauthorized user in the default Trusted Solaris distributed system cannot allocate devices such as tape drives, CD-ROM drives, or floppy disk drives.
- A normal user with the device allocation authorization (`Allocate Device` by default), can import or export information only at the single label at which the user allocates the device.
- Users can allocate devices when logged in directly, when logged in remotely, from scripts and from user-developed applications. The `Device Allocation Manager` is used when the account is logged in directly. The `allocate(1M)` command can be used during a remote login session or from a script.
- Only one authorized user at a time can access an allocatable device. After allocation, the device must be deallocated. Deallocation ensures that the device is cleared of data and frees the device for allocation by another user.
- Other devices (such as the computer memory and hard disks) are automatically allocated and deallocated on behalf of all users, and any information contained on such devices is automatically cleared between allocations.
- The label range of each device handled by the device allocation mechanism can be restricted by the Security Administrator using the `Device Allocation Manager`. Normal users are limited to accessing devices whose label range includes the labels at which they are allowed to work. The default label range is `ADMIN_LOW` to `ADMIN_HIGH`.
- Nonallocatable devices are devices such as framebuffers and printers whose data is automatically cleared between users and whose label ranges can be restricted.
- Label ranges can be restricted for both allocatable and nonallocatable devices.

Setting a Label Range on a Workstation

To restrict direct login access through the console, the Security Administrator role can set a restricted label range on the framebuffer.

For example, a restricted label range might be specified to restrict access when a workstation is not physically protected. The label range lets users access the workstation only at a label within the framebuffer's label range.

Setting a Label Range on a Local Printer

When a host has a local printer, the Security Administrator role can set a restricted label range for that printer to limit the label range of jobs that it will print.

Managing Device Access Policies Set in the `device_policy` File

In the Trusted Solaris system, as in other UNIX systems, devices are represented by files called *device special files*. The discretionary access rules for devices are based on the same UNIX permission bits that apply to other types of files. The mandatory access rules that apply to devices are slightly different from those that apply to files or directories. The following table shows the default mandatory access control policy. These policies automatically apply to any new devices added to the system.

TABLE 12-1 Default Device Access Policy

Policy Type Names	Description	Default Policy
<code>data_mac_policy</code>	Label required to access the device	For reads and writes, the processes' label must equal the device's label.
<code>attr_mac_policy</code>	Label required to access the device's attributes (by <code>acl(2)</code> , <code>chmod(2)</code> , <code>chown(2)</code> , and <code>stat(2)</code>)	For read access to the device's attributes, the processes' label must dominate the device's label. For write access to the device's attributes, the processes' label must equal the device's label.

TABLE 12-1 Default Device Access Policy (continued)

Policy Type Names	Description	Default Policy
<code>open_priv</code>	Privilege required to open the device	No privileges are required.
<code>str_type type</code>	Only for STREAMS devices, specifies how the kernel stream head should control STREAMS messages	Device type stream. Unlabeled STREAMS message are allowed.

The Security Administrator role can change default policies and define new policies on each host by editing the `/etc/security/tsol/device_policy` file. Changes go into effect after a reboot. See the `device_policy(4)` man page for the keywords and values to use, and see also “To Set Device Policy on a New Device or Modify Policy on an Existing Device” on page 291.

Managing Device Configuration and Setting Device Label Ranges

After Trusted Solaris installation, during configuration, the Security Administrator role accepts or changes the default configuration for devices and their defined characteristics. After the system is up and running, the Security Administrator role can use the Device Allocation Manager to add a new device. The Device Allocation Manager can also be used to configure a device, by restricting the label range or changing any of the defaults assigned to the newly-created device. The Device Allocation Manager can be used to revoke an allocation, reclaim an allocated device from an allocate error state, or delete a device.

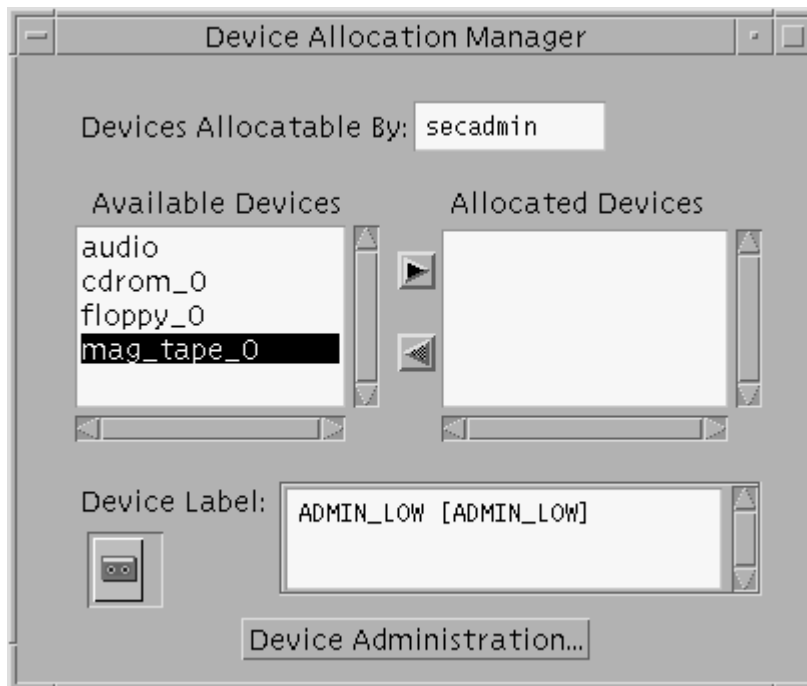
At system configuration, the Security Administrator needs to make the following decisions:

- Decide whether the default settings for the allocatable devices are consistent with the site's security policy.
- Decide whether to make additional devices allocatable.
- Decide which users, if any, should be allowed to allocate devices.
- Decide what authorization or authorizations to require for device allocation.
- Decide whether to require separate conditions for a device to be allocated locally from the trusted path and for a device to be allocated without the trusted path either remotely or from a script.

- Decide whether to use the default Allocate Device authorization or to create and require other authorizations for device allocation. See the example of adding new device allocation authorizations in “Adding New Authorizations” on page 69.
- Decide whether to accept or change the default label range settings on any listed allocatable or nonallocatable devices.

Understanding the Device Allocation Manager

The following figure shows the Device Allocation Manager. The list of devices that are available for allocation includes only devices physically present on the system.



The Device Allocation Manager can be used only by accounts that have the device allocation authorization, which is Allocate Device by default. Unauthorized users see an empty list under Available Devices. Also, when an allocatable device is currently allocated by another user or is otherwise not available, the authorized user does not see that device listed.

The `Device Administration` button displays on the bottom of the `Device Allocation Manager` only when an account has either one or both of the authorizations needed to administer devices, `Configure Device Attributes` or `Revoke` or `Reclaim Device`. (See Table 12-5 for the purposes of each authorization.)

When a Device is Not Available

If a user cannot see a device in the `Available Devices` list, the user needs to contact the responsible administrator:

If the user is not authorized but should be, the `Security Administrator` role can add the `Allocate Device` authorization to one of the account's profiles.

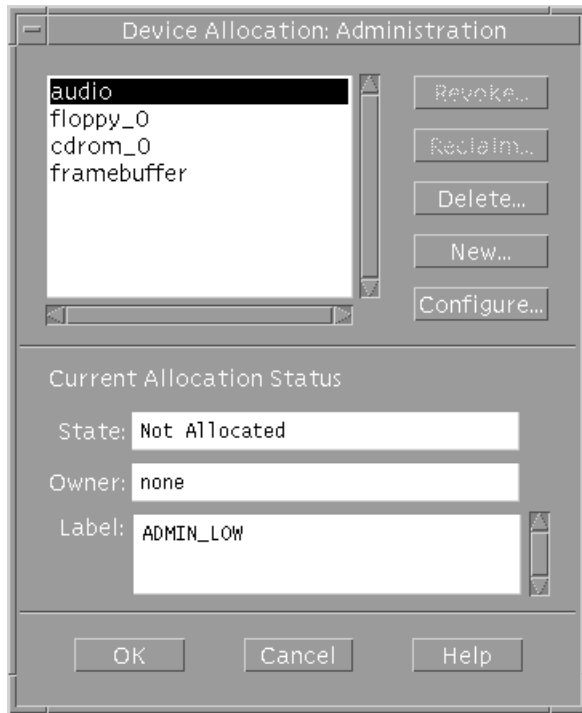
If the device is not listed because it is already allocated or it is in an `allocate error` state, the `System Administrator` and `Security Administrator` roles have the authorization to force deallocation of a device or to reclaim it from the error state.

Using the Device Administration and Configuration Dialogs

Clicking the `Device Administration` button launches the `Device Allocation: Administration` dialog box. This dialog box is used for reclaiming and revoking devices, deleting, or making entries for new devices. The `New` and `Configure` buttons launch a dialog for specifying configuring devices.

The `Device Allocation: Administration` dialog box displays a list of the devices and shows the status of the highlighted device: the `State`, `Owner`, and the `Label` of the device.

Click the `Device Administration` button to bring up the `Device Allocation Administration` dialog box.



Understanding the Device Administration Dialog

Revoke

If the **Revoke** button is active, the **State:** field for the highlighted device displays as: **Allocated**. If the account has the **Revoke** or **Reclaim Device** authorization, clicking the **Revoke** button forces deallocation of the selected device and changes the state to: **Not Allocated**.

Reclaim

If the **Reclaim** button is active, the **State:** field. for the highlighted device displays as: **Allocate Error State**. If the account has the **Revoke** or **Reclaim Device** authorization, clicking the **Reclaim** button releases a selected device from the **allocate error state** and changes the state to: **Not Allocated**.

New and Configure

If the account has the `Configure Device Attributes` authorization, clicking the `New` or `Configure` button brings up the `Device Allocation: Configuration` dialog box. See Figure 12-1.

Understanding the Device Configuration Dialog

This section describes the information that can be specified for a device using the `Device Allocation Configuration` dialog box shown in the following figure.

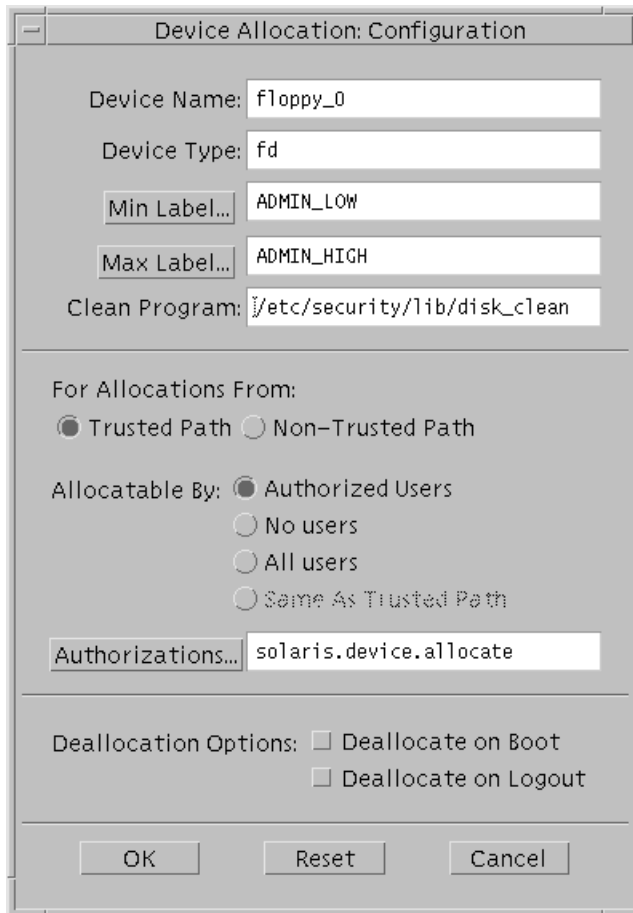


Figure 12-1 Device Allocation Configuration Dialog

Device Name and Device Type

The `Device Name` and `Device Type` display for the selected device. These fields cannot be edited.

Min Label and Max Label

Clicking the `Min Label...` and `Max Label...` buttons brings up a label builder. If no minimum label is specified at the time the device is created, the default is `ADMIN_LOW`. If no maximum label is specified at the time the device is created, the default is `ADMIN_HIGH`. See “Managing Device Configuration and Setting Device Label Ranges” on page 274 for more about setting a device’s label range. These fields are valid for allocatable and nonallocatable devices.

Clean Program

The `Clean Program` field allows the Security Administrator role to enter the path of a `device_clean(1M)script` for an allocatable device. If no `device_clean` script is specified at the time the device is created, the default is `/bin/true`. For how to write device clean scripts, see “Device-Clean Scripts” on page 286.

For Allocations From: Trusted Path or Non-Trusted Path

Devices can be configured to have different requirements depending on whether the devices are allocated through the trusted path or not. The differences between the two are types of allocations are described in the following list:

- **From Trusted Path (during a local login)**

An authorized user can allocate a device when directly logged into the computer that has that device by using the Device Allocation Manager. Allocations using the Device Allocation Manager are done through the trusted path.
- **Non-Trusted Path (during a remote login or by means of a script or program)**

Users can call the `allocate` command to allocate devices when they are logged remotely into the computer that has the device, or the `allocate` command can be called within scripts or in user-developed applications.

Sites that are concerned about the potential for misuse of the ability to allocate a device remotely can prevent remote device allocation or restrict it. For example, the site can required a device have one authorization when the device is allocated from the trusted path and another when the device is allocated without the trusted path.

The following table shows how the audio device is set up to be allocatable only during local login sessions.

TABLE 12-2 Specifying Only Local Allocation of the Audio Device

Device Name: audio

For Allocations From: Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.allocate

For Allocations From: Non-Trusted Path
Allocatable By: No users

The following table shows an example of treating local and remote allocation of atape device the same

TABLE 12-3 Treating Local and Remote Allocation of a Tape Device the Same

Device Name: st_0

For Allocations From: Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.allocate

For Allocations From: Non-Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.allocate

The following table shows an example how requiring two different authorizations for use of a CDROM device, one for remote allocations and one for local allocations.

TABLE 12-4 Requiring Two Different Authorizations for a CDROM Device

Device Name: cdrom_0
For Allocations From: Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.cdrom.local
For Allocations From: Non-Trusted Path
Allocatable By: Authorized Users
Authorizations: solaris.device.cdrom.remot

Allocatable By

The `Allocatable By` list offers three choices:

- Authorized users
- All users
- No users

This option is used most often for the framebuffer and printer, which do not have to allocated to be used. But it is also used as shown in Table 12-2, to prevent an allocatable device from being accessed.

If no authorization is specified at the time the device is created, the default is `All users`. If an authorization is specified, the default is `Authorized users`.



Caution - Because the `Add Allocatable` action sets up a new device as allocatable by all users, the Security Administrator needs to manually specify `Allocatable By No users` when a device, such as the frame buffer and printers, should not be allocatable by anyone.

Authorizations

The `Authorizations` button is active when the device is specified as `Allocatable By Authorized Users`. The default in the `authorizations` field is `solaris.device.allocate` (`Allocate Device`) authorization. The Security Administrator role can click the `Authorizations` button to bring up an authorization builder to change to another authorization, or can specify multiple authorizations. The following figure shows the `authorizations` dialog box. It displays all authorizations that are defined in the `auth_attr(4)` database.

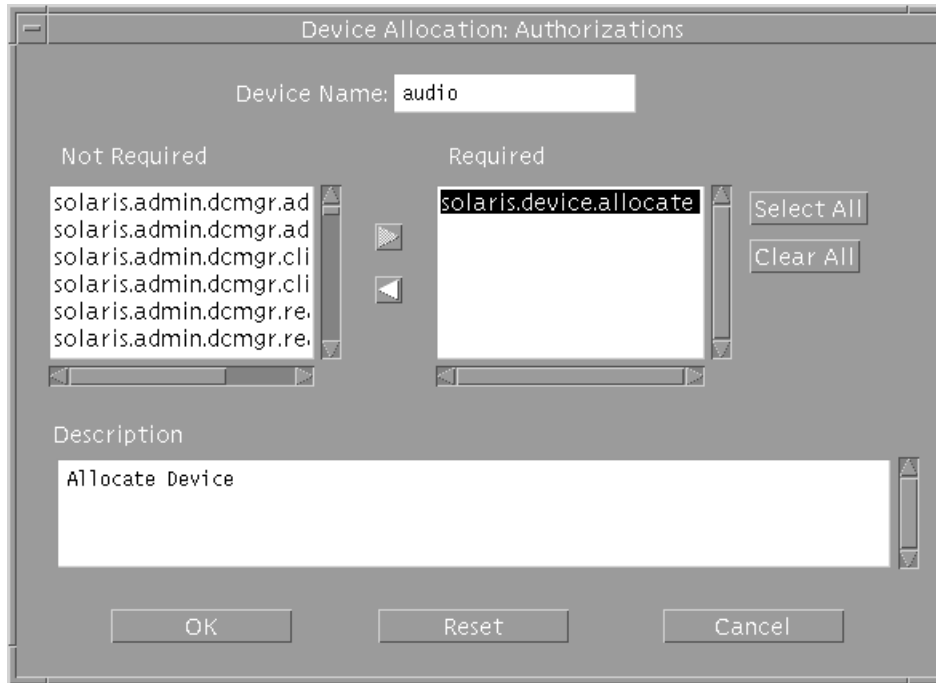


Figure 12-2 Clicking the Authorizations Button Displays the Device Allocation: Authorizations Box

See “Adding New Authorizations” on page 69 for an example of adding new device-related authorizations.

Deallocation Options

By default, any devices left allocated at boot are allocated when the system comes up and any devices left allocated at logout are allocated when the user logs in again. The `boot` command with the `-r` option can be used to force the deallocation of all devices.

Leaving devices allocated after logout could allow a user to access a device remotely that can only be allocated locally. For example, a user can log into one computer, allocate a device, and logout from the first computer. Then he can log into another computer and log in remotely back to the first computer. During that remote session, he could use the first computer’s microphone to listen into an interesting meeting.

The Deallocation Options allow the administrator to specify that any devices allocated by a directly-logged-in user are deallocated either at logout or at system boot or both. The administrator checks one or both or neither of the Deallocation Options:

- Deallocate on Boot
- Deallocate on Logout

These options do not affect any devices allocated outside the trusted path (either during a remote login, or from a script or customer-written application) .

Managing Remote Devices

The `add_allocatable(1M)`, and `remove_allocatable(1M)` commands, the `Add Allocatable Device` action, and the `Device Allocation Manager` make changes to local versions of the `device_allocate(4)` and `device_maps(4)` files on the host on which they are run.

Training Authorized Users, Defining, and Enforcing Security Procedures

By default, the Security Administrator role decides who has the authorization to allocate devices. The Security Administrator should make sure that any user authorized to use devices is trained and can be trusted to do the following:

- Properly label and handle any media containing exported sensitive information so that it does not become available to anyone who should not see it.

For example, if information at a label of `NEED TO KNOW ENGINEERING` is stored on a floppy disk, the person who exports the information must physically label the disk with the `NEED TO KNOW ENGINEERING` label and store the disk where only members of the engineering group with a need to know about the information can access it.

- Ensure that labels are properly maintained on any information being imported (read) from media on these devices.

An authorized user should make sure to allocate the device at the label that matches the label of the information being imported. For example, if a user allocates a floppy drive at `PUBLIC`, the user should not import information at any other label. A floppy labeled with `NEED TO KNOW ENGINEERING` should only be imported when the device is allocated at the `NEED TO KNOW ENGINEERING` label.

The Security Administrator role also is responsible for enforcing proper compliance with the above-mentioned requirements.

Device-related Authorizations

As is true for all other authorizations, the device-related authorizations are available to an account when they are specified in one of the profiles for an account, and when the profile has been assigned to the account by the Security Administrator role using the `User Manager`. See Chapter 3.

A site may define its own device allocation authorizations. For example, the site could set up an different authorization for each type of device, such as `allocate tape device` or `allocate floppy device`. To add a new authorization to the default list, if needed, read “Extending Extendable Security Mechanisms” on page 69 and do the procedure under “To Add An Authorization” on page 71 of Chapter 2 in this manual.

After a new authorization is added, it appears in the `Not Required` list in the `Device Allocation: Authorizations` dialog box.

The following table shows the device-related authorizations.

TABLE 12-5 Device Allocation, Configuration, and Management Authorizations

Name	Description
<code>Allocate Device</code>	Allows a user to allocate a device and to specify the label to associate with information imported from it, or exported to it.
<code>Configure Device Attributes</code>	Allows an account to configure a device. Device configuration includes setting the device name, type, label range, allocatable status, and allocation authorization list.
<code>Revoke or Reclaim Device</code>	Allows an account to deallocate a currently allocated device or reset an allocate error state to make a device allocatable again.

Tools for Device Management

The `add_allocatable(1M)`, and `remove_allocatable(1M)` commands, the `Add Allocatable Device` action, and the `Device Allocation Manager` make changes to local versions of the `device_allocate(4)` and `device_maps(4)` files on the host on which they are run. The responsible administrator needs to make the same changes on all Trusted Solaris computers on the system.

Ancillary Files for Allocatable Devices

Each allocatable device has an ancillary file, which is a zero-length file in `/etc/security/dev`. The ancillary file is also referred to as a DAC file because the file must not only exist but it has a different set of DAC permissions, owner, and group depending on whether the device is in one of the three possible following states:

- Allocatable
- Allocated
- In the `allocate error state`

The following table shows the DAC permissions, owner, and group for each of the possible states:

TABLE 12-6 Required Ancillary File Characteristics for Allocatable and Allocated Devices

Device State	DAC permissions (mode)	Owner	Group	Label
Allocatable	0000	bin	bin	ADMIN_LOW
Allocated	0600	<i>user</i>	<i>user's group</i>	<i>user's process's label</i>
Error State	0100	bin	bin	ADMIN_HIGH

Allocate Error State

As shown in Table 12-6, an allocatable device is in an *error state* if its ancillary file is owned by user `bin` and group `bin` with a device special file mode of `0100` and label of `ADMIN_HIGH`. One way that a device can be put into an `allocate error state` is by the `device_clean(1M)` scripts. A device-clean script puts a device into the `allocate error state` during deallocation until the user responds to prompts from the script and removable media is ejected. The `ReclaimReclaimReclaim` button on the `Device Allocation: Maintenance` dialog box can be used by an authorized user to reclaim devices from the error state.

Device-Clean Scripts

A device-clean script is run any time a device is allocated or deallocated. The user who allocates the device usually deallocates it. If necessary, the `Revoke` button on the `Device Allocation: Maintenance` dialog box can be used by an authorized user to forcibly deallocate a device.

If your site adds additional allocatable devices to the system, the added devices may need new scripts. See the following descriptions of the existing device-clean scripts for ideas on how they work, and see also “Writing New Device-Clean Scripts” on page 288.

Device-Clean Script for Tape Devices

The `st_clean` device-clean script is used for all tape devices.

The `st_clean` script uses the `mt(1)` command with the `-rewoffl` option to clean the device. When the script is run during system boot, it queries the device to see if it is on line and has any storage media in it. If necessary, the script prompts the operator to eject the storage media, and then it displays the appropriate label for the user to write on a physical label on the storage media.

Until deallocation completes, 1/4 inch tape devices are placed in the `allocate error` state, and 1/2 inch tape devices are taken off line. The `allocate error` state forces an authorized user to manually clean up the device before a user can allocate it again.

Device-Clean Scripts for Floppy Disks and CD-ROM

The `disk_clean` script is used for both floppy disk drives and CD-ROM devices. When the `disk_clean` script is run during boot time, any media found in a device is ejected. Whether it is run at boot time or when the device is deallocated, if the `eject` succeeds, the script prompts the user to affix to the media a physical label with the appropriate label. If the `eject(1)` command fails, the device is put in the `allocate error` state.

When a file system from either a floppy or CD is mounted as part of allocation, a `File Manager` pops up with the current directory set to the mount point. The `Security Administrator` role can prevent the automatic display of the `File Manager` by following the procedure in Step 1 on page 298. The mounting of file systems from floppy disks is handled differently from the mounting of file systems from CDs, as described in the following sections.

Handling of CD-ROM Devices

When a CD-ROM device is allocated, the user is queried whether or not to mount the CD-ROM. The user should answer `yes` if the CD contains a file system. When the answer is `yes`, the file system is automatically mounted. If the allocated CD-ROM device contains an audio CD, the user should answer `no`. When the answer is `no`, if an audio action is specified in `rmmount.conf`, the audio action executes. By default, no audio action is specified. To play an audio CD, the user must allocate both the audio and CD-ROM devices. The user can optionally manually invoke an audioplayer application after allocating the device.

For example, at a site where the commonly-used CD player, `workman`, is installed, the Security Administrator role can perform the following action in `rmmount.conf` to automatically bring up `workman` locally installed in `/usr/local/bin`.

```
action cdrom action_workman.so /usr/local/bin/workman
```

Handling of Floppy Devices

File systems on floppy disks are not automatically mounted at allocation because the user may wish to create a new file system over an existing file system already on the floppy. Programs such as `fdformat(1)` or `newfs(1M)` can create a new file system only if the file system on the floppy device is not mounted. Therefore, before mounting an existing file system on a floppy, the `disk_clean` script asks the user whether or not to mount the file system.

If a floppy disk is not formatted, the `disk_clean` script asks the user whether or not to format the floppy.

After the file system on a floppy is mounted as part of device allocation, a File Manager pops up with the current directory set to the mount point.

Device-Clean Script for Audio

The audiotool device is cleaned up using the `audio_clean` program.

This program performs an `AUDIO_DRAIN ioctl` to flush the device, and then an `AUDIO_SETINFO ioctl` to reset the device configuration to the default. In addition, this program retrieves the audio chip registers using the `AUDIOGETREG ioctl`, and any registers deviating from default are reset using `AUDIOSETREG ioctl`. Because the audio device does not contain any removable media, it does not require an external physical label, and therefore the label is not displayed by the `audio_clean` script.

Writing New Device-Clean Scripts

Some devices that can be made allocatable are modems, terminals, and graphics tablets. The task of making any of these devices allocatable would include writing a new device-clean script. Device-clean scripts should also be created for any added tape devices, except for Xylogics or Archive tape drives, which can use the default `device_clean(IM)` script (`/etc/security/lib/st_clean`).

The default location for device-clean scripts is `/etc/security/lib`.

Device-clean scripts must return 0 for success and greater than 0 for failure.

Failure or inability to forcibly eject the medium must put the device in the `allocate error` state.

The `deallocate` command passes four parameters to the device-clean scripts as shown here:

```
device_clean -[I|F|S] -[A|D] device_name label
```

The option letters `-I|F|S` help the script determine its running mode. `-I` is needed during system boot only. All output must go to the system console. `-F` is for forced clean up and `-S` is for standard cleanup. These are interactive and assume that the user is there to respond to prompts. With the `-F` option, the script must attempt to complete the cleanup if one part of the cleanup fails.

`[-A]-[D]` indicates whether the clean script is called from `allocate` or `deallocate`.

The `device_name` field is a string with the name of the device.

The `label` field is a hexadecimal representation of the label.

Handling of Allocated Devices at Boot

At boot time, by default, allocated devices are reallocated and remounted. Entering `boot` with the `-r` option forces deallocation of the devices. The administrator can change the default on the Device Allocation: Configuration dialog by checking one or both or neither of the deallocation options:

- Deallocate on Boot
- Deallocate on Logout

Device-related Commands and Databases

See the man pages for the following commands and databases:

TABLE 12-7 Device-related Commands and Databases

Command or File Name	Description
<code>allocate(1M)</code>	Device allocation command line interface
<code>add_allocatable(1M)</code>	Add a device to <code>device_allocate(4)</code> , <code>device_maps(4)</code> , and create an ancillary file in <code>/etc/security/dev</code>
<code>deallocate(1M)</code>	Device deallocation command line interface
<code>device_clean(1M)</code>	Device cleaning programs
<code>dminfo(1M)</code>	Report on specified device's entry in the <code>device_maps</code> file.
<code>list_devices(1M)</code>	List devices specified in the <code>device_maps</code> file
<code>remove_allocatable(1M)</code>	Remove a device from <code>device_allocate</code> , <code>device_maps</code> and delete its ancillary file from <code>/etc/security/dev</code>
<code>device_allocate(4)</code>	Database for managing allocatable and some nonallocatable devices
<code>device_maps(4)</code>	Database for device entries that are required for devices to be allocatable or to have their labels restricted

Device Management Procedures

▼ To Allocate a Tape Device and Use `tar` to Save Security Attributes on Exported Information

This procedure can be done by any user or role that has the `tar` command in a profile.

1. Use the Device Allocation Manager to allocate a tape device.

The example allocates a device named `mag_tape_0`. See the *Trusted Solaris User's Guide* for more about how to allocate devices and specify the label at which the device is allocated.

2. **Make sure the tape is physically labeled with the label of the current process, and insert the tape into the tape device when prompted.**

The window in the example is titled Device Allocation for mag_tape0 window.

```
st_clean: Insert tape into mag_tape0

st_clean: Make sure the tape is labeled CONFIDENTIAL

Press RETURN to quit window...
```

3. **Enter the tar command with the -T security option.**

```
trusted% tar cvT tartest
a tartest/(A) 1K

a tartest/ 0K

a tartest/file1(A) 1K
a tartest/file1 0K

a tartest/mld1/(A) 1K
a tartest/mld1/ 0K

a tartest/mld1/(A) 1K
a tartest/mld1/ 0K

a tartest/mld1/file50(A) 1K
a tartest/mld1/file50 1K

. . .
```

4. **Use the Device Allocation Manager to deallocate the device.**

Eject the tape from the device when prompted.

```
Please eject the tape in mag_tape_0
```

5. **Make sure to protect the exported information at the security level on the media's physical label.**

▼ To Set Device Policy on a New Device or Modify Policy on an Existing Device

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Determine the *driver_name* and *minor_name* and the device special file names for the device.**
 - a. **For a new device, do the following.**

- i. **Consult the hardware documentation for the device to obtain the device name and minor name and a list of all the physical device names.**

See also, *Writing Device Drivers*, PN 800-6502.

- ii. **Create a new entry for the device in the `/etc/security/device_maps` file.**

The name used for the device is arbitrary. In the third field, list all the physical device names for the device.

```
cdrom_0:\
  sr:\
    /dev/sr0 /dev/rsr0 /dev/dsk/c0t6d0s0 /dev/dsk/c0t6d0s1
/dev/dsk/c0t6d0s2 /dev/dsk/c0t6d0s3 /dev/dsk/c0t6d0s4 /dev/dsk/c0t6d0s5
/dev/dsk/c0t6d0s6 /dev/dsk/c0t6d0s7 /dev/rdisk/c0t6d0s0 /dev/rdisk/c0t6d0s1
/dev/rdisk/c0t6d0s2 /dev/rdisk/c0t6d0s3 /dev/rdisk/c0t6d0s4 /dev/rdisk/c0t6d0s5
/dev/rdisk/c0t6d0s6 /dev/rdisk/c0t6d0s7:\
```

The example shows all the physical and logical device names for the `cdrom_0` device.

- a. **For an existing device, find the device name and minor name by doing a long listing of the device.**

```
# ls -l /dev/dsk/c0t6d0s2
lrwxrwxrwx  1 root  root  51 Feb 29 1998 /dev/dsk/c0t6d0s2
-> ../../devices/sbus@1f,0/SUNW,fas@e,8800000/sd@6,0:c
```

In the final element of the pathname, the string before the `@` character is the driver name (`sd` in the example above) and the string after the colon is the minor name, (`c` in the example above).

3. Use the `Admin Editor` action to open the `/etc/security/tsol/device_policy` file for editing.
See “To Use the Admin Editor Action to Edit a File” on page 46, if needed.
4. When the default policy for devices is not consistent with your site’s security policy, create a specific entry or a wildcard entry for a new device or modify an existing entry for an already-specified device.

The default device policy is as shown in the following table. For how to specify alternate policy settings, see the `device_policy(4)` man page.

TABLE 12-8 Default Device Policy

Policy Types	Description	Default Policy
<code>data_mac_policy</code>	What label the process must have to access the device.	For reads and writes, the process’s label must equal the device’s label.
<code>attr_mac_policy</code>	How to handle access to the device’s attributes [(by <code>acl(2)</code> , <code>chmod(2)</code> , <code>chown(2)</code> , and <code>stat(2)</code>)	For read access to the device’s attributes, the process’s label must dominate the device’s label. For write access to the device’s attributes, the process’s label must equal the device’s label.
<code>open_priv</code>	Privilege required to open the device	No privileges are required.
<code>str_type type</code>	Only for STREAMS devices, specifies how the kernel STREAMS head should control STREAMS messages	Device type stream. Unlabeled streams messages are allowed.

5. Write the file and exit the editor.

▼ To Use the Device Allocation Administration Dialog Box

1. Assume the Security Administrator role and go to an `ADMIN_LOW` workspace or log in as a user with either the `Configure Device Attributes` or `Revoke or Reclaim Device` authorization.

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. **Select the `Allocate Device` option from the `Trusted Path` menu .**
3. **Click the `Device Administration` button to bring up the `Device Allocation Administration` dialog box.**
4. **Check the status of a device by highlighting the name of the device and looking at the `State:` field.**
5. **If the `State` field is `Allocate Error State`, click the `Reclaim` button to correct the error state.**
6. **If a device is `State` is `Allocated`, do one of the following:**
 - a. **Contact the `Owner` to deallocate the device.**
 - b. **If the `State` field is `Allocated`, click the `Revoke` button to force deallocation of the device.**
7. **To configure the device, go to “To Add or Configure a Device” on page 293.**

If the device is currently allocated, you must deallocate the device before configuring it.
8. **When you are done, click `OK` to save the changes and close the dialog box.**

▼ To Add or Configure a Device

Follow the instructions in the *Installing Device Drivers* manual for Solaris, if needed, then do the following Trusted Solaris-specific steps.

1. **If adding a new allocatable device, create a `device_clean` script, if needed.**

A tape drive can use the default `st_clean` script as is, or the script can be modified to suit the site’s security policy. Otherwise, a new `device_clean` script is needed. See “To Change or Add a Device Clean Script ” on page 299 if needed.
2. **Assume the `Security Administrator` role and go to an `ADMIN_LOW` workspace or log in as a role with the `Configure Device Attributes` authorization.**

See “To Log In and Assume an Administrative Role” on page 32, if needed.
3. **Select the `Allocate Device` option from the `Trusted Path` menu .**

“To Launch Administrative Actions from a Local Application Manager” on page 43, if needed.

4. Click the `Device Administration` button, and then click the `New` button on the `Device Allocation: Configuration` dialog.
 - a. Enter the `Device Name`.
 - b. Enter the `Device Type`.
 - c. In the `Device Map` field, enter the pathname for all the device special files associated with the device separated by spaces.
 - d. Change the minimum label from the default of `ADMIN_LOW`, if desired, by clicking the `Min Label...` button and using the label builder to specify a new label.
 - e. Change the maximum label from the default of `ADMIN_HIGH`, if desired, by clicking the `Max Label...` button and using the label builder to specify a new label.
 - f. Specify whether the device is treated differently for local (trusted path) or other logins (non-trusted path), whether it is allocatable, and whether allocation requires an authorization.
 - g. From the `Allocatable By:` list, choose one of the following options:

<code>Authorized Users</code> <code>No Users</code> <code>All Users</code> <code>Same as Trusted Path</code>

When configuring a printer, frame buffer, or other device that should not be allocatable, make sure to select `No Users`. When the device is specified as Allocatable by Authorized Users, the Authorizations field becomes active, and the `solaris.device.allocation` authorization name displays.

Note - Same As Trusted Path applies only when Non-Trusted Path is selected.

- h. Specify if you want the device to be deallocated on boot or on logout.

Note - The Deallocation Options only apply to devices allocated locally through the trusted path.

- i. Click **OK** to save your changes.

▼ To Configure a Serial Line for Logins

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace or log in as a role with the Configure Device Attributes authorization.**
See “To Log In and Assume an Administrative Role” on page 32, if needed.
2. **Bring up a SMC toolbox with the Files scope.**
The tools are shown in the following figure.

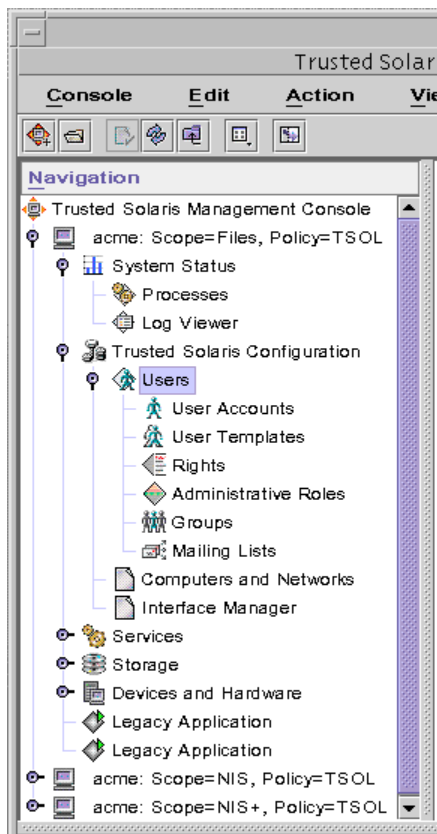


Figure 12-3 SMC Tools

3. **Select Devices and Hardware, and then Serial Ports.**
Follow the online help for how to configure the serial port.
4. **To restrict the label range, click the Device Administration button, and then click the Newbutton on the Device Allocation: Configuration dialog.**
The default label range is ADMIN_LOW to ADMIN_HIGH
 - a. **Enter /dev/term[a|b] for the Device Name.**
 - b. **Enter the tty for the Device Type.**
 - c. **Enter /bin/true for the Clean Program.**
 - d. **Enter /dev/term[a|b] again for the Device Map.**
 - e. **Change the minimum label from the default of ADMIN_LOW, if desired, by clicking the Min Label... button and using the label builder to specify a new label.**
 - f. **Change the maximum label from the default of ADMIN_HIGH, if desired, by clicking the Max Label... button and using the label builder to specify a new label.**
 - g. **Choose No Users under Allocatable By.**
 - h. **Leave the Deallocation Options unset.**
 - i. **Click OK to save your changes.**

▼ To Assign Device-related Authorization(s) to an Account

1. **Assume the Security Administrator role and bring up the User Accounts tool.**
See Chapter 1, “To Launch the Solaris Management Console” on page 39, and Chapter 5, if needed.
2. **Use the Rights tool, and make sure that the desired device allocation authorization or other device-related authorization(s) are contained in one of the user’s profiles.**
 - a. **Move a profile containing the device allocation authorization to the Available list, if desired.**

If the defaults have not been modified, you can include one of the profiles shown in the following table in the user's list of profiles to give an account the Allocate Device authorization.

TABLE 12-9 Default Device Allocation Authorization and Default Profiles that Include It

Authorization Purpose	Authorization Name	Default Profiles
Device allocation	Allocate Device	All Authorizations Convenient Authorizations Device Management Media Backup Media Restore Object Label Management Software Installation SSP Installation

- b. Move a profile containing the Revoke or Reclaim Devices authorization to the Available list, if desired.**

Note - The Revoke or Reclaim Device authorization is in administrative profiles, which should be given only to administrative role accounts.

The following table shows the Revoke or Reclaim Device authorization and the default profiles that contain it.

TABLE 12-10 Device Deallocation and Reclamation Authorization, Default Profiles that Include It, and Default Roles Assigned It

Purpose	Name	Default Profiles	Default Role
Forcing deallocation of an allocated device, or correcting a device's allocate error state	Revoke or Reclaim Devices	Device Management All Authorizations	secadmin Not assigned by default

- c. **Move a profile containing the `Configure Device Attributes` authorization to the Available list, if desired.**

The following table shows the `Configure Device Attributes` authorization and the default profiles that contain it.

TABLE 12-11 Device Configuration Authorization, Default Profiles that Include It, and Default Roles Assigned It

Purpose	Name	Default Profiles	Default Role
Configuring device's attributes: <code>device_clean</code> script, label range, required authorization(s)	<code>Configure Device Attributes</code>	<code>Device Security</code>	<code>secadmin</code>
		<code>Host Alternate Pathing</code>	<code>secadmin</code>
		<code>All Authorizations</code>	Not assigned by default

Note - If none of the default profiles are appropriate for the account being reconfigured, the Security Administrator role can create a new profile that includes the device allocation authorization(s), either by themselves or along with any other commands needed by the profile's users to perform the desired work (such as `allocate`, `deallocate` commands, and `tar`). How to create a new profile is described in "Adding or Modifying a Rights Profile" on page 144.

▼ To Prevent Automatic Display of File Manager After Device Allocation

1. **Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.**
See "To Log In and Assume an Administrative Role" on page 32, if needed.
2. **Use the `Admin Editor` action to open the file `rmmount.conf` for editing.**
See "To Use the Admin Editor Action to Edit a File" on page 46, if needed.
3. **Comment out the action for notifying the `File Manager` for the CD-ROM or floppy or both.**

The example shows the `action_filemgr.so` commented out for both the `cdrom` and `floppy` devices.

```
# action cdrom action_filemgr.so
# action floppy action_filemgr.so
```

▼ To Change or Add a Device Clean Script

1. **Write the script so that all usable data is purged from the physical device and that it returns 0 for success.**
2. **For devices with removable media, have the script attempt to eject the media if the user does not do so and put the device into the allocate error state if the media is not ejected.**
3. **Put the script at ADMIN_LOW into `/etc/security/lib`.**
4. **Use the Device Allocation Manager to specify the new script for a device.**
 - a. **As admin in an ADMIN_LOW workspace, bring up the Device Allocation Manager.**

Either select the Allocate Device option from the Trusted Path menu or launch the Device Allocation Manager action from the Trusted Desktop subpanel in the Front Panel.
 - b. **Click the Device Administration button to display the Device Allocation: Administration dialog box.**
 - c. **Highlight the name of the device in the Devices list to which you want to assign the new script.**
 - d. **Click the Configure button to display the Device Allocation: Configuration dialog box.**
 - e. **Change the name of the `device_clean(1M)` script as desired by editing the name in the text entry field to the right of Clean Program.**
 - f. **Click the OK button on the Configuration dialog box to save the label changes, click the OK button on the Administration dialog box to close it, and then close the Device Allocation Manager.**

Adding Software

This chapter covers these main topics:

- “Types of Software” on page 302
- “System Shell ” on page 303
- “The boot and inetd Profiles” on page 304
- “Types of Privileges” on page 304
 - “Trusted Processes in the Window System” on page 304
- “Processes, Programs, and Their Privileges” on page 305
- “Assigning Privileges to Commands and Actions” on page 307
- “Making Shared Library Directories Trusted ” on page 309
- “Security Administrator Role’s Tasks in Adding Software” on page 310
- “Finding Which Privileges a Program Needs” on page 315
- “Creating and Using Shell Scripts” on page 316
- “How Edited Program Files Are Prevented from Being Able to Use Inheritable Privileges” on page 320
- “Overview: Adding Boot Commands ” on page 320

This chapter provides these procedures:

- “To Create a New Administrative Action for Editing an Administrative File” on page 323
- “To Add Actions Outside of the System_Admin Folder” on page 325
- “To Mount a CD-ROM for Adding a Package” on page 325
- “To Find Out Which Privileges an Application Needs” on page 326
- “To Give Forced Privileges to a Command” on page 329

- “To Find Which Library Directories Are Used by an Application” on page 330
- “To Make a Library Directory Trusted” on page 332
- “To Write a Profile Shell Script that Runs Privileged Commands ” on page 333
- “To Write a Standard Shell Script that Runs Privileged Commands When Executed in a Profile Shell” on page 335
- “To Add Commands to /etc/inittab” on page 336
- “To Start RC Scripts with Security Attributes During Boot” on page 336
- “To Add Services to /etc/inet/inetd.conf” on page 338
- “To Save and Restore Privileges When Editing a File” on page 339

Types of Software

The following types of software can be added:

- Sun unbundled products and third-party applications that neither understand nor enforce Trusted Solaris security policy
- New programs, created using Trusted Solaris programming interfaces, that understand labels and MAC (mandatory access control) and that work within Trusted Solaris security policy
- New actions (created or approved by the Security Administrator role)
- Shell scripts (created or approved by the Security Administrator role)
- Additions to or modifications to commands that run during boot in run control scripts

This chapter summarizes how the creation and use of commands, actions, and scripts is different in the Trusted Solaris system than it is in the base Solaris system. This chapter reviews how privileges are used by commands and actions and how to do the following:

- Bring in new software
- If the software needs privileges in order to run, find out what privileges the software needs
- Decide whether giving any needed privileges to the new software is consistent with the site’s security policy, and
- Make privileges available to the software
- Make privileges available to commands in run control scripts

See the *Trusted Solaris Developer’s Guide* for how programmers can manipulate privileges.

As configured in the default system, the Security Administrator role can do the following:

- Import and export software at multiple labels
- Install software programs and CDE actions at `ADMIN_LOW` in the public directories (such as `/etc` and `/etc/dt/appconfig`) that allow use of the programs or actions by multiple users at all labels.
- Assign privileges to program files.
- Use the Profile Manager to assign privileges that are in effect when a command or action is executed in a trusted process from which the executable can inherit privileges.

Note - Because applications and shell scripts, whether they are externally or internally obtained, are added to a site's rights profiles as commands, the term *command* is used frequently in this chapter when referring to applications, site-developed executable programs, and shell scripts.

See “Types of Privileges” on page 304 and the following sections, which define what it means for a program file to have privileges and for a command or action to inherit privileges.

The Security Administrator role, in most cases, does not need to be consulted before programs or scripts are imported, created or used if the software meets the following criteria:

- Does not need to run with privilege
- Does not need to run with an effective UID or GID that differs from the real UID or GID of the invoking user
- Does not need to run at multiple labels
- Does not need to be added to a public directory

The System Administrator role controls who can bring in software by granting or denying the device allocation authorization to individual users. An account with the device allocation authorization can import or export data at any single label within that user's clearance.

System Shell

The system shell, `sysh(1M)`, is used to allow the use of privileges by commands run from *run control* (`rc`) scripts. `sysh` allows any command to execute but consults profiles for any security attributes with which the command is to be run. A profile name can be specified in a system shell script. If no profile is specified, the `/bin/sysh` shell looks at the `boot` profile by default.

The boot and inetd Profiles

The `boot` and `inetd` profiles are local to each computer. They specify commands with any security attributes needed during boot. The `boot` profile specifies security attributes for commands in `/etc/init.d` and `/etc/inittab`. The `inetd` profile specifies security attributes for commands in `/etc/inetd.conf`.

Types of Privileges

The Security Administrator role makes privileges available to software in either of two ways:

- Assigning forced privileges to the executable file itself (for commands only)
- Assigning inheritable privileges to a command or action in a rights profile.

Assigning privileges in a rights profile makes the privileges inheritable by a process as follows:

- When a command is executed in one of the shells that understands profiles (either the profile shells described in the `pfexec(1)` man page or the system shell, as described on the `sysh(1M)` man page and in “System Shell ” on page 303.
- By an action when it is launched by a trusted process in the window system.

See “Assigning Privileges to Commands and Actions” on page 307 for more details.

Trusted Processes in the Window System

The window system’s trusted processes are:

- The Front Panel
- Subpanels of the Front Panel
- The Workspace Menu
- The File Manager and
- The Application Manager

The window system’s trusted processes in the previous list are available to everyone, but accounts can access from the window system only the actions that are specified

in the account's rights profiles. For example, the administrative actions that are in the `System_Admin` folder can only be used if they are in one of the account's profiles. Therefore by default, since the `Edit Encodings` action is in the `Object Label` profile assigned to the Security Administrator role and the `Set Mount Points` action is not, the Security Administrator role can use the `Edit Encodings` action but cannot use the `Set Mount Points` action.

In the `File Manager`, if an action is not in one of the account's profiles, the icon for the action is not visible. In the `Workspace Menu`, if an action is not in one of the account's profiles, the action is visible, but an error displays if the action is invoked.

The CDE window manager, `dtwm(1)` calls the `Xtsolusersession` script, which then works with the window manager to invoke actions launched from the window system. Just as the profile shell consults an account's profiles when the account attempts to invoke a command, `Xtsolusersession` also consults the account's profiles when the account attempts to launch an action. In either case, if the action is in the user's profiles, it is run with any security attributes specified for it in a profile for the account.

Processes, Programs, and Their Privileges

Inheritable privileges are important for Security Administrator roles because privilege inheritance is used by:

- The profile mechanism to pass privileges to commands invoked in the profile shell
- The system shell to pass privileges to commands invoked in the system shell
See “System Shell ” on page 303 for more about the system shell, if desired.
- Trusted processes in the window system to pass privileges to actions
See “Trusted Processes in the Window System” on page 304 for more about trusted processes in the window system, if desired.

When a process executes a new program, the process's new inheritable set equals the process's old inheritable set before the new program was executed: $I[\text{new}] = I[\text{old}]$. The result is that the inheritable privileges available for one program to pass to another program are not affected by the forced or allowed privileges on the currently executing program. Maintaining the inheritable set without reference to the program file's forced or allowed set has the following two effects:

- The benefit of setting $I[\text{new}] = I[\text{old}]$ without reference to allowed privileges is that privileges can be passed from a process executing a program that cannot use the privileges to one that can.

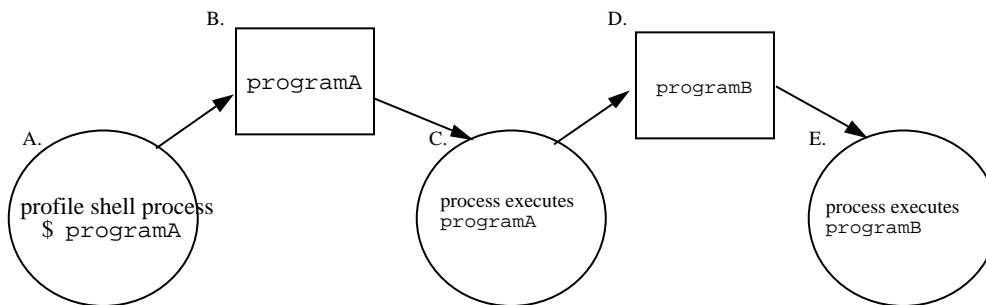
See “How Programs Without Privileges Can Pass Privileges to Other Programs” on page 306 for details.

- The benefit of setting `I[new]=I[old]` without reference to forced privileges is that forced privileges cannot be used by shell scripts.

See “When a Program File Has No Forced Privileges” on page 307 for details.

How Programs Without Privileges Can Pass Privileges to Other Programs

A process executing a program that has no allowed privileges cannot use any privileges because it cannot put any privileges into its effective set even if it inherits privileges from another trusted process. Such a process, however, can pass its inheritable privileges through to another program that it executes, one which might have allowed privileges and which therefore can use the inheritable privileges. The process executing the program without allowed privileges can pass privileges to another program because the inheritable set of the process is not affected by the lack of allowed privileges on the program. See the following figure.



- A.
In one of the invoking account's profiles, `programA` has `inheritable=10,12,19`, so `pfsh` sets its own `inheritable=10,12,19` and executes `programA`.
- B.
`programA` file's privilege sets: `forced=none`, `allowed=none`
- C.
process executing `programA` has `inheritable=10,12,19`; `effective=none`
- D.
`programB` file's privileges sets: `forced=none`; `allowed=10,12,19`
- E.
process executing `programB` has `inheritable=10,12,19`; `effective=10,12,19`

Figure 13-1 How a Program that Cannot Use Privileges Can Pass Them to a Program that Can

When a Program File Has No Forced Privileges

The inheritable set of a process cannot be increased by the forced privileges on the program. Any forced privileges on a shell script are not passed to commands invoked in a forced-privilege shell script. The result is that privileges cannot be used by shell scripts executed in standard UNIX shells, `sh(1)`, `csh(1)`, and `ksh(1)`. See the following figure.

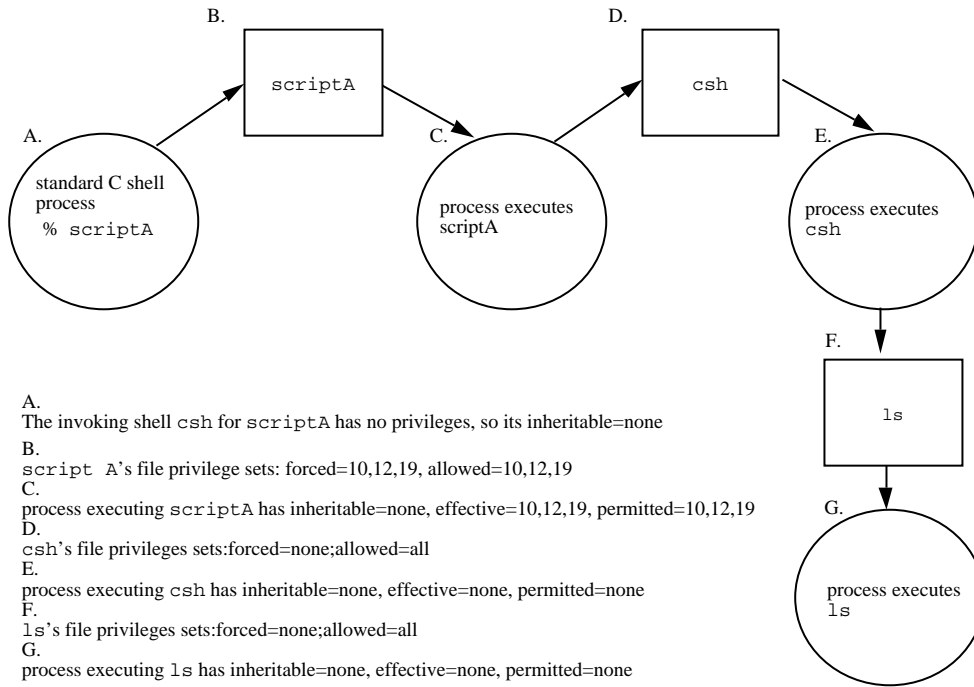


Figure 13-2 How Forced Privilege Shell Scripts Are Prevented from Passing Forced Privileges to Their Commands

Assigning Privileges to Commands and Actions

The default Trusted Solaris commands and actions have already been assessed as described in this chapter and have been assigned privileges if any privileges are required for them to do their work. After a site is configured, a privilege should be granted by a site's Security Administrator role only if the Security Administrator role is convinced that the command or action will use the privilege in a trustworthy manner.

The Security Administrator role makes privileges available by:

- Assigning forced privileges to the executable file itself (for commands only) or
- Making them inheritable by a command when it is invoked in the profile shell or when it is launched by an action when it is launched from a trusted process in the window system.

Giving Forced Privileges to an Executable File

The Security Administrator role can assign forced privileges to an executable file for a command by using the `File Manager Privileges` dialog box or by entering the `setfpriv(1)` command in a profile shell, as described under “To Give Forced Privileges to a Command” on page 329.

When a command with forced privileges is executed by any user in any shell, the forced privileges are put into the effective set of the executing program. The only way to prevent anyone from executing such a command with privilege would be by controlling access to the command itself—by giving an account a profile shell as its default shell and by not assigning the command or any other shell to any of that account’s profiles.

To change the privileges on an executable file, the process’s label must allow MAC write access to the file; DAC write permission is not required. The forced and allowed privilege sets of a file can only be changed either by the owner at the same label or lower (write-equal or write-up) or by a Security Administrator role (as configured in the default system) in an `ADMIN_LOW` workspace (write-up).

To give more detail, the forced and allowed privilege sets of a file can only be changed by:

- The owner of the file or
- A process with the `file_setpriv` privilege or
- An account with the `set file privileges` authorization

See also the `setfpriv(1)` man page.

Note - If you assign forced privileges using the `File Manager > Privileges` dialog box, it automatically assigns the same set of allowed privileges. However, the `setfpriv` command does not allow you to set any forced privileges unless they are in the file’s set of allowed privileges or unless you are setting the allowed and forced set appropriately in the same command line.

Giving Inheritable Privileges to a Command or Action

The Security Administrator role can specify inheritable privileges for a command or an action in an rights profile using the `SMC Rights` tool. The role can then assign the rights profile to a user using the `User Accounts` tool or to a role using the `Administrative Procedures` tool. Or the role can specify inheritable privileges for a program that is run by the system shell during boot. See Chapter 5 for how to use the `Rights`, `User Accounts`, and `Administrative Roles` tool.

Note - For privileges to be made available by inheritance, the privileges must be available in the command's allowed privilege set.

Making Shared Library Directories Trusted

Dynamically-shared libraries used by `setuid`, `setgid`, and privileged programs can be loaded only from trusted directories. When a privileged program cannot find its libraries, it fails with an error like the following:

```
ld.so.1: fatal: application-name: open failed: No such file or directory.  
Killed.
```

The Security Administrator role can add a privileged program's shared library directories to the list of trusted directories in `/var/ld/ld.config` by using the `crle(1)` command with the `-u` and `-s` options followed by a colon-separated list of pathnames to the library directories. The `-u` option adds the library directories specified with the `-s` option to any previously-specified trusted directories. Entering `crle` without options displays the current trusted directories. Any other use of `crle` without the `-u` option removes previously-specified entries.

To find out what libraries a program is using, anyone can use the `ldd(1)` command. See "To Find Which Library Directories Are Used by an Application" on page 330 for how the Security Administrator can check for the library directories used by the application and see "To Make a Library Directory Trusted" on page 332 for how to run `crle`.

The addition of a library directory to the list of trusted directories persists across reboots. However, the possibility exists that the `crle` command could be entered with other options but without the `-u` option (perhaps by a third party script), thereby removing the entries made on the command line.

To help ensure that all library directories needed for Trusted Solaris operation are always configured as trusted directories, the Security Administrator can create a boot-time script so that the `crle` command runs to add the desired library directories at every reboot. See “To Make a Library Directory Trusted” on page 332 for how to create such a script.

See `/etc/rc2.d/S90wbem` for an example of adding the JAVA library directories needed by the SMC to the trusted library directories list. See `/etc/init.d/README` and `/etc/rc2.d/README` for naming and numbering conventions for boot scripts.

Note - By default, boot scripts run with the system shell and with a real UID of 0 at `ADMIN_LOW`.

Also see the `ld(1)` man page for information on the link editor for object files.

Security Administrator Role’s Tasks in Adding Software

The default Trusted Solaris programs and actions have already been assigned privileges, effective UIDs or effective GIDs when any of these attributes are required for the programs or actions to do their work. This section discusses the issues and tasks associated with the adding of the following types of software:

- Sun unbundled products and third-party applications that neither understand nor enforce Trusted Solaris security policy
- New programs, created using Trusted Solaris programming interfaces, that understand labels and MAC (mandatory access control) and that work within Trusted Solaris security policy
- New actions (created or approved by the Security Administrator role)
- Shell scripts (created or approved by the Security Administrator role)

Some programs run at a single level with no privileges required, so the Security Administrator role can simply install them at `ADMIN_LOW` in a public directory and assign them as desired as commands in the rights profiles of users and roles without assigning privileges or modifying any other attributes to make the programs work. Other programs that need to bypass security policy may need to be assigned privileges, but before that is done, analysis and testing is required.

When Adding Existing Programs

Do the following when your site wishes to add any existing programs to a Trusted Solaris system, whether it is an application written outside of your organization, a Solaris unbundled software program, or a program written in house:

1. Find out if the application runs without changes in the Trusted Solaris system.

If it runs without privilege or any modification, you are done.

2. Find out why the program failed.

Some unbundled software packages and third-party applications written for Solaris cannot run because of certain modifications made to the Trusted Solaris operating environment to enforce security policy. For example, software that links with the kernel may be incompatible with Trusted Solaris modified kernel data structures. For similar reasons, loadable device drivers and other software may not be capable of operating in the environment unless changes are made to the code.

If the application is linked to the kernel or relies on aspects of the operating system that have been modified, the program probably is not capable of running on Trusted Solaris even if privileges are added unless other code changes are also made.

3. If the program does not rely on aspects of the Solaris operating environment that have been modified for Trusted Solaris, but it fails without privileges, find out what privileges or other attributes it needs.
4. If the program does require the use of privilege, assess whether the program will use its privileges in a trustworthy manner.

See “Things to Think About When a Program Fails Without Privileges” on page 312.

5. If the program can safely run with the privileges or other attributes it requires in a manner that does not violate the Trusted Solaris security policy or the security policy of your installation, you may then assign the required privilege(s) as described in “Assigning Privileges to Commands and Actions” on page 307.
6. If you have access to the source code of a program, you may add privileges in some cases, after a security consultant or programmer knowledgeable about security modifies the code.

These modifications might include privilege bracketing or adding code that makes the program aware of the Trusted Solaris security policy.

7. If you make privileges available to a program, you need to make sure that any libraries used by the program are identified as trusted. See “Making Shared Library Directories Trusted ” on page 309.
8. If the program cannot use its privileges in a trustworthy manner and it cannot be modified, do not make it available.

Things to Think About When a Program Fails Without Privileges

The most obvious type of program that fails without privileges under Trusted Solaris is one needs to run in Solaris as root (such as a program that executes with `setuid root`). This kind of program can be assigned an effective UID of root in a profile or, if it requires a real UID, the program can be assigned to the root role (at the Security Administrator's discretion).

Most applications are written in environments that do not have Trusted Solaris security mechanisms such as MAC. For this reason, the person who assesses the program needs to understand security and thoroughly understand what the new program is trying to accomplish.

While UNIX applications that need to violate DAC often are implemented to make careful checks before doing so on a user's behalf, a standard UNIX application certainly does not make similar checks about MAC. If you give such a program a MAC-override privilege, you may unintentionally provide a way for users to override MAC arbitrarily.

Some of the security considerations to be assessed are illustrated by the behavior of `rcp(1)`, which is a commonly used UNIX program. The `rcp` command, which copies files across a network, runs with `setuid root`. Running as root allows the program to run with all privileges in a standard UNIX system. Although the program is allowed to bypass DAC restrictions, it knows enough to check for DAC permissions on a file to make sure the user who executed the `rcp` command has permission to access the file. But `rcp` has no knowledge of MAC restrictions. If you gave it the `file_mac_read` or `file_mac_write` privilege, `rcp` would not do the right kinds of checks for MAC relationships when accessing a file for a user; so `rcp` would not be able to use the privileges you assigned it in a manner that enforces the security policy of the system.

If you simply assign a similar program the privileges it needs to run and do not modify it to work within the security policy of the Trusted Solaris, the program violates system security. In order to make it run without violating system security, you would need to add to the program's source code. For example, if a program needed to bypass MAC restrictions when reading and writing files, you would need to modify the source code by adding the necessary MAC checks.

Some software may need privileges for reasons that are not obvious and sometimes not necessary for the program to succeed. Even if it is not performing any function that seems to violate system security policy, an application may be doing something internally that does violate security, such as keeping track of its operations in a shared log file, or reading from `/dev/kmem` (see `mem(7D)`). Sometimes these internal policy overrides are not particularly important to the application's correct operation but merely provide a convenient feature for users. If your organization has access to the source code, check the possibility you can remove the operations that require policy overrides without affecting the performance of the application.

If the program would violate aspects of Trusted Solaris security policy, such as reading and writing files without doing MAC checks, then you should probably either make sure the required MAC checks are added to the source code, if you can, or not port the program.

When Applications Need to Run As Root

When an application has been written to run as root, the Security Administrator role has three options (all of which should be assessed for consistency with the site's security policy):

- If a *real* UID of root is not required, set up the application to run with an *effective* UID of root. Otherwise, set it up with a real UID of root.
- Find out what privileges the application needs and assign only the needed privileges, after determining that the application can use the privileges in a trustworthy manner.

See “To Find Out Which Privileges an Application Needs” on page 326.

When Adding a New Trusted Program

Even though a program's developer can manipulate privilege sets in the program's source code, if the Security Administrator role does not assign a privilege that the program needs, then the program cannot use the privilege. If it cannot override the security policy, the program may not do all the things you expect it to do, or it may not even run in the Trusted Solaris system.

Developer's Responsibilities

A developer who writes a program to be added to a Trusted Solaris system must do the following:

1. Understand whether the program requires privileges to do its work.
2. Know and follow techniques, such as privilege bracketing, for safely using privileges in programs
3. Be aware of the security implications when assigning privileges to a program and make sure that the program does not violate security policy.
4. Work with the Security Administrator role to place shared libraries linked to the program in a trusted directory as described in “Making Shared Library Directories Trusted ” on page 309 and use the trusted directory location when compiling the program in in “To Find Which Library Directories Are Used by an Application” on page 330.

See the *Trusted Solaris Developer's Guide* for additional instructions on how to use privileges in programs.

Security Administrator Role's Responsibilities

The Security Administrator role must ensure that a program that uses Trusted Solaris system calls and routines to work within security policy does not compromise the security of the Trusted Solaris system in any way.

1. Make sure the programmer and the program distribution process is trusted.
2. From one of these sources, find out which privileges are required by the program:
 - a. Ask the programmer.
 - b. Search the source code for any privileges that the program expects to use.
 - c. Use `runpd` as described in "To Find Out Which Privileges an Application Needs" on page 326.
3. Scrutinize the source code to make sure it behaves in a trustworthy manner when using the privileges it needs to operate.

When Adding Actions

The process of creating and using actions is pretty much the same in the Trusted Solaris system as it is in Solaris. Adding actions is described in the "Adding and Administering Applications" in *Solaris Common Desktop Environment: Advanced User's and System Administrator's Guide*

In Trusted Solaris, *use* of actions is controlled by the rights profile mechanism. Actions that are assigned security attributes in a rights profile can run with the assigned security attributes if they are invoked within one of the window system's trusted processes. In the Trusted Solaris system, a number of actions have been assigned security attributes in the rights profiles that are assigned to certain roles by default. The Security Administrator role can also use the Rights tool to assign security attributes to new actions.

Table 13-1 summarizes the main differences encountered in creating and using actions in the Trusted Solaris system.

TABLE 13-1 Differences in Creating and Using Actions Under Trusted Solaris Restraints

Base CDE	Trusted Solaris
<p>New actions may be created by anyone within the originator's home directory, and a new action is automatically usable by its creator.</p>	<p>An action is only usable by a user or role if the action is one of the account's rights profiles. The actions' search path has been changed so that actions in any individual's home directory are processed last instead of first. Therefore, no one can customize existing actions.</p> <p>If either the Create Action action or commands or actions that permit the editing of files are in an account's profile, the user or role <i>can</i> create a new action in the account's home directory, but the account may not be able to use the new action.</p> <p>There are two ways a user can use any new action: if the security administrator role adds the name of the new action to one of the account's rights profiles, or if the person has the All profile. The All profile turns off all checks for actions, and as a result any existing and potential actions may then be used by that account.</p> <p>If the account is allowed to use the action by its rights profiles, the account can launch the action from its home directory through the File Manager. The default system administrator and administrator roles are permitted to place actions in public directories.</p>
<p>Actions can be dragged and dropped to the Front Panel.</p>	<p>The Front Panel is part of the trusted path. The window manager recognizes only the administratively-added actions that are located in <code>/usr/dt</code> and <code>/etc/dt</code> subdirectories where system-wide action files are kept. Even if a normal user account or a non-administrative role account creates a new action in the account's home directory and has the All Accounts profile, new actions dragged to the Front Panel from the user's home directory cannot be recognized by the window manager, which only looks in the public directories.</p>
<p>The only way that actions can do privileged operations is if they are run by root.</p>	<p>If actions are specified to have privileges in one of the invoking account's rights profiles, actions can inherit privileges when they are launched from a trusted process. Therefore, the only way that actions can do privileged operation is if they have been assigned privileges in the account's profiles.</p>

Finding Which Privileges a Program Needs

The `runpd(1M)` command executes a command and logs its use of privilege.

The `runpd` command requires the trusted path attribute, which a command can only obtain when the command is listed in a profile and executed by an administrative role. Therefore, `runpd` cannot be run by a normal user to find out what privileges are required by a normal user. The root role should not be used to run `runpd` because the UID 0 may give the command more access than it would have with another UID. The `runpd` command can be used to test the command's use of privilege at the label at which the command is to be used. Running `runpd` in any administrative role except root logs the privileges needed by any normal user if the command is run at a label within the user accreditation range.

By default, the Security Administrator role is the only role with the `runpd` command in its profiles. Privilege debugging is enabled as described in "To Find Out Which Privileges an Application Needs" on page 326. The procedure describes the optional creation of an administrative role exclusively for doing privilege debugging.

The assignment of privileges a command might need should not be done automatically. It should be done after considering possible alternatives.

If a program needs to override DAC or MAC restrictions on accessing a file, the Security Administrator role might decide to assign an effective UID or GID to make the privilege unnecessary, as described in "Adding or Modifying a Rights Profile" on page 144.

When software has been assigned privileges or an alternate UID or GID, the software becomes *trusted* by virtue of the fact that it is being allowed to bypass aspects of the Trusted Solaris security policy. Be aware that you can make software trusted even though it might not be worthy of trust. . The Security Administrator role should not give any privileges to software until convinced that the software can use the privileges in a trustworthy manner. Only when it has been scrutinized and found to be using its privileges within the system security policy, can a program be called a trustworthy program.

Creating and Using Shell Scripts

If an account has been assigned a normal UNIX shell (`sh`, `csh`, `ksh`), the account can create new shell scripts that can run any command in the system without privileges. Therefore, if none of its commands need privileges, a shell script can be used by anyone who has access to the software and who is able to run a shell used to interpret the shell script.

Making privileges available to commands run in shell scripts may be done only by the Security Administrator role. Here is a review of the Trusted Solaris constraints that affect how a shell script can be made to run with or without privileges.

Remember that the two ways any command can run with privilege are:

- The command's executable file has the needed privileges in both its forced and allowed sets, or
- The command has the needed privileges specified in a profile assigned to the person or role that invokes the command, the program file for the command has the needed privileges in its allowed set, and the command is being run in a profile shell from which it can inherit privilege

Forced privilege commands are able to run with privilege in any shell because the forced privileges attached to the program file are available to the executing command even though the shell itself does not have any privileges. Assigning forced privileges to a `csh`, `sh`, or `ksh` shell script does not give any privileges to the commands executed by the shell script; even though a shell started from the script runs with the forced privileges, the shell does not have any privileges in its inheritable set. See the rules for how processes get privileges, which are described in "Processes, Programs, and Their Privileges" on page 305.



Caution - To prevent unauthorized tampering with object code or system scripts, whenever any executable program file is edited, any forced and allowed privileges previously given to that file are deleted. If a program file's allowed set is empty, it cannot use inheritable privileges, which are masked by the allowed set. However, the forced and allowed privileges for a shell script are not consulted by the profile mechanism when making inheritable privileges available. For this reason, shell scripts are more vulnerable than programs are to being modified without detection. Before making shell scripts available that use inheritable privileges, the Security Administrator role should keep in mind that the same protection against tampering that is available for programs is not available to shell scripts.

Summary of Shell Script Behavior in Trusted Solaris Systems

- If none of its commands need privileges, a shell script using any shell can be created by anyone who is allowed to use a text editor.
- A shell script can be used by anyone who has access to the software and who is able to run the shell that interprets the shell script.
- Forced privilege shell scripts do not pass privileges to commands that they contain.
- Allowed privileges on shell scripts have no effect on which privileges the programs executed by the shell script can use.

The allowed privilege set of the invoked shell's file is checked rather than that of the script's file.

- A standard shell script that is invoked in a profile shell can pass privileges to commands that it runs if the Security Administrator role lists the shell script with any privileges required by its commands in one of the invoking account's profiles.

The shell script can pass any of its privileges to be inherited by the commands it executes if the commands themselves have the allowed privileges they need on their program files. Because the commands are being run in a standard shell, it is no use to list them with privileges in one of the invoking account's profiles—because standard shells do not consult the profiles database. See the following figure.

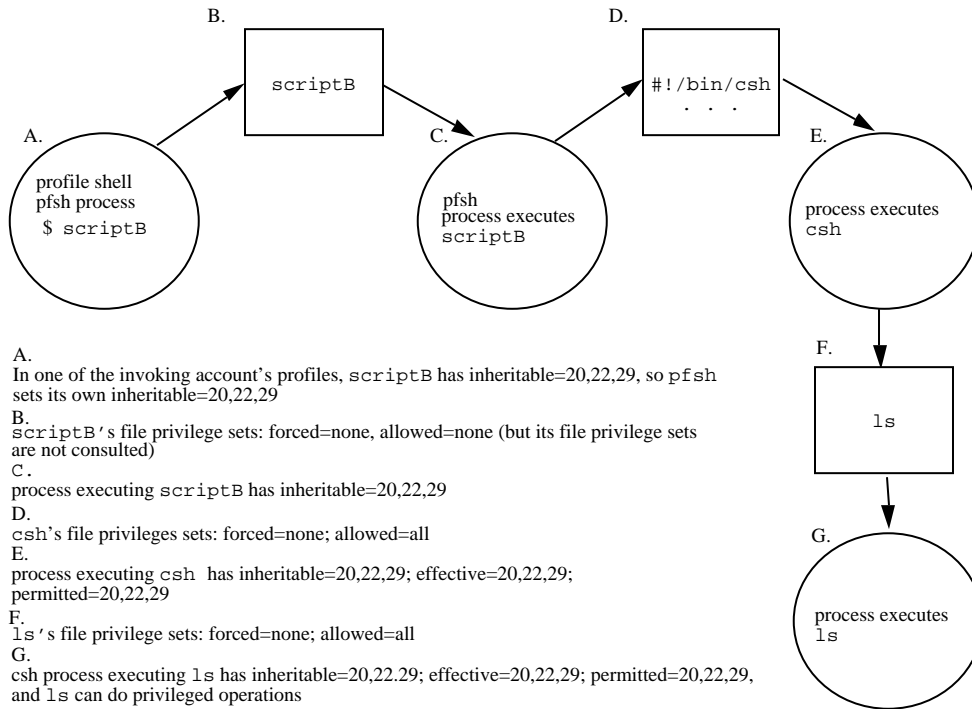


Figure 13-3 How Normal Shell Scripts Invoked in `pfsch` Can Pass Inheritable Privileges to Their Commands

- Shell scripts that use the profile shell (that is script that begin with the line `#!/bin/pfsch`) can pass privileges to their commands in whatever shell they are running, if the *commands* are listed with the required privileges in one of the invoking user's profiles.

More about Shell Scripts that Invoke the Profile Shell

Shell scripts that begin with the line `#!/bin/pfsh` behave differently when invoked by normal users than they do for administrative roles.

Normal User Behavior

- A shell script that invokes the profile shell can be executed by normal users on the command line in any shell.
- If the user has All Commands in a profile, the name of the profile-shell script does not need to be explicitly added to any of the user's profiles.
- Any commands in the profile-shell script have to be in one of the user's profiles or the user needs the All Commands profile. Commands that need privilege would have to be listed with the privileges they need in the profile.

Difference for Administrative Roles

- A profile shell script (using `#!/bin/pfsh`) must always be run on the command line of a profile shell.
- Roles cannot execute the profile shell from the command line or from a shell script (or bring up a GUI) without the trusted path. Normal users can execute a profile shell from the command line or from a profile shell script without the trusted path.
- A role must have the name of any script using `#!/bin/pfsh` *explicitly* listed in the Custom *role_name* Profile or another profile for the trusted path to be available. (For ease in troubleshooting, we recommend changing only the Custom *role_name* Profile .)

Note - Even though all roles now have the All commands profile, unlike a normal user with the All Commands profile, a role would need the script explicitly listed.

- As is true for normal users, any commands in the profile-shell script also need to be in one of the user's profiles.

See "To Write a Profile Shell Script that Runs Privileged Commands " on page 333.

How Edited Program Files Are Prevented from Being Able to Use Inheritable Privileges

To prevent unauthorized tampering with object code, any forced and allowed privileges previously given to a file are deleted whenever any executable program file is edited. This prevents someone from editing a file so that it uses privileges in a manner that was not originally intended. The Security Administrator role can save the list of privileges on such a file before editing it and restore them afterwards, as described in “To Save and Restore Privileges When Editing a File” on page 339.

Overview: Adding Boot Commands

New commands can be added to run during boot in the following ways:

- In the `/etc/inittab` file
 - See the `inittab(4)` man page, and “Adding New Commands to the inittab File” on page 321
- In scripts in the `/etc/init.d` (linked to `/etc/rc?.d`) directory
 - See the `init.d(4)` man page, and “Adding New Commands to `/etc/init.d` Scripts” on page 321. The base behavior is described in “Run Control Scripts” in the *System Administration Guide, Volume 1*.
- In the `/etc/inet/inetd.conf` file
 - See the `inetd.conf(4)` man page and “Adding New Services to `/etc/inet.d`” on page 322.

Unless other attributes are explicitly configured, commands run from the `inittab` or from scripts in `/etc/init.d` during the boot process have a label and clearance of `ADMIN_LOW`, a real and effective UID and GID of 0, and no privileges. If commands added into one of the files in the previous list need non-default security attributes, the commands need to be added to local profiles. Any profiles defined on a naming service master are not going to be available to the boot programs on a naming service client. The SMC Rights tool should be launched from a toolbox with the Files scope on the computer where the local files have been modified.

Adding New Commands to the `inittab` File

Make the changes to the See the `inittab(4)` file, and then add the commands to the boot profile. For example, if a script in `/usr/local/bin` named `mysite` needs to run with real UID of root, the Security Administrator role does the following:

- Uses the Admin Editor to add an entry for the `mysite` script to `/etc/inittab`:

```
lo:234:respawn:/usr/local/bin/mysite
```

- Uses the Rights tool with the Files scope to add the script `/usr/local/bin/mysite` to the boot rights profile with a real UID of 0.

See “To Add Commands to `/etc/inittab`” on page 336.

Adding New Commands to `/etc/init.d` Scripts

In the default Trusted Solaris system, the `/etc/init.d` scripts have been modified to use the system shell, `sysh(1M)`, instead of the Bourne shell, `sh(1)` when the service being started requires explicit privileges or other non-default security attributes that are defined in the boot profile. . In the default boot scripts, `/bin/sysh` is used without the name of a profile argument because if no profile is specified, the system shell looks at the boot profile by default.



Warning - Do not modify the commands already specified in the boot profile or modify the default `/etc/init.d` scripts. You can either add new scripts or change only scripts that may be added when a new application imported to the system.

When additional commands need to run during boot with non-default security attributes, the Security Administrator role specifies the commands with the needed attributes either by creating a new boot-time rights profile or by modifying the existing `boot` profile using the SMC Rights tool.

The role also needs do one of the following in `/etc/init.d`: modify an existing shell script, or create a new shell script so that the script starts with `#!/sbin/sysh` as the first line.

See the README in the `/etc/init.d` directory and in each `/etc/rcn.d` directory for guidelines about the numbering of the scripts that start system services.

As shown in the following example, a system shell boot script has `#!/sbin/sysh` as the first line. If the Security Administrator role has added the needed commands into the boot profile, there is no need to specify a profile name. If the Security Administrator role has created a new boot profile, the second line has the `setprof` argument followed by the name of the `local_boot_profile`.

```
#!/sbin/sysh
setprof local_boot_profile
```

For example, if a command needs a process label other than `ADMIN_LOW`, the profile needs to specify the label and if the command needs a UID of root, the profile needs to specify the required UID. See “To Start RC Scripts with Security Attributes During Boot” on page 336

Stopping or starting boot scripts in a Trusted Solaris system requires privileges, so the script must be executed by the System Administrator Role in an administrative role workspace with the trusted path attribute, and the script’s name must be in one of the account’s rights profiles.

The toolbox from which the Rights tool is invoked should be running with the local Files scope on the computer where the script is added to the `/etc/init.d`.

Adding New Services to `/etc/inet.d`

Services started by `inetd(1M)` run with the label and clearance of the client. `inetd` also runs with other attributes of the client if the following are specified in `inetd.conf(4)`:

- If the `uid` field has the keyword `CLIENT`, the services start with the client’s UID, GID, primary and any secondary groups.
- If the `wait-status` field contains the `setaudit` flag, the services are started with the client’s audit characteristics.

In addition:

- If the `wait-status` field contains the `trusted` flag, the trusted path attribute is available to the service.
- The Security Administrator can specify privileges and a label range by adding the service to the `inetd` rights profile and assigning the service the desired privileges and a label range.

If an entry in the `inetd` profile assigns privileges to the service, the service inherits the specified privileges.

If an entry in the `inetd` profile specifies minimum and maximum labels, `inetd` verifies that the label of the client is within the specified label range. If the label of the client is not the label range, the service is not executed.

See “To Add Services to `/etc/inet/inetd.conf`” on page 338.

Procedures for Adding Software

▼ To Create a New Administrative Action for Editing an Administrative File

1. **Launch the Admin Editor action to open the `/usr/dt/appconfig/types/C/TSOLadmin.dt` file for editing.**

See “To Log In and Assume an Administrative Role” on page 32, and “To Use the Admin Editor Action to Edit a File” on page 46, if needed.

2. **Copy and paste the definition for one of the existing actions in the `TSOLadmin.dt` file.**

The example in this procedure modifies a copy of the `Vfstab` action.

```
ACTION Vfstab
{
    LABEL          Set Mount Points
    ICON           Dtpenpd
    TYPE           COMMAND
    WINDOW_TYPE    NO_STDIO
    EXEC_STRING    /usr/dt/bin/trusted_edit /etc/vfstab
    DESCRIPTION    Specify the file system mount points
}
```

3. **Modify the copied action’s definitions.**

- a. **Change the ACTION name.**

This example creates a new action to edit the `system(4)` file to modify Trusted Solaris kernel switch settings.

```
ACTION EditSwitches
{
```

b. Change the LABEL.

```
LABEL          Set TSOL Switches
```

c. Change the ICON, if you have created a new icon or want to use another existing one from /usr/dt/appconfig/icons/C.

```
ICON          Dtpenpd
```

d. Change the file name in the EXEC_STRING.

```
EXEC_STRING    /usr/dt/bin/trusted_edit /etc/system
```

e. Change the text in the DESCRIPTION.

```
DESCRIPTION    Modify TSOL-related kernel switches
}
```

4. Save and close the TSOLadmin.dt file.

```
:wq
```

5. Copy and rename the Vfstab action file.

a. Go to /usr/dt/appconfig/appmanager/C/System_Admin.

b. Clone the Vfstab file and rename it to the name of the new action.
For example, rename Vfstab to EditSwitches.

c. Make the action file executable.

Select the `Permissions` option on the File Manager's `File` menu and set the permissions to executable for owner, group, and other, or enter the following on the command line:

```
$ chmod 777 EditSwitches
```

6. To make the action available to an administrative role on all hosts in the distributed system, copy the modified `TSOLadmin.dt` and action files to the NIS+ master and to all hosts in the distributed system, and bring up the Profile Manager, choosing NIS+ as the naming service.

Since the actions are not administered through NIS+, some other means of distribution must be used, such as `rdist(1)` or `sneakernet` (copying the files to a tape or floppy and carrying it around to install the files on each host).

7. To make the action available only on one host, bring up the Profile Manager, choosing None as the naming service.

8. Choose a profile, either the System Security profile or the System Management profile and choose Actions from the View menu.

The new action should be listed in the `System_Admin` application group in the Excluded list of actions.

- a. If the action edits a security-relevant file [such as the `system(4)` file] assign the action to the System Security profile.
- b. If the action edits an administrative file that would normally be modified by a UNIX system administrator and that does not contain labels or other security attributes [such as the `group(4)` file] assign the action to the System Management profile.

9. Assign to the new action the same privileges that are assigned to the `Set Mount Points` action: `file_dac_read`, `file_dac_write`, `proc_audit_appl`, `proc_audit_tcb`.

10. Log out and log in again.

▼ To Add Actions Outside of the System_Admin Folder

Adding actions can be done as described in the *Solaris Common Desktop Environment: Advanced User's and System Administrator's Guide*, within the limits of the Trusted Solaris MAC restrictions. Actions can be created either by using `CreateAction` or manually. The action should be labeled at `ADMIN_LOW` and placed in the `/etc/dt/appconfig/types/C` directory.

▼ To Mount a CD-ROM for Adding a Package

1. Assume the administrator role, and go to an `ADMIN_LOW` workspace.

See “To Log In and Assume an Administrative Role” on page 32 if needed.

2. Allocate the CD-ROM device.

Use the `Allocate Device` option from the `Trusted Path` menu or launch the `Device Allocation Manager` action from the `Trusted Desktop` subpanel in the `Front Panel`. The `Device Allocation Manager` dialog box displays.

3. Double-click the name of the CD-ROM device in the list of Available Devices to transfer it to the list of Allocated Devices.

The `Device Allocation: Select Label` dialog box displays.

4. In the Select Label dialog box, ensure that ADMIN_LOW is specified as the label and click the OK button.

A prompt then appears with the specified label, as shown below:

```
Insert disk labeled [C] in /dev/dsk/c0t6d0s0.
```

5. Insert the CD-ROM into the drive and click the OK button.

A dialog displays, as shown below:

```
Do you want cdrom_0 mounted?
```

6. Click the Yes button.

The `/cdrom` directory is created if it does not already exist, and the CD-ROM is mounted on it.

7. Press return when prompted to close the window.

▼ To Find Out Which Privileges an Application Needs

1. Assume the Security Administrator role and go to an ADMIN_LOWworkspace.

See “To Log In and Assume an Administrative Role” on page 32, if needed.

2. OPTIONAL: Create a new profile and assign it to an administrative role.

Note - The Security Administrator role is the only role assigned the `rumpd` command in the default configuration. The administrative role that can be created in this step could be used exclusively for privilege debugging. The role can be used to find out what privileges a command needs when run by a normal user in a label within the user accreditation range or by an administrative role within one of the administrative labels.

- a. **Use the SMC Rights tool to create a new rights profile (such as Analyze Program Privileges).**
Give the new profile `/usr/sbin/rumpd`, `/bin/getfpriv`, and `/bin/setfpriv` commands and the Shutdown the System, Enable Logins, and Set File Privilege authorizations, and the Admin Editor action, if desired, to allow the role to do the steps to enable privilege debugging before running `/usr/sbin/rumpd`.
- b. **Use the SMC Administrative Roles tool to create an administrative role (such as `prvdbg`).**
Assign the role the new profile along with the All profile, if desired.
- c. **Use the SMC Users tool to assign the privilege debugging role to an account.**

3. **Use the Admin Editor action to change the `tsol_privs_debug` setting to 1 in the `/etc/system` file.**

```
set tsol_privs_debug=1
```

4. **Use the Admin Editor action to remove the comment (#) at the beginning of the line that begins `kern.debug` in the `/etc/syslog.conf` file.**
The following line logs the privileges requested by system calls and daemons in the `/var/log/privdebug.log` file.

```
kern.debug;daemon.debug;local0.debug /var/log/privdebug.log
```

5. **Shutdown the machine using the Shut Down option on the Trusted Path (TP) menu and reboot.**

```
ok boot
```

6. Log in and assume the Security Administrator role or, if you created a new privilege debugging role in Step 2 on page 326, assume the new role.
7. In a terminal at the appropriate label, enter the `runpd` command followed by the name of the command and any options whose use of privilege you want to check.

Note - The label of the workspace should be that at which the command is typically run. While an administrative role might run the application at one of the administrative labels, a normal user would run the application at one of the labels in the user accreditation range.

As shown in the following example, `runpd` displays the name of the privilege(s) that the program needs in order to succeed followed by the type of access attempted (for example, `create`) followed by the name of the resource (for example, `RAW_SOCKET`).

```
$ runpd pathname_of_command_and_any_options
runpd: child terminated with a status of 0
process pathname_of_command pid process_ID lacking privilege privilege_name
to perform type_of_access upon resource resource_name (MM DD HH:MM)
```

The following example shows the result of running `runpd(1M)` on `ping(1M)` (for the purpose of the example `ping`'s forced privileges were removed).

```
$ runpd /usr/sbin/ping sif
sif is alive runpd: child terminated with a status of 0
process /usr/sbin/ping pid 5138 lacking privilege net_rawaccesssto create raw
socket (Oct 25 18:33) process /usr/sbin/ping pid 5138 lacking privilege sys_
net_config to manage transport opts (Oct 25 18:33)
```

8. Go to an `ADMIN_HIGH` workspace, and check the log file for the privilege debugging messages.

A typical privilege debugging log entry looks like the example shown in the following screen.


```
$ cat /var/log/privdebug.log
Mar 29 12:18:43 hostname unix: DEBUG: pathname_of_command pid
process_ID lacking privilege number to
number_of_type_of_access number_resource
```

The following screen shows the `privdebug.log` entries from when `runpd` was run on `ping`.

```
Oct 25 18:33:35 tribble unix: DEBUG: /usr/sbin/ping pid 5138 lacking privilege
36 to create raw socket
Oct 25 18:33:35 tribble unix: DEBUG: /usr/sbin/ping pid 5138 lacking privilege
68 to manage transport opts
```

The privilege numbers appear after the word “privilege.” You can look up the privilege number in the `/usr/include/sys/tsol/priv_names.h` file to find its name. For example, the privilege number 36 is associated with the name `net_rawaccess`. The numbers following the privilege number and the word “to” are the number of the type of access attempted followed by the number of the resource.

9. To assign the needed privileges see “To Give Forced Privileges to a Command” on page 329 or and “Adding or Modifying a Rights Profile” on page 144 for how to use the Rights tool to assign inheritable privileges.

Note - For a command to be able to use either forced or inheritable privileges, the privileges must be available in the command’s allowed privilege set.

10. Turn off privilege debugging: restore the changes you made to the `/etc/system` file and the `/etc/syslog.conf` file in Step 3 on page 327 and Step 4 on page 327 and reboot the machine.

▼ To Give Forced Privileges to a Command

1. As the file’s owner, as any user with the act as file owner authorization, or as the Security Administrator, go to the directory where the program file is located. Use the File Manager to navigate to the directory or use `cd(1)` on the command line. See the *Solaris User’s Guide* and the *Solaris Common Desktop Environment: User’s Guide*, if needed.
2. Make sure the file is executable.

Use the File Manager---Permissions dialog box to make sure that the Execute box is checked for Owner, Group and Other. See the *Trusted Solaris User's Guide*, if needed for more about the Permissions option on the File Manager. Alternately, if you have the `setfpriv(1)` command in a profile and are the owner of the file, or if you are in the default Security Administrator role, or if you have the `change file owner` authorization, you can use `setfpriv` on the command line to make the command executable by everyone.

3. **Make sure the command has allowed privileges equal to the forced privileges you plan to assign.**
 - a. **If you are using the File Manager Permissions dialog box, select the Allowed button, assign Allowed Privileges, and then select the Forced button, and assign the Forced Privileges.**
 - b. **If you have `setfpriv(1)` with the needed privileges in one of your profiles (as the Security Administrator role does in the default configuration), use `setfpriv` to assign the same privileges in both the allowed and forced sets.** The example shows the setting of `file_dac_read` and `file_dac_write` as allowed and forced privileges.

```
$ setfpriv -s -f file_dac_read,file_dac_write \  
-a file_dac_read,file_dac_write test.priv.file
```

▼ To Find Which Library Directories Are Used by an Application

1. **Assume the Security Administrator role and go to an `ADMIN_LOW` workspace.**
2. **Remove any forced privileges assigned to the program so that `ldd(1)` can get the needed information.**

If the program does not have forced privileges, then you do not need to do this step or Step 4 on page 331.

- a. **Check the command for forced privileges.**

The following example gets the list of privileges and saves them in a file. The file can be used to reset the privileges, if any, after the `ldd` command is run.

```
$ getfpriv -s -f program_name > filename
```

b. Remove the privileges from the program.

```
$ setfpriv -d -f privset program_name
```

The following example removes the privileges listed in *filename* from the program file.

```
$ setfpriv -d -f none program_name
```

3. Use the `ldd(1)` command to find out what library directories the application program is using.

```
$ ldd program_name
```

Note - The program that is using the shared libraries may be imbedded in a script that calls it, so make sure you are running `ldd` on the actual program that needs the libraries.

4. Add the privileges back to the program.

The following example uses `setfpriv` to set the privileges stored in the *filename* in Step 2 on page 330.

```
$ setfpriv -s -f `cat filename` program_pathname
```

5. Run the `crle(1)` command with the `-u` and `-s` options to add the library directories from step Step 3 on page 331 to the list of trusted library directories. See To Make a Library Directory Trusted for how to run `crle(1M)` .

▼ To Make a Library Directory Trusted

Note - This procedure assumes you have found out which shared libraries need to be trusted by following steps similar to those described in To Allow Trusted Programs to Link to Trusted Libraries for any privileged applications you have added since installation.

1. **Assume the Security Administrator role and go to an ADMIN_LOW workspace.**

2. **Use the SMC Rights tool to add the `crle(1)` command to the Custom Secadmin Profile with a real UID 0, and a default label and clearance of ADMIN_LOW**

3.

Use the `crle` command with the `-u` and `-s` options followed by a colon-separated list of pathnames to the library directories.

The following screen shows entering the `crle` command on the command line.

```
$ crle -u -s [directory_1[: . . . :directory_N]
```

4. **To regenerate the list of trusted directories at every reboot, add `crle` to a boot-time script.**

Use the Admin Editor to create or modify a script in the `/etc/init.d` directory.

If a privileged application already has a script, modify the existing script.

Otherwise, create a new script.



Caution - Do not modify any default Trusted Solaris scripts. Modify only scripts that are installed with new applications that need privileges.

Following is an example `crle` command line with the `-u` and `-s` options followed by a colon-separated list of library directories:

```
$ crle -u -s directory_1[: . . . :directory_N]
```

5. **Make a hard link from the script in the `/etc/init.d` directory.**

Use the `S` prefix in the target file's name for starting the script. Use the proper two-digit number in the target file's name to determine the order in which the script is executed during the run level. See the `README` in `/etc/init.d` and `/etc/rc2.d`.

```
$ cd /etc/rc2.d
$ ln /etc/init.d/scriptname $ NNscriptname
```

In the following example, the name of the new script in `/etc/init.d` is `new_script`, which is linked to `/etc/rc2.d/S87new_script`.

▼ To Write a Profile Shell Script that Runs Privileged Commands

Note - When adding a profile shell script that runs commands with inherited privilege, the Security Administrator role needs to use the Profile Manager to update an appropriate profile with a list of each of the commands that run within the shell script and to assign the commands any privileges they need. If a new shell script needs to be used by a role, all the commands that need security attributes must be added to the Custom `role_name` Profile or other profile that applies to the role, along with the name of the script itself.

Anyone with a text editor can write the shell script.

1. **Start the script with `/bin/pfsh` (instead of another shell) on the first line.**

```
#!/bin/pfsh
```

2. **Determine which commands need privileges and which privileges are needed.**

In the example, `/usr/lib/fs/nfs/nfsfind` is a cron job owned by root that needs privileges in order to run successfully at `ADMIN_HIGH`. The `tfnd` command needs the `file_dac_search` and `file_dac_read` privileges and the `rm` command needs the `file_dac_search`, `file_dac_write`, `file_dac_read`, and `file_mac_write` privileges. See “To Find Out Which Privileges an Application Needs” on page 326, if needed.

```
#!/bin/pfsh
# Copyright (c) 1993, 1997, 1998, 1999 by Sun Microsystems, Inc.
#ident "@(#)nfsfind.sh 1.5 97/05/21 SMI; TSOL 2.x''
#
# Check shared NFS filesystems for .nfs* files that
# are more than a week old.
#
```

(continued)

```

# These files are created by NFS clients when an open file
# is removed. To preserve some semblance of Unix semantics
# the client renames the file to a unique name so that the
# file appears to have been removed from the directory, but
# is still usable by the process that has the file open.
if [ ! -s /etc/dfs/sharetab ]; then exit ; fi
for dir in `awk '$3 == "nfs" {print $1}' /etc/dfs/sharetab`
do
    tfind $dir -M -name .nfs\* -mtime +7 -mount -exec rm -f {} \;
done

```

3. Assume the Security Administrator role and go to an ADMIN_LOW shell.

See “To Log In and Assume an Administrative Role” on page 32

4. Use the Profile Manager to update an appropriate profile to list the script, each of the commands that need to run within the shell script and to assign the commands the privileges they need.

See “To Launch the Solaris Management Console” on page 39, if needed.

To continue with the example, to enable the System Administrator role to run the example cron script with the needed privileges, the Security Administrator uses the Profile tool to update the Custom Admin Profile and makes sure it is assigned to the System Administrator role. The profile is modified to include the `/usr/lib/fs/nfs/nfsfind` script, the `tfind` command with the `file_dac_search` and `file_dac_read` privileges and the `rm` command with the `file_dac_search`, `file_dac_write`, `file_dac_read`, and `file_mac_write` privileges.



Caution - When you add commands to a profile and give them privileges or other security attributes, the commands execute with those attributes, not only in the profile shell script but whenever they are invoked in any profile shell, as long as the profile is in effect for the invoking account. The order of profiles is also important: the profile shell executes a command or action with whatever security attributes are specified in the first profile in the account’s list of profiles. For example, if `tfind` is in the Custom Root Profile with privileges, and the Custom Root Profile is the first profile in which `tfind` is found, then `tfind` will inherit the privileges specified in the Custom Root Profile when the root role executes `tfind` on the command line in a profile shell.

▼ To Write a Standard Shell Script that Runs Privileged Commands When Executed in a Profile Shell

Note - You can create a standard shell script to run its commands with privileges by adding the script to a profile and specifying the script to run with all the privileges needed by any of the script's commands. The script then inherits privileges when invoked in a profile shell, when an account has a profile containing the script.

1. **shell scripts:writing privileged using standard shells**Start the script with any standard shell (not `/bin/pfsh`) on the first line.

```
#!/bin/csh
```

Anyone with a text editor can write the shell script.

2. **Determine which commands need privileges and which privileges are needed.**
See “To Find Out Which Privileges an Application Needs” on page 326, if needed. The example, called `autosetpriv`, would allow the Security Administrator to assign a defined set of forced and allowed privileges to a file called `executable`. The `setfpriv` command in this script needs the `file_setpriv` privilege.

Note - This shell script is just an example. A normal shell script accepts the privileges and the filename as arguments and does error checking. Do not use this shell script unless you want to assign the named privileges to an executable file called `executable`, which will have those forced and allowed privileges available no matter who executes it.

```
#!/bin/csh
setfpriv -s -f ipc_mac_write,ipc_upgrade_il,proc_setsl,sys_trans_label
-a ipc_mac_write,ipc_upgrade_il,proc_setsl,sys_trans_label executable
```

3. **Assume the Security Administrator role and go to an ADMIN_LOW shell.**
See “To Log In and Assume an Administrative Role” on page 32
4. **Use the Profile Manager to update an appropriate profile to list the script, each of the commands that need to run within the shell script and to assign the commands the privileges they need.**
See “Adding or Modifying a Rights Profile” on page 144, if needed.

To enable the script called `autosetpriv` to run with the `file_setpriv` privilege needed by the `setfpriv` command, the Security Administrator role would use the Profile Manager to update the Custom Secadmin Profile (which is assigned to the Security Administrator role by default) to include the `autosetpriv` script and assign to `autosetpriv` the `file_setpriv` privileges.

5. **Test, debug, and execute the shell script as desired in the profile shell.**

```
$ autosetpriv
```

▼ To Add Commands to `/etc/inittab`

- Assume the security administrator role and use the Admin Editor action at `ADMIN_LOW` to edit `/etc/inittab`.

The following example adds `/usr/local/bin/myscript` to the file.

```
lo:234:respawn:/usr/local/bin/mysite
```

See the `inittab(4)` man page.

- Save and quit the file.

```
:wq
```

- Use the Rights tool from an SMC toolbox with the Files scope to add the script `/usr/local/bin/mysite` to the boot rights profile with a real UID of 0.

▼ To Start RC Scripts with Security Attributes During Boot

Note - You can change the default `boot` profile to add commands that need to start with security attributes during boot. Or you can create a new profile and use the `setprof` command to refer to the new profile in a new `sysh(1M)` script, as described in this procedure.

1. **Assume the security administrator role and use the Admin Editor action at `ADMIN_LOW` to create a new `sysh` script in `/etc/init.d`.**

See the `sysh(1M)` man page and “System Shell ” on page 303. The first line of the script should read as follows:


```
#!/bin/sysh
```

2. On the second line of the script, type in the `setprof` option to identify the name of a rights profile.

```
setprof new_profile_name
```

3. Save and quit the file.

```
:wq
```

4. In the `/etc/init.d` directory make a hard link from the new script to the desired `/etc/rcn.d` directories.

In the following example, the name of the new script in `/etc/init.d` is `new_script`, which is linked to `/etc/rc2.d/S89new_script` and `/etc/rc2.d/K8new_script`.

```
$ pwd
/etc/init.d
$ ln new_script /etc/rc2.d/S89new_script
$ ln new_script /etc/rc2.d/K89new_script
```

- a. For each run level at which the command should be started or stopped, go to the appropriate `/etc/rcn.d` directory and create a hard link from a properly-named target file to the `/etc/init.d` directory.
- b. Use the proper prefix in the target file's name for either starting (S) or stopping (K).
- c. Use the proper numbers in the target file's name to help determine the order in which the script is executed during the run level.

```
$ cd /etc/rc2.d
$ ln /etc/init.d/scriptname [s|k]nscriptname
```

5. Use the SMC `Rights` tool to create a new rights profile or to modify the boot profile.
 - a. Choose a toolbox with the Files scope.
 - b. Use the `Rights` tool to create a new profile that lists the command and any desired security attributes.
6. Shut down the system using the Shut Down option from the TP Menu and reboot.

```
ok boot
```

▼ To Add Services to `/etc/inet/inetd.conf`

1. Assume the Security Administrator role and use the Admin Editor action in an ADMIN_LOW to open the `/etc/inet/inetd.conf` file for editing.

For example, the following line adds a service in `/usr/local/bin` named `newservice` with the CLIENT keyword in the UID field so that the service executes with the UID and GID(s) of the CLIENT. In the flags field, the `trusted` keyword causes the service to run with the trusted path attribute, and the `setaudit` keyword causes the service to run with the client's audit characteristics:

```
myport      stream  tcp6     nowait,trusted,setaudit CLIENT    /usr/local/bin/newservice
```

2. Save and quit the file.

```
:wq
```

3. If the service needs to run with privileges or a restricted label range, use the `Rights` tool to add the service to the `inetd` rights profile along with any desired security attributes.

▼ To Save and Restore Privileges When Editing a File

1. **Assume the Security Administrator role and use `getfpriv(1)` to list the privileges on the executable file and save the output.**

The following example directs the output into a temporary file.

```
$ getfpriv executable_file > tempfile
```

2. **After editing the executable file, use the File Manager to make the file executable again (if needed) and then restore the privileges listed in the temporary file.**

The following example uses `setfpriv` to set the privileges stored in the *tempfile* created in Step 1 on page 339.

```
$ setfpriv -s -f `cat tempfile` program_pathname
```


Index

A

- access
 - administrator responsibilities 52
- access policy
 - devices 273
- accounts
 - assigning clearances 132, 141
 - assigning home directories 126
 - assigning idle time 133
 - assigning labels 132
 - assigning passwords 127, 146, 148, 149
 - assigning profiles 130
 - assigning roles 130, 131
 - assigning shells 124
 - assigning users to roles 139
 - deletion precautions 54
 - managing 82, 110
 - planning 83, 84
 - procedures for setting up 102
 - security precautions 55
 - startup files 91, 110
- accreditation checks 194, 197
- actions
 - adding outside the System_Admin folder 325
 - administrative
 - adding new 323, 325
 - restricted by account profiles 305
 - Share file system 240
 - using actions 314
- add_allocatable(1M) command
 - described 289
- admin
 - adding a host 248
- ADMIN_LOW label
 - installing publicly available software 302
- administrative actions
 - creating 323, 325
 - in Solaris Management Console 43
 - introduced 305
- administrative roles
 - assigning passwords 138
 - assuming
 - background 30
 - procedure 32, 38
 - changing workspace SLs 41
 - described 112
 - password 30
 - switching workspaces 41
 - workspaces
 - display when role is assumed 30
 - using 47
- Administrative Roles tool
 - assigning passwords 138
 - assigning users to roles 139
 - Home Directories tab 140
- administrator role
 - saving and restoring NIS+ tables 249
- adminvi(1M) command
 - aliasing vi(1) 114
- allocate error state
 - caused by failure of eject(1) 242, 287
 - defined 285
 - procedure for correcting 293
- allocate(1MTSOL) command
 - described 289

- Application Manager
 - as trusted process 304
 - passing inheritable privileges 309
- applications
 - assigning forced privileges 329
- at command
 - running privileged commands 100, 102
- at(1) command
 - administrative differences 99
- at.allow file 101
- at.deny file 101
- atq command
 - administrative differences 102
- atq(1) command
 - administrative differences 99
- atrm command
 - administrative differences 102
- atrm(1) command
 - administrative differences 99
- attr_mac_policy 273
- audio coprocessor 287
- AUDIO_DRAIN ioctl
 - run by device_clean(1M) 287
- AUDIO_SETINFO ioctl
 - resetting device to default 287
- AUDIOGETREG ioctl
 - run by device_clean(1M) 287
- audit IDs
 - purpose when a role is assumed 30
 - purpose when role assumed 30
- auth_name file 70
- authorizations
 - adding 70
 - Allocate Device 272
 - device-related
 - procedure for assigning to an account 296
 - for device administration 276
 - Manage All Jobs 101
 - Manage Owned Jobs 101
 - procedure for adding 71
 - table of device-related 284

B

- banner pages
 - printing without 266
- batch (1)command

- administrative differences 99
- batch command
 - running privileged commands 100, 102
- boot rights profile 321

C

- cachefs filesystem type 228
- CD players
 - launching automatically 242
- CD-ROM devices
 - accessing 272
 - device_clean script 286
- CIPSO
 - use in packets 190
- clearances
 - assigning 132, 141
- commands
 - privileged
 - run by cron(1M) 100
 - privileges 316
 - trusted 316
- commercial applications
 - adding 311
- .copy_files file
 - using 97
- cron command
 - running privileged commands 100, 102
- cron(1M) command
 - administrative differences 99
- cron.allow file 101
- cron.deny file 101
- crontab command
 - administrative differences 102
- crontab(1M) command
 - administrative differences 99
- customizations
 - changing printer output 260
- cut and paste
 - c 61

D

- DAC
 - cautions about override privileges 316
 - policy for devices 273
- data_mac_policy 273

- Database Manager
 - using 43
- deallocate command
 - described 289
- default shells
 - assigning to accounts 124
- /dev/kmem kernel image file
 - security violation 312
- developers
 - responsibilities 313
- Device Administration button 276
- device allocation
 - authorization 272
- Device Allocation Manager
 - allocating and administering
 - devices 275, 298
- device policy
 - procedure to set 291
- device special files
 - access policy 273
- device_allocate(4) file
 - described 289
- device_clean(1M) command
 - described 289
- device_clean(1M) script
 - in procedure for adding devices 293
- device_clean(1M) scripts
 - for tape devices 286
 - review 286
- device_maps(4) file
 - described 289
- device_policy(4TSOL) file
 - described 274
- devices
 - access policy
 - defaults 273
 - defining or redefining 274
 - accessing 275
 - administering 273, 298
 - non-allocatable
 - setting the label range 273
 - policy
 - table of defaults 273
 - policy for new 274
 - setting policy
 - procedure 291
 - setting policy for 273
 - table of related authorizations 284

- dfstab file 240
- directories
 - changing flags 222
 - changing labels and privileges 234
 - making available for mounts 240
 - procedure for sharing with other
 - hosts 240
 - security attributes 220, 225
 - upgraded
 - privileges 218
- dminfo(1M) command
 - reporting entry in the device_maps 289
- dtssession(1) command
 - running updatehome(1M) 98
- dtterm(1) terminal
 - forcing the sourcing of .profile 95, 114
 - forcing the sourcing of .profile 108
- dtwm(1) command 305

E

- editing privileged executables 320
- email
 - managing 156, 175
 - options 165
 - switching mail tools 166, 175
 - troubleshooting 161, 165
- emetric
 - described 199
- /etc/cron.d/CRON
 - cron(1M) lock file 100
- /etc/default/login file
 - specifying RETRIES 56
- /etc/init.d directory
 - RMTMPFILES script 60
- /etc/init.d scripts
 - Trusted Solaris modifications 321
- /etc/nologin file
 - disabling logins 60
- /etc/skel directory 96
- exec system call
 - inheriting privileges across 305
- executable files
 - assigning forced privileges 329
 - editing while preserving privileges 320
- exporting directories 240
- exporting software 302

F

- failsafe session
 - recovering from startup file errors 94
- fallback mechanism
 - creating 211
- FDFS
 - mounting in Trusted Solaris 227
- File Manager
 - as trusted process 304
 - changing security attributes 220
 - passing inheritable privileges 309
 - Privileges dialog box 308
- file systems
 - action for sharing 240
 - cachefs type
 - mounting 228
 - changing security attributes using mount 238
 - changing security attributes using newsecfs(1M) command 236
 - changing security attributes using setfsattr(1M) command 237
 - changing security attributes using vfstab file 239
 - fdfs type
 - mounting 227
 - hsfs type
 - mounting 227
 - lofs type
 - mounting 227
 - managing 219, 244
 - nfs type
 - mounting 227
 - pcfs type
 - mounting 227
 - security attributes 222, 225
 - single label 225
 - table of supported types, examples, notes 227
 - tmpfs type
 - mounting 228
- file_mac_write privilege
 - resulting in a file's dominating its directory's SL 68
- file_upgrade_sl privilege
 - resulting in upgraded names 68
- files

- backing up 234
- changing flags 222
- changing labels 221
- changing privileges 221
- managing 219, 245
- procedure for changing labels and privileges 234
- restoring 234
- upgraded 218

- floppy disk devices
 - accessing 272
 - device_clean script 286
- fork system call
 - inheriting privileges across 305

- Front Panel
 - as trusted process 304
 - passing inheritable privileges 309
- Subpanels
 - as trusted processes 304

G

- getdents system call
 - restricting from returning upgraded names 68
- getfattrflag command
 - described 222
- getfpriv command
 - using to save privileges 339
- getfsattr command
 - described 224
- GIDs
 - effective
 - defaults 310
- groups
 - deletion precautions 54
 - security requirements 54

H

- hexadecimal label equivalents
 - determining 76
- Home dialog box
 - User Manager 126
- Home Directories tab
 - Administrative Roles tool 140
- Host Manager

- adding a host to a running system 248
- host types
 - networking 181
 - table of templates and protocols 181
- hosts
 - networking concepts 177
- HSFS
 - mounting in Trusted Solaris 227
 - procedure for mounting 242

I

- icons
 - visibility
 - in the File Manager 305
 - in the Workspace Menu 305
- identification and authentication
 - before assuming a role 30
- IL floating
 - policy for devices 273
- il_float_policy 273
- ILs
 - devices
 - il_float_policy 273
- inheritable privileges 305
- init.d directory
 - RMTMPFILES script 60
- initialization files
 - Trusted Solaris differences
 - shells 92
- internationalization
 - changing printer output 260
- IP Options
 - using for trusted routing 186

K

- kadb(1M) command
 - overriding default system(4) switch
 - setting 59
- kernel switches
 - configurable 66
- kmem(7D) kernel image file 312

L

- label ranges
 - setting on individual computers 273
- label-relation 65

- label_encodings file
 - procedures 268
 - printing without banners and trailers 266
 - printing without labeled pages 261, 268

- labels
 - changing on files and directories 234

- Labels dialog box
 - User Manager 132

- libt6(3NSL) library 100

- .link_files file
 - using 97

- links
 - symbolic
 - MAC and IL attributes 218

- list_devices(1M) command
 - described 289

- local.login file
 - defining printers 268

- LOFS
 - mounting in Trusted Solaris 227

- log files
 - security violation from sharing 312

- login
 - by administrative roles 30

- login sequence 60

- logins
 - maximum allowed number of failures 55
 - opening an account closed by too many failed logins 55
 - setting the maximum number of failures 56

M

- MAC
 - cautions about override privileges 312
 - incoming packets
 - packets 197
 - outgoing packets 195
 - policy for devices 273

- mail
 - managing 156, 177
 - options 165, 166
 - switching mail tools 166, 175
 - troubleshooting 161, 165

- .mailrc file 94
- man(1) command 97
- man pages
 - accessing for all bundled products 97
- Manage All Jobs authorization 101
- Manage Owned Jobs authorization 101
- MANPATH environment variable 97
- minimum labels
 - assigning 132, 141
- mldpwd(1) command 219
- MLDs
 - mounting 218
 - mounting on unlabeled hosts 219
 - privilege requirements 219
- mounts
 - managing 219, 244
 - procedure for TMPFS file systems
 - tmpfs type 242
 - troubleshooting 243

N

- naming service
 - choosing in Solaris Management Console 43
- network interfaces
 - configuring 212
 - requirements 193
- networking
 - concepts 177
- networks
 - default labeling 196
 - procedures for configuring 210
 - using templates 204
- NFS
 - mounting in Trusted Solaris 227
- NIS
 - maps
 - unique in Trusted Solaris 247
- NIS+
 - tables
 - unique in Trusted Solaris 247
- NIS+, managing 246, 248
- normal user
 - accessing devices 272
- nsswitch.conf(4TSOL) file
 - trusted network database entries file 206

O

- object reuse
 - requirements
 - clearing names of empty directories 218
- open_priv 273

P

- packets
 - IP options 187
 - IP options field 186
 - outgoing
 - MAC rules 195
 - security attributes 191
- passwords
 - assigning 127, 138, 146, 148, 149
 - role 30
 - storage 53
- PCFS
 - mounting in Trusted Solaris 227
- permissions
 - on devices 273
- Printer Administrator
 - launching 261
- printers
 - label ranges
 - setting 273
- printing
 - configuring labels and text 260
 - managing 254, 271
 - without banners and trailers 266
 - without page labels 268
 - without page labels, procedure 268
- priv_name(4) file 74
- priv_names.h file 73
- privilege debugging
 - setting tsol_privs_debug 68
- privileged commands
 - run by cron(1M) command 100
- privileged programs 316
- privileges
 - adding 72
 - allowed 309
 - changing on files and directories 234
 - forced
 - assigning 308, 329

- inheritable 305, 309
- making available to commands 307
- non-obvious reasons for requiring 312
- procedure for adding 74
- saving and restoring an edited executable 339
- PROCFS
 - mounting in Trusted Solaris 227
- Profile Manager
 - specifying privileges for commands and actions 309
- profile shell
 - startup algorithm 94
- profiles
 - assigning 130
- programs
 - commercial
 - assigning privileges to 304, 308
 - new, trusted
 - assigning privileges to 304, 308
 - trusted
 - defined 316
 - trustworthy
 - defined 316

R

- rcp command
 - required privilege 312
- real UID
 - root
 - required for applications 313
- reboot
 - effecting changes to device_policy(4) 274
- remove_allocatable(1M) command
 - described 289
- rights profiles
 - boot 321
 - controlling the use of actions 305
 - creating new for boot commands 321
- Rights tab
 - User Accounts tool 130
- RIPSO
 - use in packets 190
- roles
 - administrative 112
 - assigning home directories 140
 - assigning passwords 138

- assigning shells 137
- combining administrative roles 112
- creating 113
- managing 111
- password 30
- procedure for creating 116
- Roles dialog box
 - User Manager 131
- root role
 - using to install applications 313
- root UID
 - required for applications 313
- routers 198
- routing
 - assigning default 213
 - concepts 21, 197, 201
 - tables
 - defined 199
- run control scripts
 - shell use 303, 321
- runpd command
 - dependency on tsol_priv_debug setting 68

S

- /sbin/sysh shell
 - using during boot 321
- Security Administrator role
 - administering use of devices 283
- Security Administrators
 - accessing the Printer Administrator 261
- security administrators
 - modifying window configuration files 78
- security attributes
 - file systems 220, 225
- security features
 - identification and authentication
 - for roles 30
 - identification and authentication for roles 30
- security mechanisms
 - extendable 69
- security policy
 - allowing a wildcard in special boot files 207
 - training users 50, 51

- sel_config file
 - configuring selection transfer rules 61
 - sections 64
 - sel_mgr command 64
 - sendmail command
 - using 158, 160
 - setfattrflag command
 - described 222
 - setfpriv command 308
 - restricting assignment of forced and allowed privileges 308
 - setfsattr command
 - described 224
 - share(1M) command 240
 - shareall(1M) command 240
 - sharing directories 240
 - shell scripts
 - summary of Trusted Solaris behavior 317
 - user and role requirements 319
 - writing privileged 333
 - writing privileged using standard shells 335
 - shells
 - assigning to accounts 124, 137
 - profile
 - startup algorithm 94
 - sysh(1M) 321
 - skeleton directories
 - defining printers 268
 - use in Trusted Solaris 96
 - software
 - exporting
 - multiple SLs 302
 - importing
 - multiple SLs 302
 - installing publicly available software at ADMIN_LOW 302
 - porting
 - reasons against 313
 - Solaris Management Console action
 - using 43
 - startup files
 - configuring accounts 91, 110
 - procedures for customizing 107, 110
 - read at window system startup 92
 - RMTMPFILES 60
 - .mailrc file 94
 - str_type_type 273
 - symbolic links
 - MAC and IL attributes 218
 - sysh shell
 - using during boot 321
 - system security
 - violations 312
 - System_Admin folder
 - using administrative actions 305
- T**
- tape devices
 - accessing 272
 - device_clean scripts 286
 - tar
 - saving security attributes 289
 - TMPFS
 - mounting in Trusted Solaris 228
 - procedure for mounting 242
 - troubleshooting
 - mounts 243
 - trusted networking
 - databases
 - creating fallback entries 211
 - host types 181
 - trusted path attribute
 - when it is needed 112
 - Trusted Path menu 30
 - trusted processes
 - defined 304
 - launching actions 305
 - passing inheritable privileges 309
 - trusted programs 316
 - adding 313
 - trusted_edit script
 - assigning as default editor 115
 - trustworthy programs 316
 - tsol_hide_upgraded_names configurable
 - kernel switch
 - defined 68
 - tsol_privs_debug configurable kernel switch
 - defined 68
 - tsol_privs_debug switch
 - described 316
 - TSOLadmin.dt file
 - adding an administrative action 323
 - tunnel file

- procedure for creating 215
- setting up tunneling 207
- tunnelling
 - passing emetrics through non-TSOL hosts gateways 207
- two-person control
 - adding privileges to programs 313

U

UFS

- mounting in Trusted Solaris 228

UIDs

- effective
 - defaults 310
- effective UID of root 313

unbundled Sun applications

- adding 311

UNIX domain socket

- used by cron(1M) and its clients 100

unlabeled hosts

- mounting MLDs 219

.updatehome(1 command

- using 97

upgraded names

- defined 68

User Accounts tool

- Roles tab 130

User Manager

- assigning passwords 127, 146, 148, 149
- assigning profiles to accounts 309
- Home dialog box 127

- Labels dialog box 132

- opening an account closed by too many failed logins 55

Password dialog box

- opening a closed account 55

- Roles dialog box 131

- using 43

users

- assigning to roles 139

- security training 50, 55

- /usr/dt/appconfig/appmanager/C/System_Admin file

- adding an administrative action 324

- /usr/dt/appconfig/types/C/TSOLadmin.dt file

- adding an administrative action 323

W

- window manager 305

window system

- trusted processes 304

- passing inheritable privileges 309

Workspace Menu

- as trusted process 304

workspaces

- administrative

- using 47

X

- Xtsolusersession script 305