Trusted Solaris Administrator's Procedures

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

Part No: 805–8055
October 12 1998

Contents

Preface

This *Trusted Solaris Administrator's Procedures* manual provides procedures for managing users and machines 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™ 2.x administration, CDE, Solstice™, and NIS+**

The procedures described here are unique to Trusted Solaris administration. Administrators of Trusted Solaris operating environments must already understand how to work within and administer the Solaris 2.x operating environment, upon which the Trusted Solaris system is based, and understand how to use and administer the Common Desktop Environment (CDE) window system, Solstice™ AdminSuite™ system administration tools and NIS+.

Note - AnswerBooks for the above-mentioned products that are bundled into Trusted Solaris are available on the *Trusted Solaris 2.5.1 AnswerBook*.

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

Administrators should understand 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 has been assigned to the task in the default configuration. Reference tables in Appendix A list the responsibilities, the relevant actions and commands for each of the default roles. Each local security administrator is responsible for making all administrators aware of the new configuration if the default administrative roles have been reconfigured at your site.

How This Book Is Organized

This book has three parts and a total of 18 chapters.

Chapter 1

Reviews how to log in, enable logins, assume an administrative role, work in an administrative role workspace, launch administrative actions from the workspace and invoke administrative commands in the profile shell.

Chapter 2

Describes security processes for administrators to put into effect and provides procedures for how to distribute changed configuration files to remote hosts, change the maximum number of failed passwords, get hexadecimal equivalents for labels and clearances, extend extendable mechanisms, including adding authorizations and privileges, and administer multilevel directories.

Chapter 3

Describes how the responsibilities for managing user accounts are divided between two administrative roles, what decisions to make and what to do before setting up accounts for users and roles, and how to administer startup files and `at` and `cron` jobs.

Chapter 4

Describes the differences between user and role accounts, how the responsibilities for managing all roles are given to the security administrator, except for the responsibility of managing the security administrator's own role, which is given to the system administrator. Describes the Custom Role Profiles and how the User Manager is used in administering role accounts and introduces the profile manager that is used to configure the profiles for new roles or to modify the profiles for existing ones.

Chapter 5

Describes the information that must be provided in each of the dialog boxes of the User Manager, then provides step by step procedures for entering the required information in the fields.

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

Provides a checklist of all the tasks for setting up a new user account.

Chapter 8

Describes how to use the Profile Manager to create, modify and delete profiles. This chapter includes tables listing the actions, commands and command privileges assigned to each of the default role and user profiles.

Chapter 9

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

Chapter 10

Describes how to specify the security attributes and set up routing for trusted network communications between machines.

Chapter 11

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 12

Describes how NIS+ is used to centrally administer the Trusted Solaris distributed system and lists the new NIS+ tables.

Chapter 13

Describes how to modify the system configuration switches that set whether ILs are displayed, whether ILs float, whether ILs are reset when a new program is executed with execve(2TSOL), whether the names of upgraded files and directories are displayed, whether privilege debugging mode is turned on, and whether a host can be brought down to the monitor prompt by use of the Stop A keys. Also describes how to modify certain aspects of the window system.

Chapter 14

Describes how to configure printing, how to add and delete labeled and unlabeled printers, and provides pointers to the manual that describes how to specify handling caveats for printer banner and trailer pages.

Chapter 15

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 devices.

Chapter 16

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

Chapter 17

Provides a worksheet for collecting information for setting up a host.

Appendix A

Summarizes the contents of the default set of execution profiles.

Related Books

- *Trusted Solaris User's Guide*

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

- *Trusted Solaris Administration Overview*
- *Trusted Solaris Administrator's Procedures*

- *Trusted Solaris Installation and Configuration*
- *Trusted Solaris Label Administration*
- *Trusted Solaris Developer's Guide*
- *Trusted Solaris Reference Manual*
- *Trusted Solaris 2.5.1 Release Notes*
- *Trusted Solaris 2.5.1 Transition Guide*
- *Compartmented Mode Workstation Labeling: Encodings Format*

What Type Styles Mean in Text and Examples

Table P–1 explains the type styles used in this manual.

TABLE P–1 Typographic Conventions

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

TABLE P–1 Typographic Conventions *(continued)*

Which Trusted Solaris Prompts Go With Which User Types

Table P–2 identifies which prompts go with which shells.

TABLE P–2 Trusted Solaris Prompts

Shell	Prompt
C shell prompt	`hostname%`
root role prompt	#
Bourne shell, Korn shell, and Profile shell prompt	$

PART I

Procedures Common to All Tasks and Administrative Roles

This part of the *Trusted Solaris Administrator's Procedures* manual contains two chapters that describe the tasks and procedures common to all roles.

Chapter 1 includes these topics:

Chapter 2 includes these topics:

CHAPTER 1

Assuming a Role and Working in a Role Workspace

As described in the *Trusted Solaris Administration Overview*, administrative tasks in the Trusted Solaris system are performed by multiple administrative roles. This chapter describes the sequence for assuming a role and provides guidance on how to work in administrative role workspaces. This chapter covers the following major topics:

This chapter provides the procedures listed here:

Review of Administrative and Non-administrative Role Concepts

The following sections review some concepts introduced in the *Trusted Solaris User's Guide* for convenience, so you can see the complete process from the point of view of someone logging in to do administrative work. Trusted Solaris has two types of roles: *administrative roles* and *non-administrative roles*.

Administrative Roles

Most administrative work is done administrative roles.

Administrative roles are similar in most ways to non-administrative roles, except that:

- Administrative roles are in sysadmin group 14, which is necessary for performing NIS+-related tasks, and
- Administrative roles work in a special *administrative role workspace* whose processes have the *trusted path attribute*, which is required by many programs that perform administrative tasks.

Most tasks are divided between the security administrator (secadmin) role and the system administrator role (admin), both of which are administrative roles. A third administrative role, the root role, is used to install software that requires a real UID of root in order to succeed.

Non-administrative Roles

In the default system, the only non-administrative role used to do administration work is the operator role (oper), which is assigned the backing up of file systems.

Logging In and Assuming a Role

In the Trusted Solaris system, administrators do not log in directly as administrators. Following is the sequence of how roles of both types log in and begin work in the window system (the only difference between administrative and non-administrative roles are the final two bullets):

- The security administrator assigns an administrative or non-administrative role to a user's account, using the User Manager.

- The security administrator gives to the user both the password for the user account and the password for the role.
- The user who is configured to be able to assume a role logs in by supplying his or her own username and password.
- To begin work in the role, the user assumes the role by choosing the assume role option from the Trusted Path menu, and by providing the role password to the role password dialog box.
- For administrative roles, a special *administrative role workspac*e becomes active, and the person then may act in the administrative role to perform administrative tasks.
- For non-administrative roles, a normal user workspace is created that does not have the trusted path attribute that is checked by many administrative programs.

Auditing of Administrative Activities

At login, a UID that also serves as an audit ID becomes associated with all processes generated by the logged-in user. If a user assumes a role, the user's effective ID changes, but the audit ID does not. The same is true for programs that run with an effective UID, the audit ID of the process running the program still identifies the real user. The audit ID makes it possible to trace back to an identified user account the actions that a user performs while in a role. Requiring users to log in and identify themselves before assuming a role plugs one possible security hole that can occur in a normal UNIX system, in which any individual who discovers the root password can log in directly and have unlimited anonymous access to all the information on the system.

How Logins are Enabled

Logins are disabled by default after every reboot. When logins are disabled, only a user account with the *enable logins* authorization can log in or enable logins. The purpose of disabling logins is to make it possible for the system administrator or other trusted employee to verify that the security of the system has not been compromised before allowing anyone else to log in again.

If the system has been rebooted and logins have not yet been enabled, one of two dialog boxes displays after the user enters his or her password. If the current user account is not authorized to enable logins, the dialog box shown below displays.

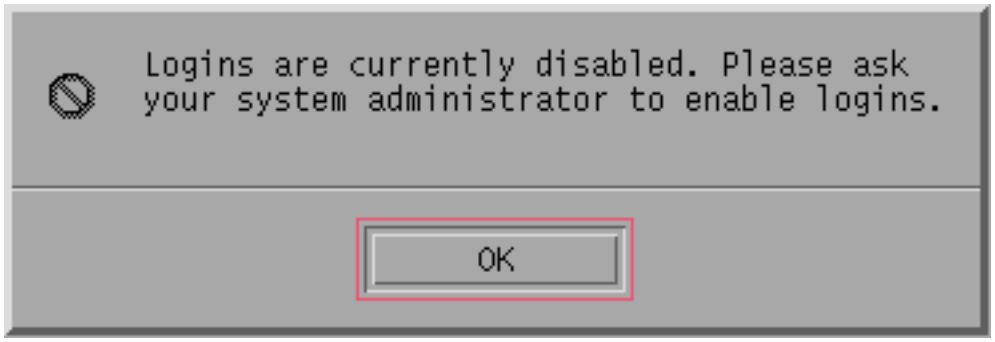


Figure 1–1 Disabled Logins Dialog Box for a User Not Authorized to Enable Logins

If the account logging in is authorized to enable logins, the dialog box shown below displays.

Figure 1–2 Disabled Logins Dialog Box for a User Authorized to Enable Logins

Preventing Logins From Being Disabled After a Reboot

The presence of the `/etc/nologin` file is checked during the login sequence after a host is rebooted, and logins are disabled if the `nologin` file is present. If it is consistent with the site's security policy to allow logins by everyone after all reboots, the security administrator can edit the `S05RMTMPFILES` script in `/etc/rc2.d` to comment out the lines that create the `/etc/nologin` file. See "To Prevent Logins from Being Disabled After a Reboot" on page 32, if changing the default is consistent with your site's security policy.

Before removing the lines that create the `nologin` file, save a copy of `S05RMTMPFILES`, making sure that the copy has the original files' DAC and MAC attributes in case you wish later to restore it. The `S05RMTMPFILES` file's attributes to be maintained are ADMIN_LOW[ADMIN_LOW], owner root and group sys.

Assuming an Administrative Role

The logged in user assumes an administrative role by doing the following:

- By clicking and holding the right (menu) mouse button on the workspace switch area to drag down the Trusted Path (TP) menu

- Or by clicking and holding down the menu button on a workspace button to drag down the Workspace *button_name* menu
- And selecting Assume *role_name* Role

The following figure shows the Workspace *One* menu and the TP Menu.

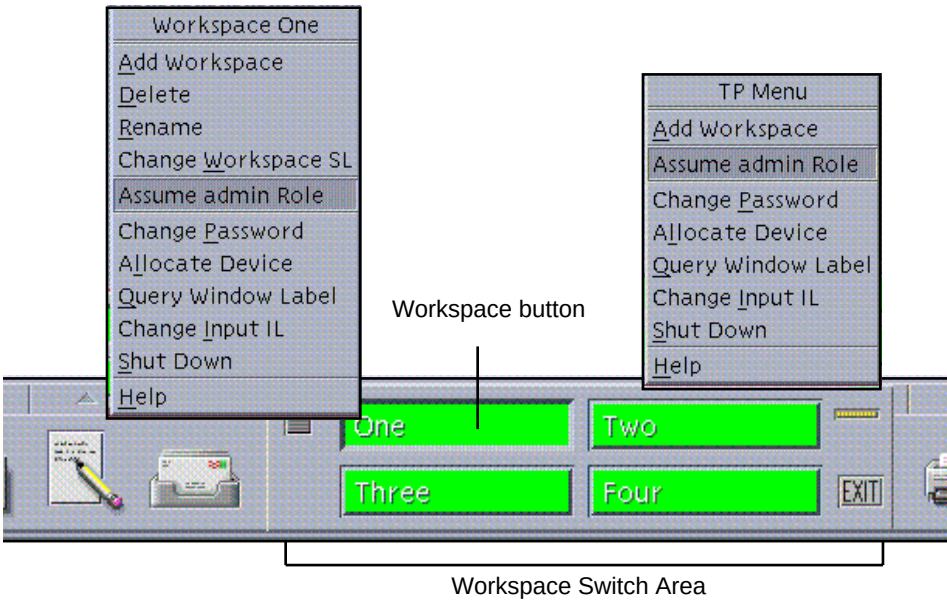


Figure 1–3 Choosing the Assume admin Role Option from the Workspace *button_name* Menu or Trusted Path Menu

---

Note - The Shut Down option only appears if the user has the shutdown authorization,

---

Working in the Administrative Role Workspace

Immediately after a user assumes an administrative role for the first time, an administrative role workspace becomes active at the role's minimum label, ADMIN_LOW, and a new workspace button for the role displays in the workspace switch area. The new role workspace button is highlighted along the top and left edges and is given the name of the administrative role account, as shown in Figure 1–4. The administrative role can change the name of the button just like anyone can change the name of any other workspace.

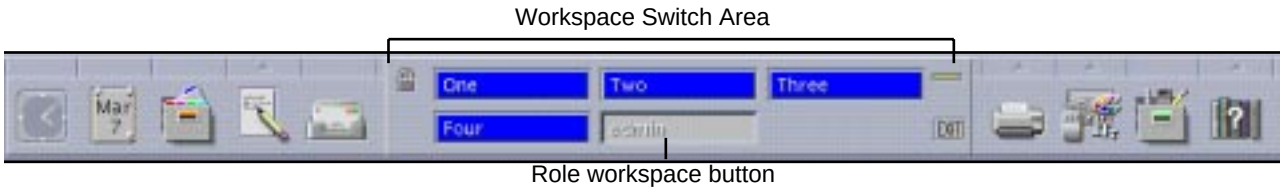


Figure 1–4 Workspace Switch Area with a Button for the admin Administrative Role

If an administrator invokes the Add Workspace option from the menu on a normal workspace, a *normal workspace* is created. If Add Workspace is invoked from the menu on an administrative workspace, then a new *administrative role workspace* is created. See “To Bring Up New Role Workspaces and Relabel Them,” and the following figure.

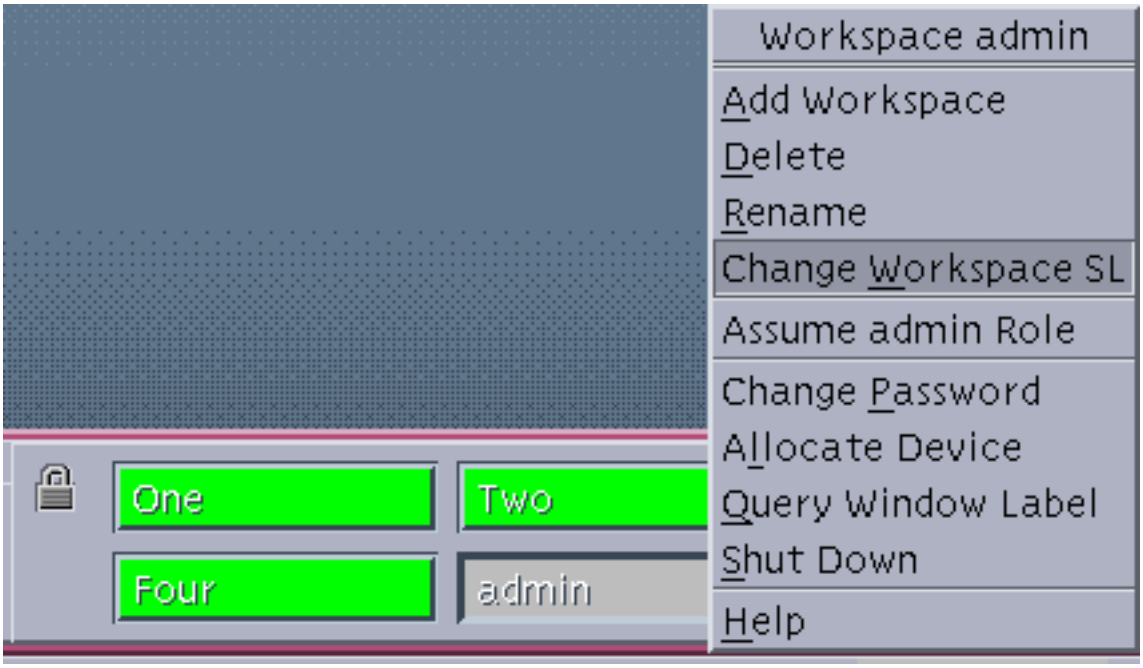


Figure 1–5 Creating a New Role Workspace from an Administrative Workspace Menu

The user who has assumed a role may return to any non-administrative role workspace at any time to work with the commands and actions specified in that account’s execution profile. See “To Switch Among Administrative Role Workspaces and the Normal User Workspaces” on page 25.

Administrative roles most often work at the ADMIN_LOW and ADMIN_HIGH administrative labels. After the first workspace comes up at ADMIN_LOW, if you want to create a new workspace and relabel it for working at ADMIN_HIGH or

another sensitivity label, follow the instructions under “To Work at Multiple Labels While in an Administrative Role ” on page 25.

The buttons for any new workspaces are saved with their names when the administrator logs out. After logging in again as a normal user, the administrator can access any of his or her administrative workspaces again by clicking on the administrative workspace’s button and supplying the role password.

Application Manager Folders and Actions Icons

Much of the administrative work performed by administrative roles is done by launching administrative programs from folders within the Application Manager. In keeping with the principle of least privilege, the set of administrative programs is divided between the two major administrative roles. If a role is not allowed to use an application, its icon does not display.

Using Solstice Administrative Tools in the Solstice_Apps Folder

Administrative roles do most administrative work by launching administrative applications from the Solstice_Apps folder within the Application Manager. See “To Launch Solstice Administration Tools” on page 27. All configuration files that are managed as NIS+ maps are edited through the applications in the Solstice_Apps folder.

Using Administrative Actions in the System_Admin Applications Folder

Other tasks performed by administrative roles, such as the editing of most configuration files that are not managed through NIS+, are done by using the administrative actions that are found in Trusted Solaris version of the System Administration Folder in the Applications Manager.

Accessing the Application Manager

The Application Manager is accessed by clicking its icon. Figure 1–6 shows the Application Manager icon selected in the mid-right side of the Front Panel and the Application Manager folder displayed above the Front Panel.

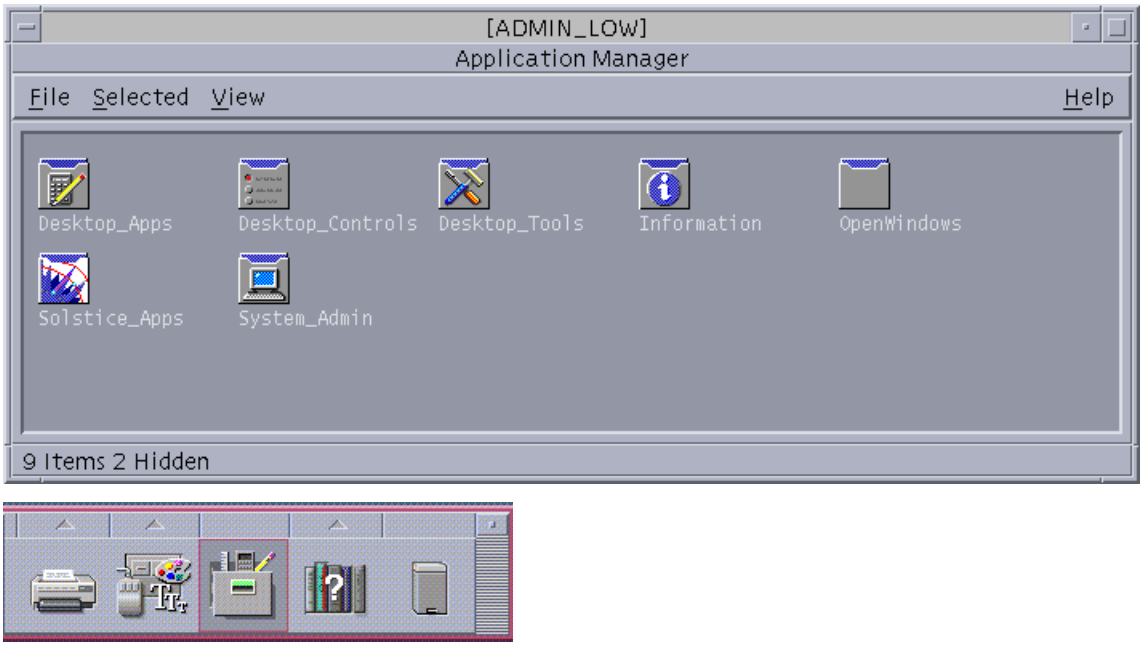


Figure 1–6 Application Manager Icon Selected in the Front Panel, and the System_Admin Folder Selected in the Application Manager Folder

Administrative Actions in the System_Admin Folder

The following figure shows all the administrative actions that can be accessed in the System_Admin folder by any of the roles. Table 1–1 describes the purpose of each action and the default role or roles to which it is assigned.

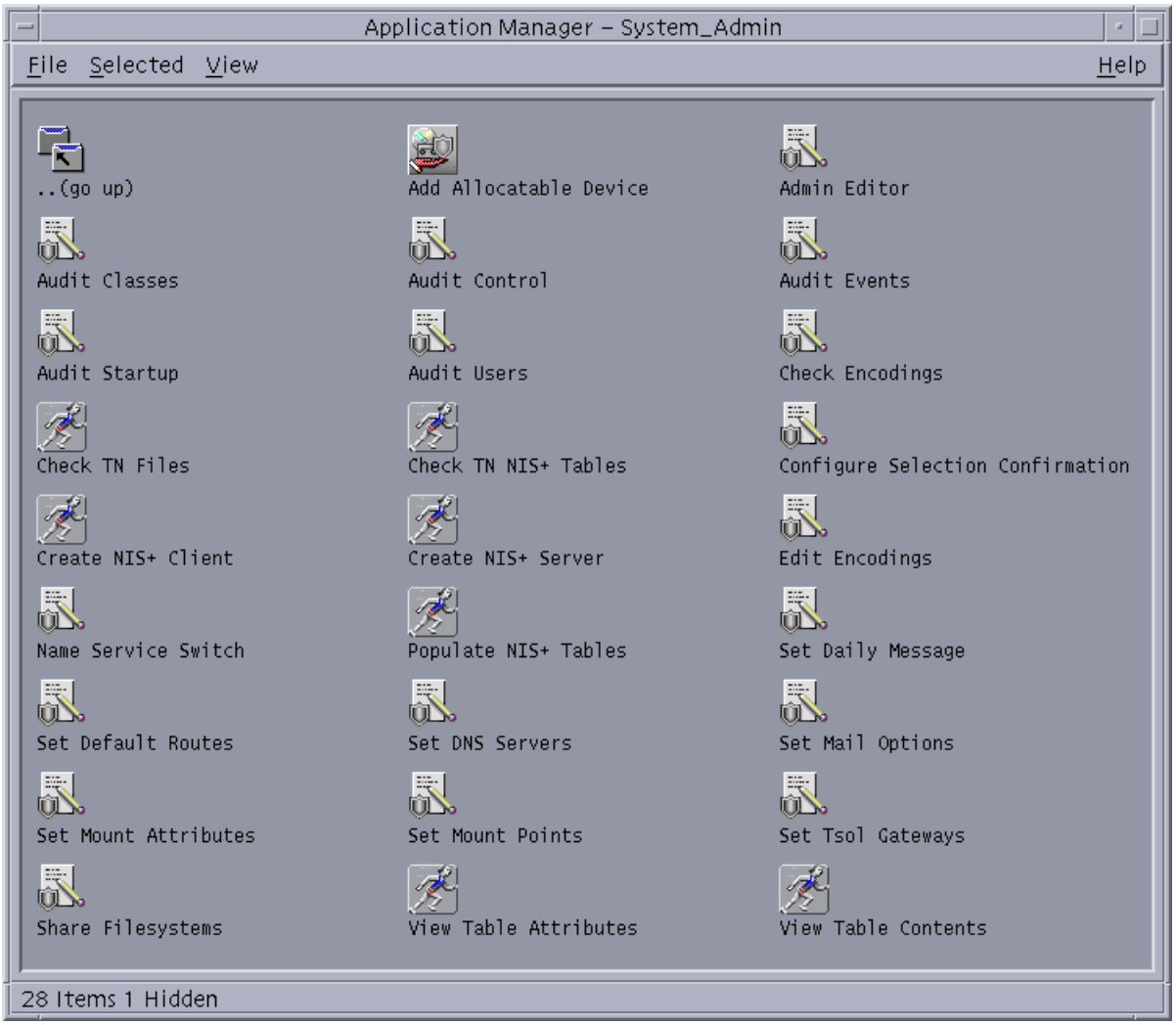


Figure 1–7 Administrative Actions in the System_Admin Folder

TABLE 1–1 Administrative Actions, Purposes, and Default Roles

Action Name	Purpose of Action	Default Role
Add Allocatable Device	Creates entries in device_allocate(4TSOL), and device_maps(4TSOL), and creates an auxiliary file for a new allocatable or non-allocatable device. User enters device name, device type, and lists all device special files associated with the device. See add_allocatable(1MTSOL).	secadmin
Admin Editor	Edits any specified file	secadmin
Audit Classes	Edits audit_class(4TSOL)	secadmin
Audit Control	Edits audit_control(4TSOL)	secadmin
Audit Events	Edits audit_event(4TSOL)	secadmin
Audit Startup	Edits the `audit_startup.sh` script [see audit_startup(1MTSOL)]	secadmin
Audit Users	Edits audit_user(4TSOL)	secadmin
Check Encodings	Runs chk_encodings(1MTSOL) on specified encodings file	secadmin
Check TN Files	Runs tnchkdb(1MTSOL) on local tnidb(4TSOL), tnrhdb(4TSOL), and tnrhtp(4TSOL) files	secadmin and admin
Check TN NIS+ Tables	Runs tnchkdb(1MTSOL) on tnrhdb(4TSOL) and tnrhtp(4TSOL), NIS+ trusted network table	secadmin and admin
Configure Selection Confirmation	Edits `/usr/dt/config/sel_config` [see sel_config(4TSOL)]	secadmin
Create NIS+ Client	Runs nisclient(1M)using the specified hostname for the NIS+ master and the specified domain name	secadmin
Create NIS+ Server	Runs nisserver(1M)using the specified domain name	secadmin
Edit Encodings	Edits specified label_encodings(4TSOL) file and runs chk_encodings(1MTSOL)	secadmin
Name Service Switch	Edits nsswitch.conf(4TSOL)	admin
Populate NIS Tables	Runs nispopulate(1MTSOL)from the specified directory	secadmin
Set Daily Message	Edits `/etc/motd`	admin
Set Default Routes	Edits /etc/defaultrouter [see route(1MTSOL)]	admin

TABLE 1–1 Administrative Actions, Purposes, and Default Roles *(continued)*

Action Name	Purpose of Action	Default Role
Set DNS Server	Edits resolv.conf(4)	admin
Set Mail Options	Edits `/etc/mail/sendmail.cf` [see sendmail(1MTSOL)]	admin
Set Mount Attributes	Edits vfstab_adjunct(4TSOL)	secadmin
Set Mount Points	Edits vfstab(4TSOL)	admin
Set Tsol Gateways	Edits tsolgateways(4TSOL)	admin
Share Filesystems	Edits dfstab(4); does not run share(1MTSOL)	admin
View Table Contents	Runs niscat(1) on the specified NIS+ trusted network database	secadmin and admin

See also “To Launch Administrative Actions” on page 29.

All the administrative actions that edit files use the `/usr/dt/bin/trusted_edit` shell script, which brings up a restricted editor and audits any changes made at the time the file is saved.

Whichever editor is specified in the $EDITOR variable for the role is used, `/bin/admin`vi by default. The security administrator can redefine the $EDITOR variable to `/usr/dt/bin/dtpad`. When adminvi(1MTSOL) is specified, `adminvi` is invoked as root to edit the file. The `adminvi` command prevents the saving of the file with any other name. [See “Administrative `vi`” on page 15 and the man page for the characteristics of adminvi(1MTSOL).] If `dtpad(1)` is specified, the New, Save, and Open items in the File menu are disabled when the action runs, so that the file cannot be renamed.

If desired, see “Assigning `trusted_edit` as a Role’s Default Editor” on page 105 and “To Assign the `trusted_edit` Editor to a Role” on page 105.

Accessing Commands and Actions

While in the administrative role workspace, the person acting in the role has access only to the commands and actions with the attributes specified for them in the role’s execution profile or profiles. The application or action is launched at the sensitivity label of the currently active role workspace.

Using the Profile Shell to Do Tasks on the Command Line

The profile shell, pfsh(1MTSOL), is the default shell for administrative roles. It is used in Trusted Solaris administration to restrict each role to the minimum set of commands that the role needs for doing its tasks, while allowing the commands to inherit any privileges they need while executing. Profile shells may be run at any sensitivity label. Administrators access the profile shell by bringing up `dtterm` or any other terminal emulator from the front panel. The profile shell is started in the terminal at the sensitivity label of the current workspace.

Administrative `vi`

The adminvi(1MTSOL) command is a modified version of the `vi` command that is restricted to prevent the user from executing shell commands and from writing to (saving to) any file other than the original file being edited. The Admin Editor action, which is assigned to the security administrator role by default, should be used in most cases instead of `adminvi` on the command line to edit or create administrative files. The `adminvi` command may be assigned to any users whose default shell is the profile shell, if the security administrator wishes to allow those users the use of a text editor while constraining them to edit files with `adminvi`'s built-in restrictions.

Administrative Role Procedures

▼ To Login and Assume an Administrative Role

1. **To work on a remote host, use the remote log in option from your local host's Login Screen.**
   a. **Log out.**
   b. **On the Login dialog box, click and hold the right mouse button over Options, drag the mouse button down to select Remote Login from the Options menu, and select either Enter Host Name... or Choose Host From List....**
   c. **Enter or highlight the name of the remote host.**
2. **Log in as a normal user, supplying your own username and password.**
   See the following figure.

Figure 1–8 Login Dialog Box

After you enter your username and password, if the workstation information dialog box displays as shown in Figure 1–11, go to Step 4 on page 17.

If logins are disabled, either the following figure or Figure 1–10 display.

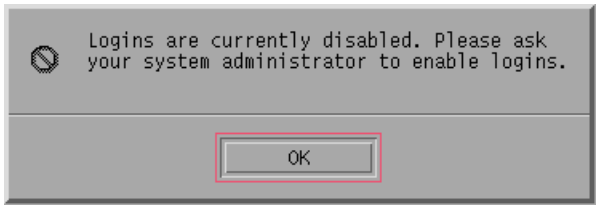


Figure 1–9 Disabled Logins Dialog Box for a User Not Authorized to Enable Logins

3. Enable logins if necessary.

a. If your account is not authorized to enable logins, notify the security administrator.

b. If your account is authorized to enable logins, click the Enable Logins radio button to enable logins.

See Figure 1–10.

Figure 1–10 Disabled Logins Dialog Box for a User Authorized to Enable Logins

4. **Review the information provided on the workstation information dialog box (Figure 1–11) and, if allowed, choose between a single sensitivity label or multiple sensitivity label session.**
   a. **Check the date and time of the last login to ensure there is nothing suspicious 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 suspicious activity, 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, depending on whether you want to start work in a multilabel or single-label session.**

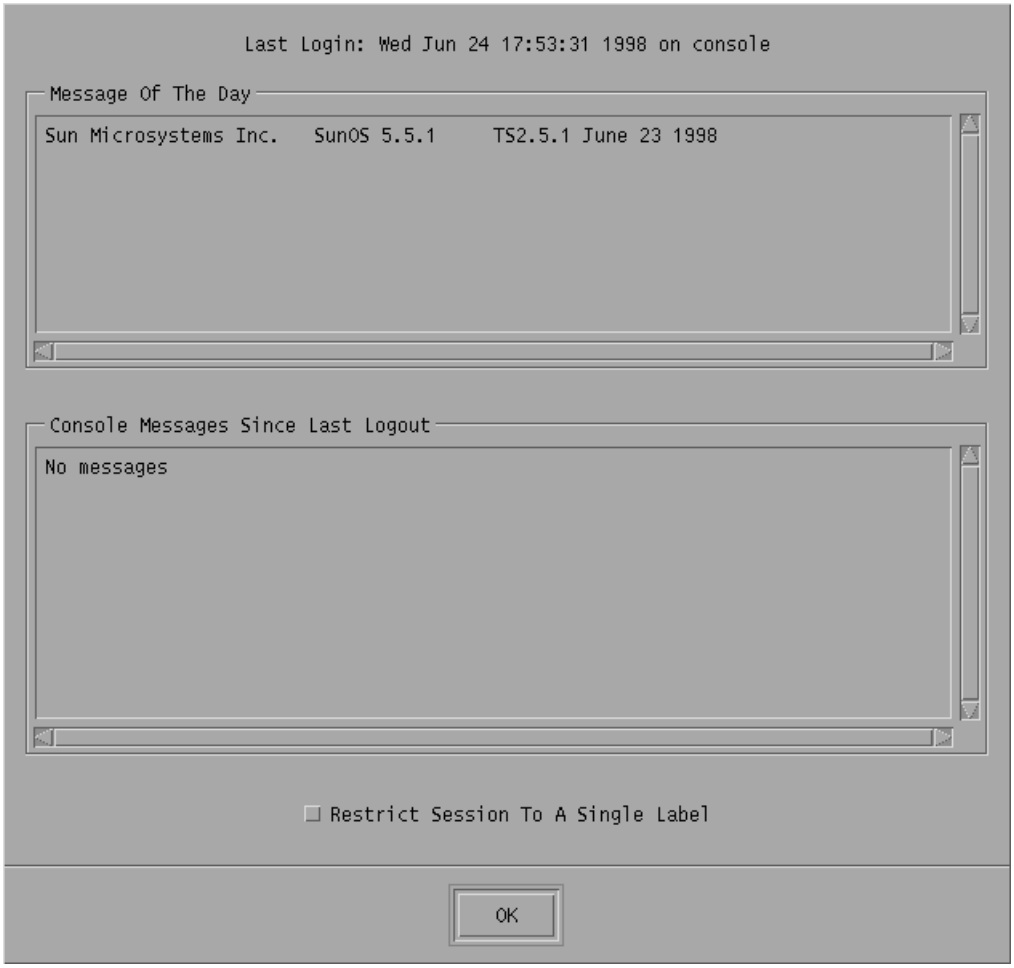


Figure 1–11 Workstation Information Dialog Box

Note - If your account is configured to work at only one sensitivity label, “Single Level Session SL:” displays at the bottom of the workstation information dialog box followed by the sensitivity label at which you are configured to work. For example, if your single sensitivity label is CONFIDENTIAL, the text shown in Figure 1–12 displays.

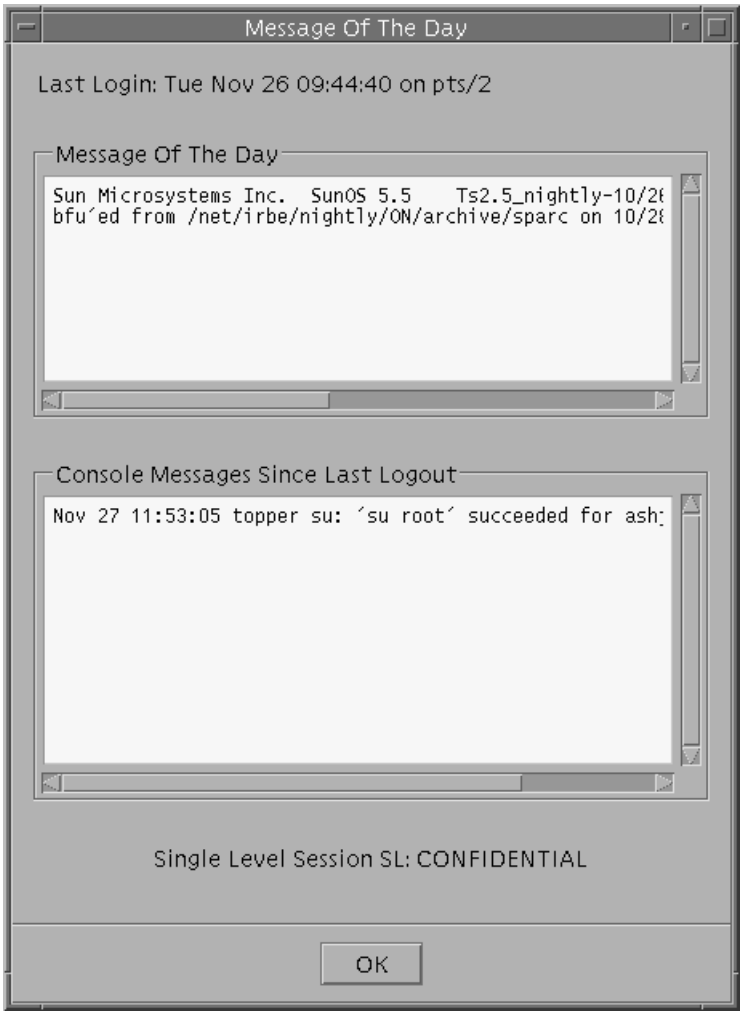


Figure 1–12 Single Label Indicator on the Workstation Information Dialog Box

5. **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 Setting User Session SL dialog box displays (see Figure 1–13 and go to Step 6 on page 20).

   If you are allowed to work at multiple sensitivity labels and if you did not check the box next to Restrict Session to a Single Label, on the Workstation Information dialog box, the Setting User Session Clearance dialog box displays (see Figure 1–14 and go to Step 7 on page 21).

6. **To specify a sensitivity label for a single-label session, either accept the default sensitivity label or specify a sensitivity label in the Single Level Session: Setting User Session SL Label Builder.**

   a. **To type in the sensitivity label, use the text entry field under Update With, and click Update when done.**

   b. **To use the mouse to build the sensitivity label, choose a classification from the CLASS menu and select the compartments by checking the boxes next to the compartment names under COMPS.**

   c. **Click OK.**

   d. **Go to Step 8 on page 24.**

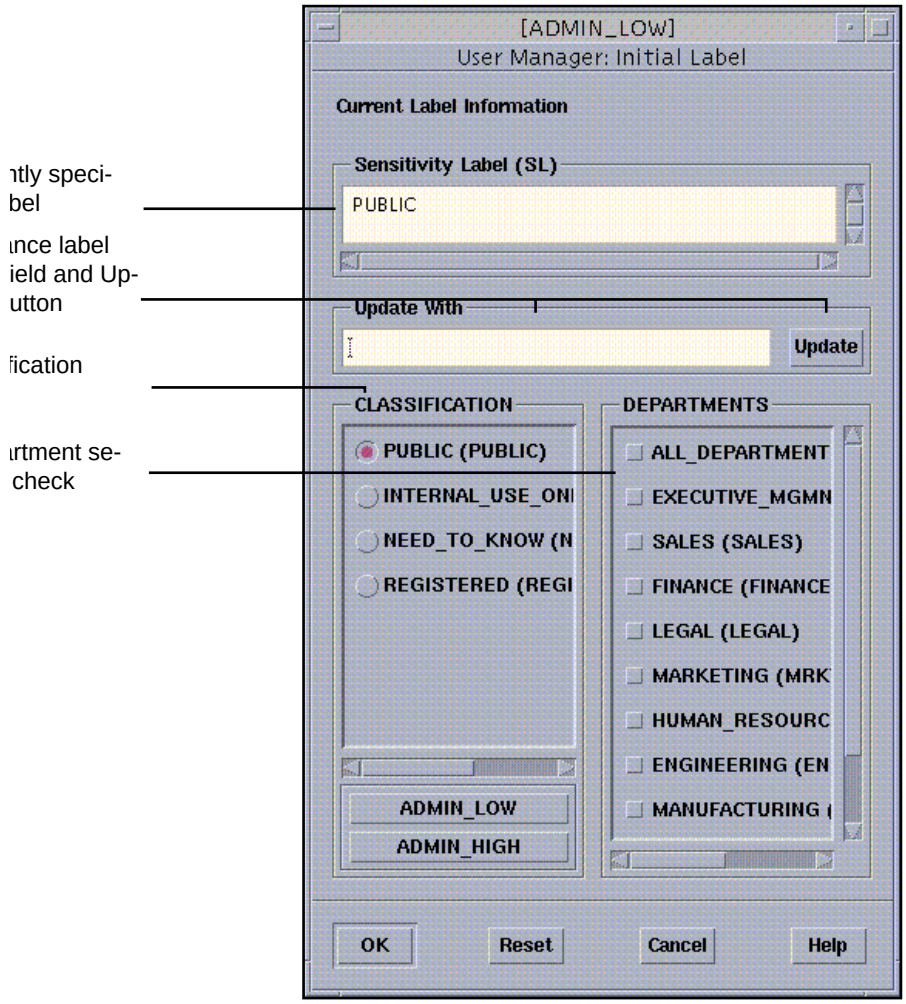


Figure 1–13 Label Builder Dialog Box for a Single-sensitivity Label Session

7. To specify the clearance for a multilabel session, either accept the default clearance shown in the Multi Level Login Setting User Session Clearance label builder, 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.

c. Click OK.

d. Go to Step 8 on page 24.

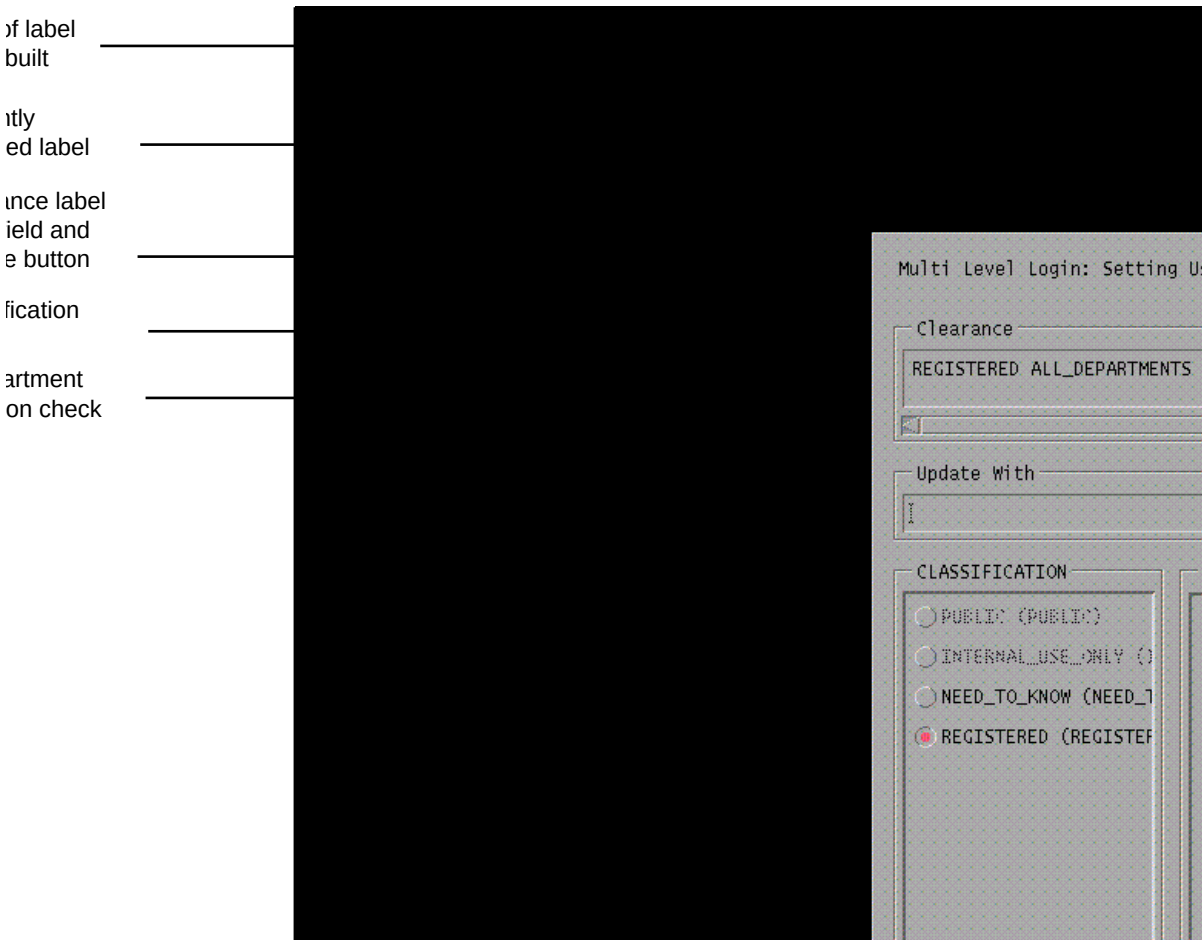


Figure 1–14 Session Clearance Dialog Box

8. Choose the assume role option from the Trusted Path menu

Figure 1–15 shows the Trusted Path 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. A user configured for the security administrator role sees `Assume secadmin Role`; a user configured for the system administrator role sees `Assume admin Role`, and so on. (secadmin is the account name for the security administrator role and admin is the account name for the system administrator role).

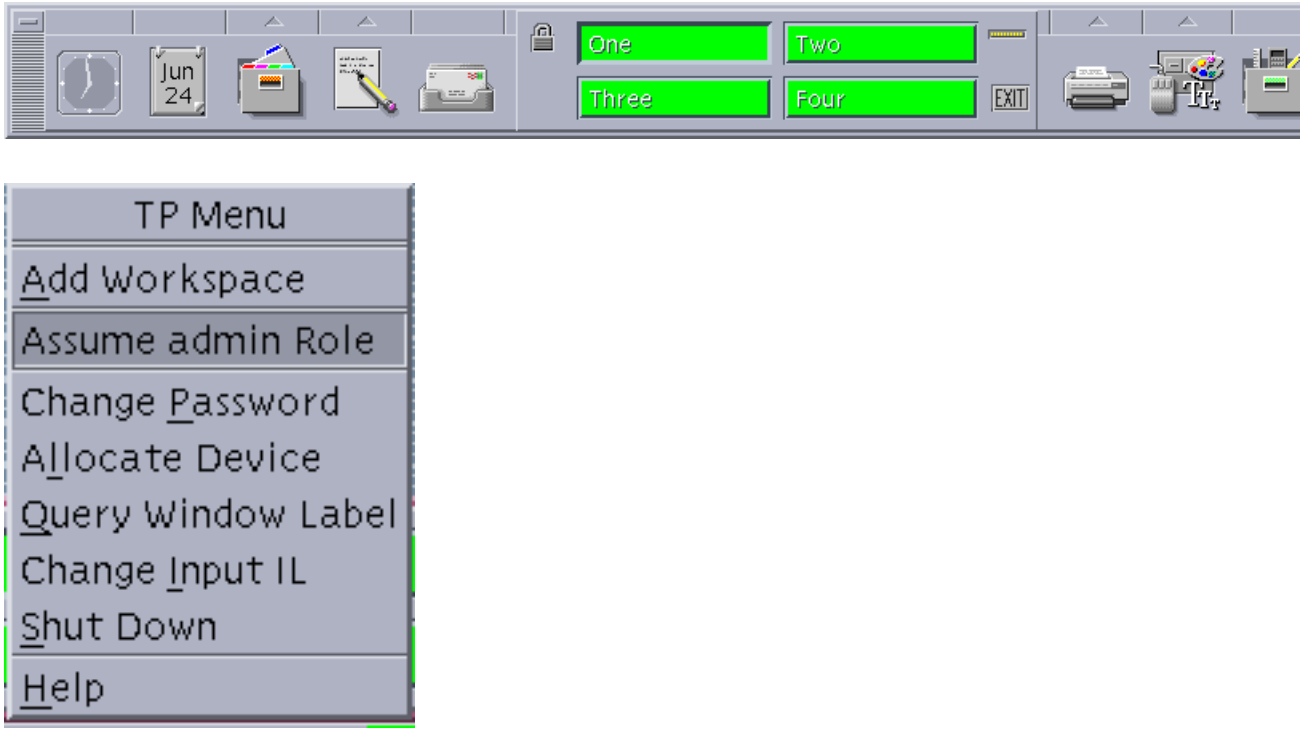


Figure 1–15 Choosing the Assume admin Role Option from the Trusted Path Menu

The Role Password dialog box displays.

9. Enter the role password in the role password dialog box (see Figure 1–16), and click OK.

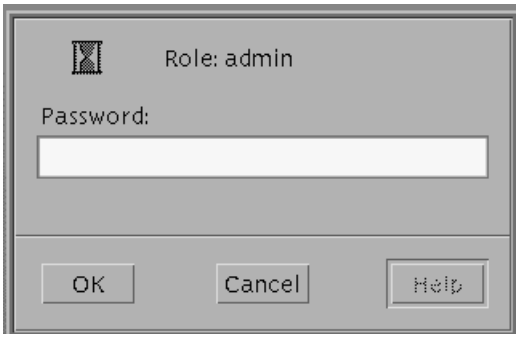


Figure 1–16 Role Password Dialog Box

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

▼ To Switch Among Administrative Role Workspaces and the Normal User Workspaces

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

▼ To Work at Multiple Labels While in an Administrative Role

Note - 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 a role workspace button, click and hold the menu (right) mouse button to bring up the Trusted Path menu.**

      Note - As shown in the following figure, he cursor must be over an administrative role workspace's button, for Add Workspace to add an administrative role workspace.

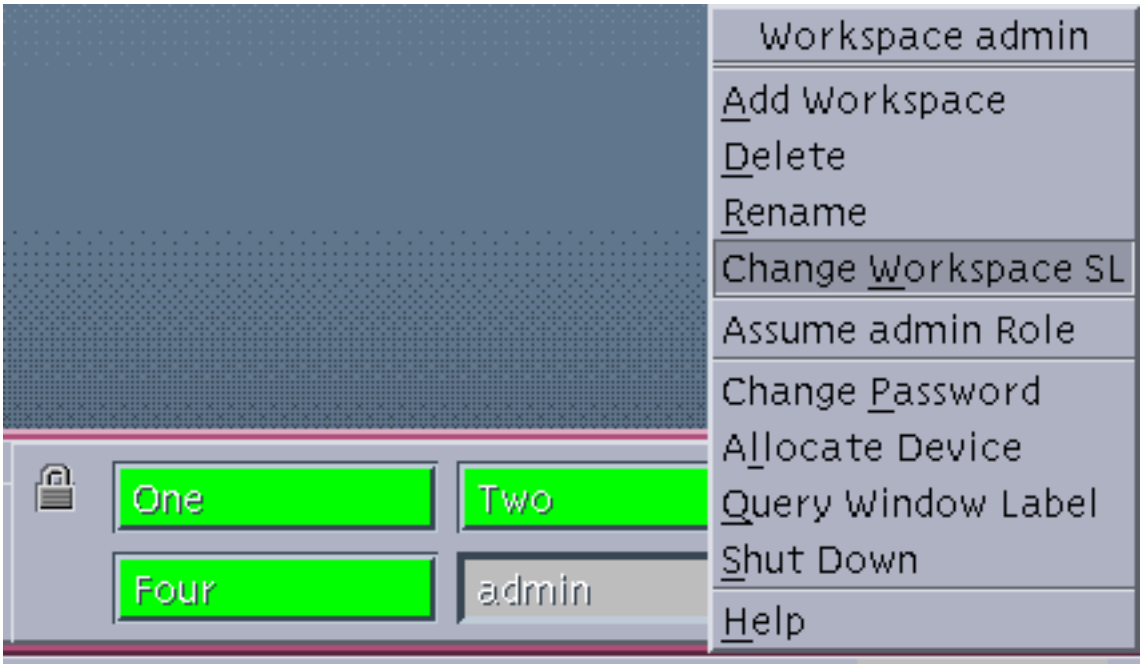


Figure 1–17 Creating a New Role Workspace from an Administrative Workspace Menu

b. Choose Add Workspace from the Workspace *role* 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, as shown in Figure 1–18.

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`.

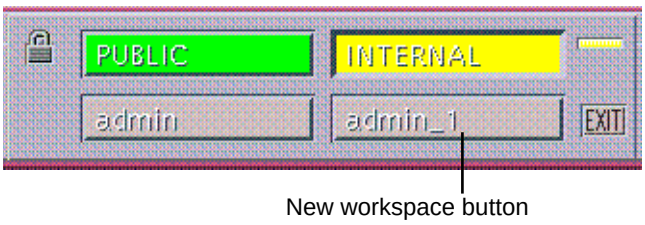


Figure 1–18 A New admin_1 Workspace Button for a New Administrative Role Workspace

2. **Change the sensitivity label of the workspace.**
   a. **Click the cursor on the new role workspace button.**
   b. **Click the right (MENU) mouse button on the workspace switch button to bring up the Workspace *role_name* menu.**
   c. **Choose Change Workspace SL from the Workspace *role_name* menu.**
      The Label Builder displays.
   d. **Type in the desired sensitivity label in the text entry field under Update With on the Label Builder dialog box, and then click the Update button and click OK at the bottom of the dialog box when done.**
      The sensitivity label on the workspace changes to the sensitivity label you specified in the label builder.

▼ To Launch Solstice Administration Tools

1. **Log in as a user who is able to assume the role and assume the role.**
   See "To Login and Assume an Administrative Role" on page 15 if needed.
2. **Single-click the Application Manager icon on the Front Panel.**

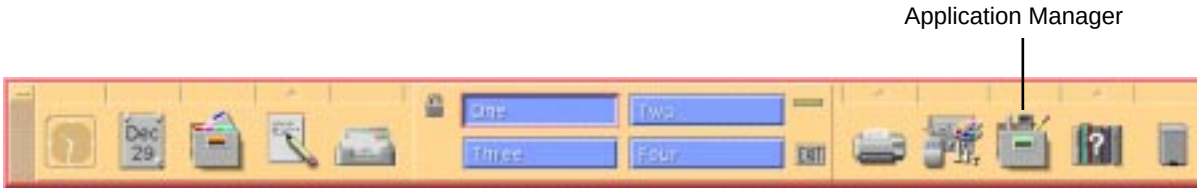


Figure 1–19 Application Manager Icon on the Front Panel

The Application Manager folder displays.

3. **Double-click the Solstice_Apps icon in the Application Manager.**

The Solstice_Apps folder displays as shown in Figure 1–20.

Figure 1–20 Solstice Applications in the Solstice_Apps Folder

4. **Double-click the desired application's icon in the Application Manager to bring up the Load List.**

5. **Select a naming service from the Naming Service menu on the application's Load list.**

Note - By default, as shown in Figure 1–21, the NIS+ naming service is selected, and the name of the NIS+ domain displays in the Domain text entry field.

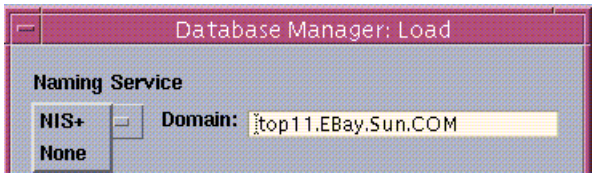


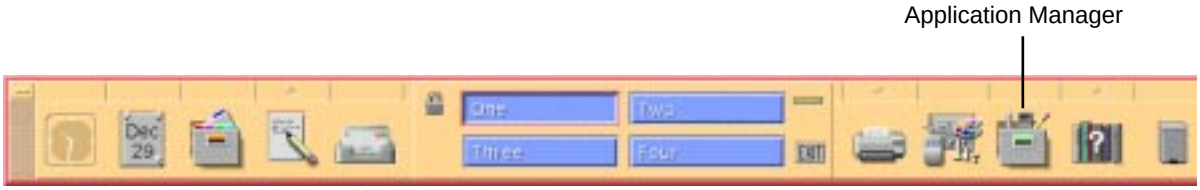
Figure 1–21 Loading a Naming Service in a Solstice Application

Choose None if you want the changes to be in a local file, or if you are on a workstation that is not running the NIS+ naming service. Choose NIS+ for the Naming Service if you want the changes to be in a NIS+ table, seen by all workstations on the network.

▼ To Launch Administrative Actions

1. **Log in as a user who is able to assume the role and assume the role.**
   See “To Login and Assume an Administrative Role” on page 15 if needed.

2. **Single-click the Application Manager icon 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 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” on page 29, 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. **Save your changes and quit the file.**

```
:wq
```

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

▼ 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 Login and Assume an Administrative Role" on page 15, and "To Use the Admin Editor Action to Edit a File" on page 29, 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.**

```
    E XEC_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` tile.

```
: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 placed in the `/etc/dt/appconfig/types/C` directory.

▼ To Prevent Logins from Being Disabled After a Reboot

1. Assume the security administrator role.

See "To Login and Assume an Administrative Role" on page 15, if needed.

2. In an ADMIN_LOW workspace, use the file manager or commands in a terminal emulator to make a backup copy of `/etc/rc2.d/S05RMTMPFILES`.

```
$ cd /etc/rc2.d
   $ cp S05RMTMPFILES S05RMTMPFILES.orig
```

3. Use the Admin Editor action to open the `/etc/rc2.d`/S05RMTMPFILES for editing.

See "To Use the Admin Editor Action to Edit a File" on page 29, if needed.

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

a. Find the lines shown in the screen below.

```
# The file /etc/nologin will disable all logins until the
   # logins are enabled by an authorized user via login(1)
   # or dtlogin(1).
```

```
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
```

b. 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
```

5. Write and quit the file.

```
:wq
```

CHAPTER 2

Miscellaneous Tasks and Procedures

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

This chapter includes the following procedures:

Security Requirements

The rules in this section are especially important if your site's security policy mandates that your practices be consistent with those required for an evaluated configuration. Administrators need to protect passwords and files and to administer auditing in a manner that ensures that the security of the system is not compromised.

User Training About Security Requirements

The security administrator should set up processes to train users. The following rules should be handed off to new employees, and existing employees should be reminded of these rules from time to time. (You may wish to provide additional suggestions.)

- Do not disclose your password to anyone, since anyone with your password can access the same information that you can without being authenticated and therefore without being accountable.
  - Do no 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.

  This requirement is enforced by the Choose Password option from the Trusted Path menu, as shown in Table 2–1.
- Do not leave your workstation unattended without locking the screen or logging off.

- Be aware that mail can be changed to forge the sender information or the X-sender information label.
- Be aware that administrators at your secure installation do not rely on email to pass instructions to users. Do not ever follow instructions from administrators in an email without verifying with the administrator first.
- Do not send your password to anyone by email.
- Be aware that files and directories that you create are protected by the access permissions that you set. Make sure that the permissions on files and directories are set so that no unauthorized user can read or write a file or list the contents of or write into a directory.

User and Role Account Security

The security administrator is responsible for specifying the original password for each account. The passwords chosen by the security administrator should meet the same requirements enforced by the Change Password Option in the Trusted Path menu, which are shown in Table 2–1.

TABLE 2–1 Password Rules for Manually-Created Passwords

Rules
The password must be eight characters in length.
The password must contain at least two alphabetic characters.
The password must contain at least one numeric or special character.
The password must differ from the user's login name and any reverse or circular shift of that login name. (Fo >comparison, upper case letters and lower case letters are considered to be equal.)
A new password must have at least three characters different from the old. (For this comparison, upper case lette >lower case letters are considered to be equal.)

The administrator role must specify a unique user name, and the security administrator must specify a unique user ID when creating a new local or NIS+ managed account. When choosing the name and ID for a new account, the administrators must ensure that both the user name and associated UID are not duplicated anywhere on the network.

The administrators must also make sure never to 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 user did what or over which user owns which files when archived files are restored or audit records are analyzed.

The security administrator can change any account's password at any time (except for the password of the security administrator account, which requires the *permit self modification* authorization). The security administrator should change an account's password if there is any suspicion that the password has been discovered by anyone who should not know it. The security administrator should hand over the password to an account in such a way that the password cannot be eavesdropped by anyone else.

Administrators should not use email to instruct users to take an action and should 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 tell users to change a password to a certain value, which could subsequently be used to login and compromise the system.

The security administrator specifying passwords for accounts should make sure that the accounts for users who are able to assume the security administrator role are configured to be Always Open. This ensures. that at least one account can always log in and assume the security administrator role and re-open everyone's account, if ever all accounts are locked.

The accounts for administrative roles should not have any password aging specified for them, so that their passwords do not expire.

Protecting Information

Administrator 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, local tsolprof(4TSOL) and tsoluser(4TSOL) databases, and the audit trail).

Because the protection mechanisms that apply to NIS+ tables are not subject to the access control policy enforced by the Trusted Solaris system, the default NIS+ tables should not be extended and their access rules should not be modified in any way.

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
```

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

TABLE 2–2 Required Attributes of /etc/shadow

MAC	**ADMIN_LOW[ADMIN_LOW]**		
DAC	owner	group	permissions
	root	sys	400

The password field in the NIS+ passwd.org_dir table is protected by NIS+ access restrictions to fields within tables. When any user or administrator tries to view the passwd.org_dir table, the only encrypted password shown is that belonging to the account, and there is no shadow.org_dir table.

The following example shows that user ricc's password field shows as *NP*, for the user, roseanne, who invoked the niscat(1) command, but roseanne can view 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::::::
. . .
trusted6% niscat shadow.org_dir
shadow.org_dir: Not Found, no such name
```

Creating Groups

The administrator needs to verify on the local system and on the network that all groups have a unique GID.

Deleting Users

When an account is deleted from the system, the following must be done:

- The account's home directory must be deleted.

  See "To Remove an MLD" on page 59, if needed.

- Any processes or jobs belonging to the deleted account must be removed.

- Any objects owned by that account must be deleted or the ownership must be assigned to another user.
- Any batch jobs scheduled on behalf of the user must be deleted.

- The user (account) name and UID must be retired and never reused.

Deleting Groups

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

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

Distributing Changed Configuration Files to Hosts Across the Network

The following procedure is for distributing configuration files are not distributed by NIS+. Some configuration files that should be the same on all hosts in a distributed system are:

- label_encodings(4TSOL)
- system(4)

If need be, the security administrator can use the rdist(1) command to automatically distribute identical copies of the file to machines in the distributed system.

▼ To Remotely Distribute Configuration Files

Note - Make sure that every host has only a plus in the `hosts.equiv` file, and that there are no entries in either the `/.rlogin` or in any `$HOME/.rlogin` files.

---

1. **Login, assume the security administrator role, and use the Admin Editor action from the System_Admin folder in the Application Manager to open a `distfile` for editing.**

   See “To Login and Assume an Administrative Role” on page 15 and “To Use the Admin Editor Action to Edit a File” on page 29, if needed.

2. **Add entries to the** `distfile` **to copy the configuration files from a master directory.**

   The example shows a sample `distfile` set up to tell rdist(1) to copy the `label_encodings` and `system` file to all the listed hosts in the distributed system.

```
# # HOSTS = ( machiavelli muckraker mugwump diehard warhorse )
    FILES = ( /etc/security/tsol/label_encodings /etc/system )${FILES} -> ${HOSTS} install ;
```

3. **Save and close the file.**

```
:wq
```

4. **Run the** `rdist` **command.**

   `rdist` can be run in the same directory as the `distfile` or use `rdist` can be entered with the `-f` option followed by the full pathname of a file with some other name.

```
$ rdist  \* OR *\
$ rdist -f /home/machiavelli/jedgar/label_encodings.master/distfile.sample
```

See also the rdist(1) and hosts.equiv(4) man pages.

Reboot each machine.

Changing the Maximum Number of Bad Password Entries

By default, the Trusted Solaris system allows an account a maximum of three failed attempts to enter the correct password when logging into a host or when authenticating oneself in order to change a password. Enforcing this maximum helps forestall brute force attempts to guess an account's password by multiple attempts using different passwords. When an account enters the wrong password one time too many, the account is closed.

If an account is accidentally closed, the security administrator role can change the account's status from Closed to Open on the Status Menu in the User Manager Password dialog box.

The security administrator can also reset the maximum to be any number that is consistent with the site's security policy. The maximum is set on each host in the local `/etc/default/passwd` file.

The security administrator can also specify any user's account to be Always Open in the User Manager Password dialog box. The MAXBADLOGINS value does not apply to an account that is set up as "always open" (which is called "FIXED" in the `/etc/default/passwd` file). The accounts for administrative roles and the account of one user who is able to assume the security administrator role should be configured to be Always Open.

▼ To Change the Maximum Number of Failed Password Entries

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

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **As security administrator at ADMIN_LOW, use the Admin Editor action to open the `/etc/default/passwd` file for editing.**

   See "To Use the Admin Editor Action to Edit a File" on page 29, if needed.

3. **Search for the string MAXBADLOGINS.**

```
# MAXBADLOGINS determines how many login attempts will be allowed before
# a user's account is locked. Users with FIXED accounts will never be
# locked
MAXBADLOGINS=3
```

4. **Change the value set for MAXBADLOGINS= as desired.**

5. **Save and close the file.**

```
:wq!
```

Entering Labels in Configuration Files

Each site can choose whether to require labels in configuration files to be in text form or in hexadecimal form. Which form to use depends on the site's security policy. Labels entered in text form must be quoted.

Note - When labels are stored in human-readable form, the files that contain them must be protected at ADMIN_HIGH, so that only administrative roles with ADMIN_HIGH in their clearances can read them.

Getting a Hexadecimal Equivalent for Labels and Clearances

Depending on the site's security policy, administrators may need to enter labels and clearances into the trusted network databases and other configuration files in hexadecimal form.

Use atohexlabel(1MTSOL) with the options shown in the following steps to get the hexadecimal equivalents of a CMW label, sensitivity label, information label or clearance. The atohexlabel(1MTSOL) 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 CMW Label, an SL, IL, or Clearance

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

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **In an ADMIN_LOW workspace, bring up a terminal emulator with a profile shell [pfsh(1MTSOL)].**

3. **To get the hexadecimal value for a CMW label (an information label followed by a sensitivity label in brackets), use atohexlabel(1MTSOL) followed by the name of the CMW label.**

```
$ /usr/sbin/atohexlabel ADMIN_LOW
0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000[0x00000000000000000000000000000000000000000000000000000
00000000000000000000000]
```

4. **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 s
     0x00060c0000000000000000000000000000000000000000000003fffffffffff0000
```

5. **To get the hexadecimal value for an information label, use** `atohexlabel` **with the** `-s` **option followed by the name of the sensitivity label.**

```
$ atohexlabel -i s
        0x00060c0000000000000000000000000000000000000000000003fffffffffff0000
```

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

```
$ atohexlabel -c sa
     0x0006ac0000000000000000000000000000000000000000000003fffffffffff0000
```

Extending Extendable Security Mechanisms

The following Trusted Solaris security mechanisms can be added to by the security administrator role:

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

Adding audit events and audit classes is described in the *Trusted Solaris Audit Administration*. Adding execution profiles is described in the current manual in Chapter 8” under “Use of the Profile Manager to Create or Modify Execution Profiles” on page 201. Adding roles is described in this manual’s Chapter 4” under “Creating a New Role” on page 101. The rest of this section describes how to add authorizations and privileges.

Understanding Authorizations

Authorizations are granted to users only indirectly through the execution profiles that may be assigned to the user.

Zero or more authorizations are assigned to each execution profile through the Profile Manager, and are stored in the authorization field in the tsolprof(4TSOL)

entry for each profile. The authorizations field contains either a comma-separated list of authorization numbers or names or the key words *all* or *none*. The format of a `tsolprof` entry is as follows:

profile:*description*:*authorizations*:*actions*:*commands*:*links*:*flags;*

Caution - Do not edit the tsolprof file directly.

As shown in the following figure, the Audit Review profile has the terminal login authorization (#3) and the set file privilege authorization (#9).

CODE EXAMPLE 2–1 An Example `tsolprof` Entry

```
Audit Review:View The Audit Trail:none:none:/var/sbin;praudit;3,9;none;none,0x00000000000
0000000000000000000000000000000000000000000000000000000000000,0x7fffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffff:1:none
```

When setting up or modifying an account, the security administrator role assigns zero or more execution profiles using the Solstice User Manager. The names of the assigned profiles are stored in the profiles field in the tsoluser(4TSOL) entry for that account.

The following figure shows the comma-separated list of profile names in the profile field of the `tsoluser` entry for a locally-created administrative role account, *auditadmin*.

CODE EXAMPLE 2–2 Profiles in a `tsoluser` Entry for an Administrative Account

```
auditadmin:fixed:automatic:Audit Control,Audit Review,Media Restore,:none:5:lock:internal,
showil,showsl:0x0000:0x0000000000000000000000000000000000000000000000000000000000000000:
0x7fffffffffffffffffffffffffffff
     >fffffffffffffffffffffffff:utadm:res1:res2:res3
```

In the example shown in Code Example 2–2, the Audit Review profile is assigned along with the Audit Control and Media Restore profiles to the new role.

Extending the Trusted Solaris Authorizations

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

- `/usr/include/tsol/auth_names.h`
- `/usr/lib/tsol/locale/C/auth_name`

auth_names.h

The /usr/include/tsol/auth_names.h header file contains the manifest constants and associated numbers for authorizations. Up to 128 possible authorizations are allowed. As shown in the following figure, fifty-three (53) TSOL_AUTH_* authorizations are already defined; the TSOL_AUTH definitions for the default authorizations number from 1 to 52 (with 0 meaning no authorizations).

CODE EXAMPLE 2–3 TSOL_AUTH Defined Authorizations in auth_names.h

```
TSOL_AUTH_ENABLE_LOGIN = 1,  /* indirectly required*/
          .
          .
          .
TSOL_AUTH_REVOKE_DEVICE = 52,
```

As shown in the following figure, the authorizations from 53 to 89 are reserved for later Sun extension, and these are identified by the prefix tsol_auth_reserved.

CODE EXAMPLE 2–4 tsol_auth_reserved Authorizations in auth_names.h

```
tsol_auth_reserved53 = 53,
          .
          .
          .
tsol_auth_reserved89 = 89,
```

As shown in the following figure, the remaining authorizations identified with the auth_reserved prefix are available for any site to extend.

CODE EXAMPLE 2–5 Authorizations Available for Extension

```
auth_reserved90 = 90,
          .
          .
          .
 auth_reserved127 = 127
```

auth_name(4TSOL)

In the default system, the auth_name(4TSOL) file is in /usr/lib/tsol/locale/C. The following figure shows the format for an entry in the file for the default locale:

CODE EXAMPLE 2–6 Format of the auth_name File

constant:*name*:*description*

The constant value in /usr/lib/tsol/locale/*locale_name*/auth.h must be exactly the same as the manifest constant name for the authorization in the /usr/include/tsol/auth.h file. The name field is a concise and descriptive

name for the authorization, which is used in various GUIs, including the profile manager.

Note - The authorization name (in the example in Code Example 2–7, enable_logins) is displayed in the Profile Manager and other GUIs that display a list of authorizations.

The description field is a description of the activity permitted by the authorization. The definition may be as long as you need it to be. The definition is used in the Profile Manager to guide the security administrator when assigning authorizations to profiles.

Code Example 2–7 gives an example of an actual authorization in the default `auth_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 (\).

CODE EXAMPLE 2–7 Definition for the enable logins Authorization in the `auth_name` File

```
TSOLAUTH_ENABLE_LOGIN:enable logins:Allows a user to enable logins on a \
 machine that was just booted.  Until logins are enabled there is \
 no interactive use of the machine's resources.
```

▼ To Add An Authorization

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

---

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

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **As security administrator at ADMIN_LOW, use the Admin Editor action to open the `/usr/include/tsol/auth_names.h` file for editing.**

   See "To Use the Admin Editor Action to Edit a File" on page 29, if needed.

3. **Create an entry in the `auth_names.h` file with the manifest constant for the privilege.**

CODE EXAMPLE 2–8 Specifying a Manifest Constant for a New Authorization in `auth_names.h`

```
AUTH_POWER = 127,  /* To leap tall MAC check failures and otherwise be irresponsible and
    unaccountable*/
```

4. **Save and close the file.**

```
:wq
```

5. **Use the Admin Editor action to open the** `/usr/lib/tsol/locale/`***locale_name***`/auth_name` **file for editing.**

6. **Create an entry with the manifest constant, name, and definition for the authorization in the** `auth_name` **file.**

Caution - Make sure that the first field has the identical manifest constant defined for the authorization in the `auth_names.h` file.

CODE EXAMPLE 2–9 Specifying a Name and a Description for a New Authorization in `auth_name`

```
AUTH_POWER:power user:Allows a user to bypass all MAC and DAC
   checks and auditing flag settings and to be otherwise totally
   unaccountable for his or her actions.
```

7. **Save and close the file.**

```
:wq
```

Extending the Trusted Solaris Privileges

The use of privileges is discussed in great detail elsewhere in the Trusted Solaris document set, so this section focuses on how to add privileges. Adding a new privilege consists of adding an entry for the privilege into these two files:

- `/usr/include/sys/tsol/priv_names.h`
- `/usr/lib/tsol/locale/C/priv_name`

`priv_names.h`

The `/usr/include/sys/tsol/priv_names.h` header file contains the manifest constants and associated numbers for privileges. Up to 128 possible privileges are allowed. As shown in Code Example 2–10, approximately eighty-six (86) `privileges` are already defined; the definitions for the default privileges number from 1 to 86 (with 0 meaning no privileges, and the numbers 28, 29, and 62 retired, as is explained below).

CODE EXAMPLE 2–10 Manifest Constants and Numbers for Default Privileges in `priv_names.h`

```
 PRIV_FILE_AUDIT = 1,  /* operational */
 PRIV_FILE_CHOWN = 2,  /* operational */
 PRIV_FILE_DAC_EXECUTE = 3, /* policy */
          .
          .
          .
 PRIV_WIN_SELECTION = 84, /* operational */
 PRIV_WIN_UPGRADE_IL = 85, /* operational */
 PRIV_WIN_UPGRADE_SL = 86, /* operational */
```

As shown in Code Example 2–11, four (4) `privileges` are reserved for Trusted Solaris extension, and these are identified by the name `tsol_reserved`. The number 28 at the top of the reserved list is the number of a retired privilege, which is a privilege that once was defined but now is not. The policy, which is explained at the top of the file, is for anyone retiring a privilege to add it to the top of the reserved list.

CODE EXAMPLE 2–11 Privilege Numbers Reserved for Trusted Solaris Use

```
/* Reserved for Trusted Solaris */

 tsol_reserved28 = 28,
 tsol_reserved29 = 29,
 tsol_reserved62 = 62,
 tsol_reserved87 = 87,
 tsol_reserved88 = 88,
 tsol_reserved89 = 89,
```

As shown in Code Example 2–12, the remaining privileges identified with the reserved name are available for any site to extend.

CODE EXAMPLE 2–12 Privileges Available for Extension

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

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

priv_name(4TSOL)

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

constant:*name*:*description*

The constant field must have exactly the same manifest constant name assigned to the privilege in the `/usr/include/sys/tsol/priv.h` file. The name field is the name of the privilege, which must be concise and descriptive for display in several user interfaces.

Note - The privilege name is displayed in the Profile Manager, File Manager, and other GUIs that display a list of privileges.

The description field is a description of the activity permitted by the privilege. The definition may be as long as you need it to be. The definition is used in the Profile Manager to guide the security administrator when assigning privileges to programs.

Code Example 2–13 gives an example of an actual 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 (\).

CODE EXAMPLE 2–13 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. **Login and assume the security administrator role.**

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **As security administrator 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 29, if needed.

3. **Read the comment at the top of the `priv_names.h` file.**

   The comment is shown in the following figure.

CODE EXAMPLE 2–14 Comment from the `priv_names.h` File

```
/ *
 * ********************** 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(3TSOL) 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.

CODE EXAMPLE 2–15 Specifying a Manifest Constant for a New Privilege in `priv_names.h`

```
PRIV_RISK = 90,
```

5. **Save and close the file.**

```
:wq
```

6. **As the security administrator, use the Admin Editor action to open the `/usr/lib/tsol/locale/`*locale_name*`/priv_name` file for editing.**

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 identical manifest constant defined for the privilege in the `priv_names.h` file.

A sample entry is below.

CODE EXAMPLE 2–16 Specifying a Name and a Description for a New Privilege in `priv_name`

```
PRIV_RISK: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
```

Working with MLDs

Multilevel directories are called MLDs, and single-level directories are called SLDs. As described in the *Trusted Solaris User's Guide*, files and directories at differing sensitivity labels in an MLD are automatically and transparently segregated within SLDs at differing sensitivity labels, while the existence of the SLDs is hidden during ordinary use.

MLDs were created to allow applications that are running at differing sensitivity labels to write into what appears to be the same directory. `/tmp` is the directory most often used by multiple applications, and for that reason, `/tmp` is an MLD in the default system. Applications are not aware that when they write a file into `/tmp`, they are actually writing the file into an SLD within `/tmp` that has the sensitivity label at which the application is running. Home directory MLDs allow accounts to create files and folders at different sensitivity labels within their home directories. When user or role accounts change into their home directories, they do not need to be aware that they have actually changed into an SLD that is at the same sensitivity label as their current workspace.

For example, when setting up a new account for user roseanne, the User Manager creates the home directory `/export/home/roseanne` as an MLD. When the user

changes to her home directory, for example by entering `cd ~` or `cd /export/home/roseanne` on the command line, she is automatically and transparently redirected to an SLD within her home directory MLD. The SLD has the same sensitivity label as her current process, so if she changes to her home directory while in a NEED_TO_KNOW workspace, she actually changes into the SLD that has the NEED_TO_KNOW sensitivity label.

The rest of this section covers the prerequisite information and describes how to work with MLDs and how to refer to an MLD as a whole so you can see and directly access any SLDs that it contains.

MLD Prefix/MLD Adornment

The MLD prefix is also called the *MLD adornment*, and, when the name of an MLD includes the *MLD prefix*, it is referred to as the *adorned name*. The adorned name of an MLD is used to directly access the top level directory where the SLDs are stored. In the default system, the MLD prefix is "`.MLD.`". For example, the adorned name for the `/tmp` MLD is `/.MLD.tmp`.

If an MLD is accessed using its *unadorned* name, the application goes to the appropriate SLD at the same sensitivity label within the MLD, while if an MLD is accessed by its *adorned* name, the application goes to the top level of the MLD and can view individually all the SLDs the MLD contains.

How SLDs Are Created

Each time a user accesses an MLD at a new sensitivity label for the first time, a new SLD at the sensitivity label of the current process is created

The sensitivity label of the first SLD created for a new user at first login is at the user's minimum sensitivity label. User roseanne's minimum sensitivity label is PUBLIC, so the sensitivity label of the first SLD created for her when she logs in for the first time is PUBLIC.

How SLDs Are Named

SLDs always have the prefix `.SLD.`, and, unlike the MLD prefix, the SLD prefix cannot be changed. SLD names are given sequential numbers as new SLDs are created within an MLD. The name of the first SLD created in an MLD is numbered 0 (.SLD.0), the second is numbered 1 (.SLD.1), and so forth.

When working in a PUBLIC workspace, if roseanne uses `pwd` on the command line or views her home directory folder in the File Manager, she sees the name of the directory as `/export/home/roseanne`, but while she apparently is working in her

home directory, she is actually in an SLD whose sensitivity label is PUBLIC and whose adorned name is `/export/home/.MLD.roseanne/.SLD.0`.

The numbers in the SLD names have no relation to the sensitivity label of the directory. For example, `/export/home/.MLD.roseanne/.SLD.0` is given the number 0 because it is the first SLD created for the user at any sensitivity label

The MLD keeps track of the label of each SLD, and each time the MLD is accessed a second and subsequent time at a particular sensitivity label, the user is transparently changed into the SLD at the correct sensitivity label.

With the first SLD created at PUBLIC and assigned the name .SLD.0, the next time roseanne changes to her home directory while running at PUBLIC, she actually changes transparently into the PUBLIC SLD, .SLD.0, and any files she creates are stored within .SLD.0. All this is transparent to the user unless the user directly refers to the MLD or SLD using the prefixes, as described under “MLD and SLD Prefixes” on page 54.

Restriction on the Creation of MLDs and Its Effects

An MLD can only be created within a directory that is not an MLD.

The effect of this restriction is that normal users cannot create MLDs in the default system because they cannot write into any directory other than their home directories and because their home directories are MLDs and they cannot create an MLD within an MLD If the system administrator creates a new directory that is not an MLD and that is writable by normal users, only then will it be possible for normal users to create MLDs.

For example, the system administrator at one site has created a directory called `/shark/doc` mounted by and writable by all developers at a single sensitivity label, so that design specifications and other project-wide documentation can be kept in one commonly-accessible place. Anyone in the development group can create new directories within that directory, and anyone can change an existing directory to an MLD. See “Creating, Changing, Finding Your Way Around In, and Deleting MLDs ” on page 55 for more about this topic.

MLD and SLD Prefixes

The adorned name of the MLD is used to access the top level directory in an MLD where the SLDs at various sensitivity labels are stored.

Using the adorned name of an MLD allows you to bypass the transparent redirection to an SLD within an MLD, lets you go to the MLD directly, and lets you view each of the single-level directories it contains. Similarly, using the adorned name of an SLD brings you directly to the SLD itself.

The following screen shows a listing of the `/tmp` MLD using the .MLD. prefix at the NEED TO KNOW ENG label, and shows the labels of the SLDs. .SLD.1 and .SLD.2 are at sensitivity labels that dominate the sensitivity label at which getlabel(1TSOL) was invoked, so the error `Not owner` displays for them.

```
trustworthy%  ls /.MLD.tmp/.*

/.MLD.tmp/.SLD.0
bt+_errlog.26629      dtbcache_:0
bt+_errlog.26630      ps_data

/.MLD.tmp/.SLD.4
bt+errlog.631         mpa000_M
mailBAAa000K4         ps_data
mpa000.E              sel_mgr.err

trustworthy% getlabel /.MLD.tmp/.*
.SLD.0 PUBLIC [PUBLIC]
.SLD.1 Not owner
.SLD.2 Not owner
.SLD.4 PUBLIC [NEED TO KNOW ENG]
```

Creating, Changing, Finding Your Way Around In, and Deleting MLDs

Anyone who can write into a directory (who has the DAC and MAC write access or has the authorization to bypass MAC when writing into a directory or has the commands required with the needed privileges) can create a new MLD or change an existing directory into an MLD.

- Create a new MLD
  - Using the File Manager as shown in “To Create an MLD from the File Manager” on page 57
  - Using the mkdir command as shown in “To Create an MLD from the Command Line” on page 58
- Change any directory to an MLD as shown in “To Create an MLD from the Command Line” on page 58

Other useful commands and options for working with MLDs are listed in Table 2–3. See the man pages for complete information.

TABLE 2–3 MLD-related Commands and What They Do

MLD-related Commands and Options	What They Do
adornfc *dirname* See adornfc(1TSOL)	Display the pathname of a directory with the final component adorned (with the MLD prefix that is set for the filesystem, which is .MLD. by default). Does exactly as described, displaying the pathname with the MLD prefix, whether or not the final component is an MLD and whether or not it is a directory. This command does not *change* anything.
getfattrflag -m *dirname* See getfattrflag(1TSOL)	Tells you whether or not the directory is an MLD.
getmldadorn *filesystem_name* See getmldadorn(1TSOL)	Displays the MLD prefix of the file system.
getsldname *filesystem_name* OR **getsldname** -s *sensitivity_label filesystem_name* See getsldname(1TSOL)	Displays the single-level directory name for the file system. Displays the SLD name that corresponds to the sensitivity label of the current process. Displays the single-level directory name for the SLD that has the specified sensitivity label file system.
mldpwd See mldpwd(1TSOL)	Displays the pathname of the working directory including the MLD prefix if it is an MLD and the SLD name, if it is an SLD.
mldrealpath *dirname* See mldrealpath(1TSOL)	Displays the pathname of the specified directory including the MLD prefix if it is an MLD and the SLD name, if it is an SLD.
rm -RM *MLD_dirname* See rmdir(1TSOL)	Recursively removes all SLDs to which it has DAC and MAC access within the specified MLD
mkdir -M *dirname* OR **mkdir** *.mld_prefix.dirname* See mkdir(1TSOL)	Make a new MLD.
setfattrflag -m existing.dirname See setfattrflag(1TSOL)	Change an existing directory to an MLD

To illustrate the relationship between the MLD and the SLD and the sensitivity label of the process, Figure 2–1 shows the output of the pwd(1), plabel(1TSOL), and mldpwd(1TSOL) commands in a terminal at the CLASSIFIED sensitivity label, when the user‘s home directory (/export/home/roseanne) is an MLD, and .SLD.2 is the

SLD created at the CLASSIFIED sensitivity label. In the following figure, the normal user command pwd shows the home directory name, while mldpwd shows that the user is actually working in .SLD.2 within the .MLD.roseanne directory. The CLASSIFIED SLD was created third, so it has an SLD number of 2: .SLD.2.

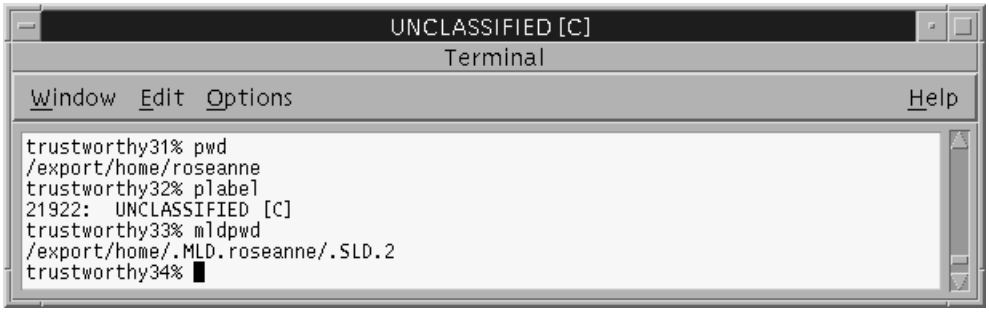


Figure 2–1 Example SLD Name for the Third SLD Created in a Home Directory

Figure 2–2 shows the output of the `pwd, plabel`, and `mldpwd` commands in a terminal at the TOP SECRET sensitivity label. The TOP SECRET SLD was created fourth, so it has an SLD number of 3: .SLD.3.

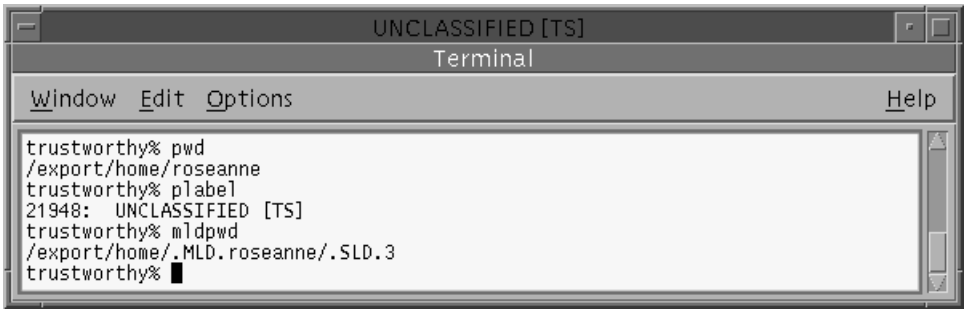


Figure 2–2 Example SLD Name for the Fourth SLD Created in a Home Directory

▼ To Find Out if a Directory is an MLD

- **As a normal user, enter getfattrflag(1TSOL) followed by the directory name (without the MLD prefix).**

```
trusted% getfattrflag -m olddir
olddir: is a multilevel directory
```

▼ To Create an MLD from the File Manager

1. **From the File Menu, select New.**

2. **Enter the name of the new directory.**

3. **Click the button "Create New Directory as an MLD."**

▼ To Create an MLD from the Command Line

♦ **Use mkdir(1TSOL) and specify the directory name with the MLD prefix.**

```
trusted% mkdir .MLD.newdir
```

OR

♦ **Use** `mkdir -M` **and specify the directory name (without the MLD prefix).**

```
trusted% mkdir -M newdir
```

OR (to change an existing directory to an MLD)

♦ **Use setfattrflag(1TSOL) and specify the name of an existing directory.**

```
trusted% setfattrflag -m oldir
```

▼ To Identify an MLD

Several methods exist to identify a multilevel directory.

♦ **Use the file(1B) command on the command line.**

```
trustworthy16% file /export/home/roseanne
/export/home/roseanne:multi-level directory
```

♦ **Use mldrealpath(1TSOL) from the command line.**

```
trustworthy15% mldrealpath /export/home/roseanne
/export/home/.MLD.roseanne
```

♦ **Use getfattrflag(1TSOL)** `-m` **and specify the name of an existing directory.**

```
$ getfattrflag -m /tmp
/tmp: is a multilevel directory
```

▼ To Identify an SLD

♦ **From the command line, use getsldname(1TSOL) followed by the name of the MLD.**

```
trustworthy16% getsldname /export/home/roseanne
.SLD.2
```

♦ **Within the MLD, use mldpwd(1TSOL) from the command line.**

▼ To Address the Entire MLD

♦ **Use ls to recursively list the MLD by specifying the MLD prefix:**

```
$  ls -R /export/home/.MLD.admin/.SLD*
```

What displays depends on the label of the current process. Referring to the entire MLD is especially useful when you want to use `tar` to copy the directory to a backup tape.

▼ To Remove an MLD

Note - Before being able to remove an MLD, you need to remove every SLD within it and any contents contained within each SLD.

---

1. **Bring up a workspace at the highest sensitivity label in your clearance.**

2. **Do one of the following to remove all the SLDs and their contents that are in your account's sensitivity label range:**

   a. **In the File Manager, enter the adorned name of the MLD in the text entry field, and choose Go To from the File Menu.**

   b. **Choose Show Hidden Objects from the View menu.**

      Drag all the SLDs from the MLD into the trash or select them and choose Put in Trash from the Selected menu.

      Figure 2–3 shows the File Manager in the `/tmp` MLD, which has been accessed directly using the adorned name, `/.MLD.tmp`, and it shows that all

the SLDs (`.SLD.1`, `.SLD.2`, and so on) are shown by choosing Show Hidden Objects from the View menu.

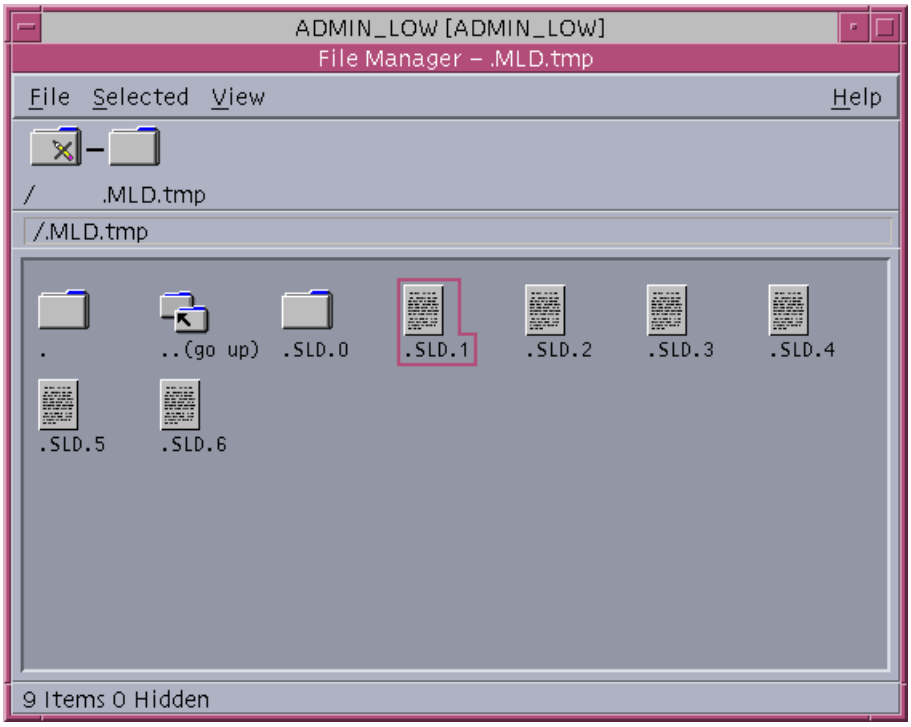


Figure 2–3 Preparing the File Manger Before Deleting an MLD

3. On the command line, use rm(1) with the `R`, `M` and `f` options to remove the MLD, all the SLDs it contains and all of their contents.

```
$ rm -RMf MLD_dirname
```

PART II Administering Users, Roles, Profiles, and Mail

This part of the *Trusted Solaris Administrator's Procedures* manual contains six chapters that provide the needed background and the procedures for managing user and role accounts, mail, and printing.

Chapter 3 includes these topics:

Chapter 4 includes these topics:

Chapter 5 includes these topics:

- “Setting Up or Modifying a User or Role Account” on page 128

Chapter 6 includes these topics:

- “Changing Mail Aliases” on page 173
- “Allowing Users to List the Entire Mail Queue” on page 176
- “Tracing Sendmail’s Activities” on page 177
- “Troubleshooting Mail Delivery Difficulties” on page 180
- “Configuring Trusted Solaris Mail Delivery Options for Mail Below Users’ Minimum Labels” on page 185
- “Substituting an Alternate Mail Application” on page 187

Chapter 7 provides a worksheet to collect data before you set up user and role accounts.

Chapter 8 includes these topics:

- “Background on Execution Profiles” on page 200
- “Use of the Profile Manager to Create or Modify Execution Profiles” on page 201

CHAPTER 3

Managing User Accounts

During initial configuration of the Trusted Solaris system, two user accounts are set up to assume the two administrative roles that work together as the "install team." This chapter describes how to finish essential tasks that are part of planning and setting up for all the remaining users on the system, which are not covered in the *Trusted Solaris Installation and Configuration* manual. This chapter also gives additional background needed for managing user accounts.

This chapter includes the following procedures:

Things to Do Before Setting Up Accounts

Set up user accounts after all these preconditions are met:

- All of the following are configured:
  - The NIS+ master
  - Any install and/or boot servers for the distributed system
  - Each user's primary workstation
  - Each user's home directory server
  - Each user's audit server(s)
  - Any other workstations and servers that provide any other services to the user's primary workstations, including the mail server.

Decisions to Make Before Setting Up User Accounts

Make the following decisions before starting, because they affect how you configure accounts.

Some decisions are similar to those you would make if installing a network of standard Solaris or other UNIX machines. However, because you are configuring the Trusted Solaris environment, you should consider these decisions in the light of any implications they might have for your site's security and of any special requirements you might have.

♦ **Decide on a convention for user login names**

The Trusted Solaris system enforces a minimum of six characters and a maximum of eight. Some organizations use first names followed by last initial; others use last names followed by first initial. Others have other conventions.

♦ **Review default groups and decide whether to add any groups**

If you need to define new groups, use the Solstice Group Manager. No new fields have been added for Trusted Solaris group administration. Refer to the *Solstice AdminSuite 2.1 User's Guide*, if needed.

Make the following decisions and gather the specified details that are specific to Trusted Solaris user administration, based on your site's security policy.

- **Choose the overall method of password generation, automatic or manual**

- **Decide how to handle mail that is sent at an SL below the recipient's minimum SL.**

  Make sure that the Trusted Solaris-specific privacy options in the `sendmail` configuration file `sendmail.cf` have values consistent with your site's security policy. See "How Sendmail Handles Mail Below the Recipient's Minimum SL" on page 185 and "To Configure Mail Delivery Options for Mail Below Users' Minimum Labels" on page 186 in Chapter 6."

- **Have the following information available**:
  - A list of the available execution profiles

    The execution profiles shipped with the system are designed so that you may never have to change or extend them. See Appendix A," for lists of which execution profiles are assigned to each role and their purposes, along with lists of all commands, actions, and authorizations assigned to each profile.

    If your site has special requirements, the security administrator can create more profiles. If other execution profiles exist at your site, see your own internal documentation for a description.
  - A list of the available administrative and non-administrative roles.

    The roles shipped with the system are designed so that you may never have to change or extend them. If your site has special requirements, more roles may be added or the roles may be reconfigured. If your site has added or modified roles, see your own internal documentation for a description of the added or modified roles available for users at your site.
  - A list with the name of each of your site's employees who need accounts, along with the responsibilities, roles, clearances and minimum labels assigned to each.

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

- **For each user account you plan to create, assign a unique user name and associated UID that is not duplicated anywhere on the network.**

- **For each group you plan to create, assign a unique GID that is not duplicated anywhere on the network.**

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

  See “Managing Startup Files in a Trusted Solaris System” on page 71.

- **Decide which files, if any, should be copied from the minimum-sensitivity-label $HOME directory SLD created for a user into subsequent SLDs, and which files should be linked, and then modify the** `.copy_files`**, and** `.link_files` **in** `/etc/skel`**.**

  “To Propagate Startup Files to Everyone’s Home Directory SLDs” on page 91.

- **Decide whether to accept or change the maximum number of bad password entries that are allowed before an account is closed.**

  See Chapter 2,” for a description of the Trusted Solaris mechanism for closing a user account after a certain number of failed passwords have been entered. “Changing the Maximum Number of Bad Password Entries” on page 41 describes how the security administrator can change the setting for the number of failed attempts that will cause an account to be closed.

How Responsibilities for Managing Users Are Divided

This section gives the background needed for managing user accounts under the following headings:

Managing Users: Divided Between Two Administrative Roles

In the default Trusted Solaris system, managing users is set up to be a two-person responsibility so that no single administrator has total control of the system. However, because more than one person may be enabled to assume any administrative role, administering users cay be more accurately thought of a *two-role* responsibility than a *two-perso*n responsibility.

The tasks of managing user accounts are divided between these two roles:

- System Administrator
- Security Administrator

Authorizations that are configured into the default execution profiles for each of these two administrative roles control which fields in the user (account) manager may be specified by each role. The authorizations that apply to the Trusted Solaris version of the Solstice User Manager are described under “Authorizations for Access to Account Management Tasks” on page 68.

System Administrator Responsibilities

The system administrator role is authorized to specify all the aspects of the user account that are not security relevant, including

- User ID
- Group ID
- Home directory location, and
- The user’s default shell

The tasks assigned to the system administrator role are the same as those performed by the system administrator (superuser or root user) in standard UNIX systems.

Security Administrator Responsibilities

The security administrator specifies all security-related aspects of the user’s account, including:

- The method of password generation
- The duration of password validity (minimum and maximum days before change is required, maximum days of inactivity, expiration date, warning)
- The user’s minimum label
- The user’s clearance
- The user’s external or internal label view

- Whether the user sees CMW labels, SLs only or ILs only
- Whether the user is audited to a greater or lesser degree than all other users in the distributed system
- Which execution profiles the user has
- Which roles the user may assume
- What action to take, if any, when the workstation is idle for a specifiable amount of time

Alternatives to Two-Role Administration

Your site may choose not to divide user administration between two roles and to reconfigure the administrative roles accordingly. See Chapter 4," under "Dividing the Tasks of Managing User and Role Accounts " on page 97.

Note - If administrative roles have been reconfigured at your site, the site's security administrator is responsible for informing administrators which responsibilities and capabilities have been assigned to each administrative role.

Authorizations for Access to Account Management Tasks

Figure 3–1 shows the User Manager Navigator (Add) menu for adding a user or role account. Because no information has yet been specified, the User Name, UID, and Comment fields are empty. The labels on the eight buttons below the Comment field indicate the types of information that is specified for each account. Each button brings up a dialog box in which the named type of account information is specified. These buttons appear on the User Manager: Navigator whether the navigator is being used to add or to modify a user account.

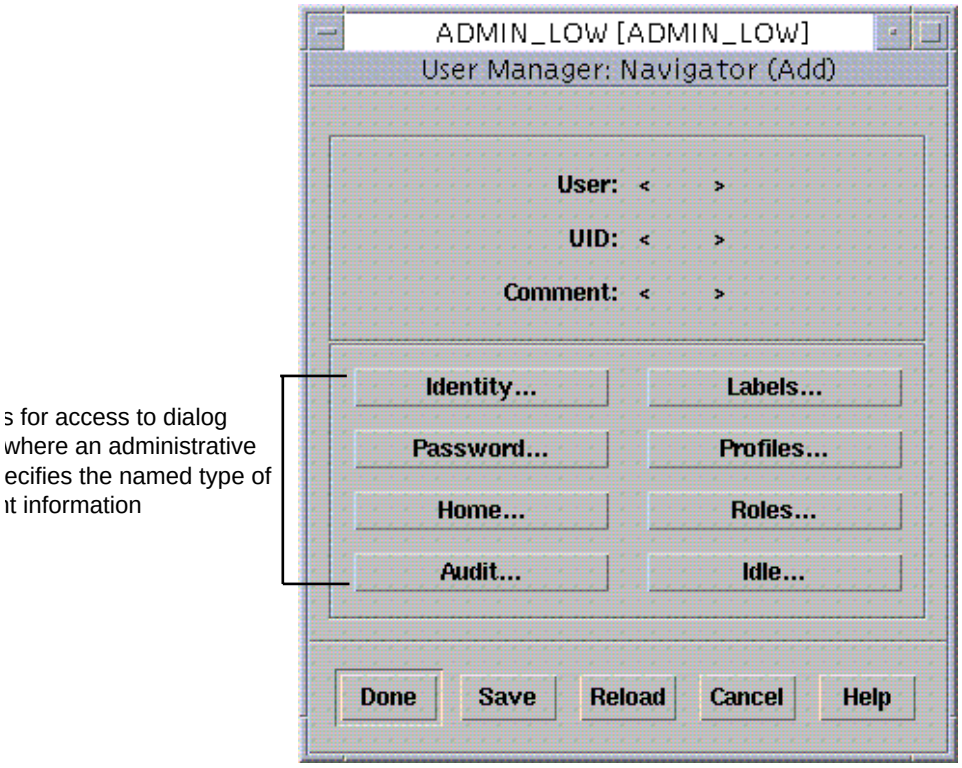


Figure 3–1 User Manager: Navigator

Each button requires an authorization. The authorizations are divided between the default system administrator and security administrator roles to maintain the separation of duties that is a requirement for two-person control of account administration.

Note - If any buttons are grayed out when an administrative role brings up the User Manager, this means that the authorization needed for that button is not in the role's execution profile.

The authorization required for each button, and the information specified on the corresponding dialog box is shown in Table 3–1.

TABLE 3–1 Authorizations for User Manager Buttons and Types of Information Specified

Button	Authorization Name	Default Role	Information
Identity	set user identity	sysadmin	set account name, UID, primary and, supplementary GIDs, comment, shell, type of account: normal, admin role, non-admin role
Password	set user password	secadmin	create a password for the account or set up account without a password
			set duration of password validity (min and max days before change required, max days of inactivity, expiration date, date for warning)
			set user's password generation method
			specify account state: open, closed, always open
			choose whether or not to update NIS+ cred table
Home	set attributes related to home directories	sysadmin	specify whether or not to create home directory at *pathname* on *server*
			specify path for shell initialization files
			set home directory permissions
			specify mail server
			choose whether or not to automount home directory
Labels	set user labels	secadmin	set clearance and minimum label
			set external or internal label view (hide or show the names of administrative labels)
			specify viewing or hiding of SLs
			specify viewing or hiding of ILs
Profiles	set user profiles	secadmin	for user accounts only: select user profiles
Roles	set list of assumable roles	secadmin	for user accounts only: select one or more roles
Idle	set idle time	sysadmin	specify action to take (if any) if screen is idle for the amount of time selected from the given list

Managing Startup Files in a Trusted Solaris System

In the Trusted Solaris system, administrators who are setting up startup 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
- The profile shell can be used to restrict an account's access to commands
- Commands run from a `.profile`(4) during workspace creation on behalf of a role are not restricted by the profile shell mechanism

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 csh(1), ksh(1), pfsh(1MTSOL), or sh(1) shells.

The word *source* is often used as a verb to mean *to read in* or *to execute* the commands in a startup file. (The `csh` even has a built-in source command for executing the commands in dot scripts.) One set of startup files is sourced by the window system. 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.

TABLE 3–2 Startup Files Read by the Window System for Each Type of Login Shell

Login Shell	Startup File
C shell	`/etc/.login` and `$HOME/.login`
Bourne shell or Korn shell	`/etc/.profile` and `$HOME/.profile`
Profile shell (only in Trusted Solaris systems)	`/etc/.profile` and `$HOME/.profile`

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 74).

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(1) pages for login(1TSOL)and profile(4). The exception is that when an account has the Trusted Solaris profile shell, pfsh(1MTSOL), 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 73.

`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, if the file exists, or
- The default `/usr/dt/config/sys.dtprofile`

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

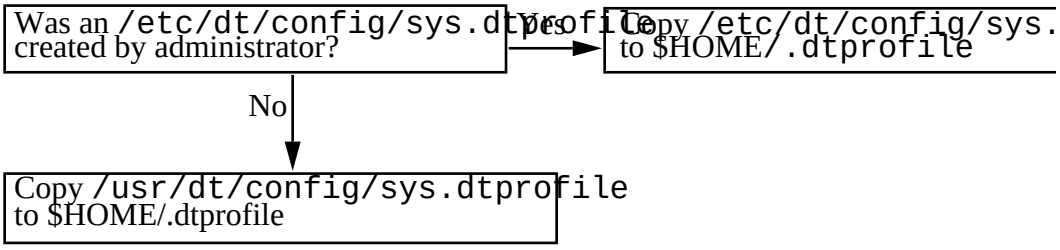


Figure 3–2 How `$HOME/.dtprofile` is installed

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

CODE EXAMPLE 3–1 Default Setting in the`/usr/dt/config/sys.dtprofile`

```
# DTSOURCEPROFILE=true
```

Removing the # before the DTSOURCEPROFILE definition in any of the copies of the `/usr/dt/config/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 `*.dtprofile` file encourage the site's administrator or individual users to make sure that the startup files do not do the following:

- Do not do anything that requires a terminal emulator
- Do not do 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 89, 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 for an account with a profile shell as its login shell is different from the algorithm that applies to the other shells, for security reasons. When a user's login shell is specified as the Profile shell, the `.profile` file is read during login before the Profile shell is in effect, but the version of the `.dtprofile` file in the account's $HOME directory does not get sourced. Even if the user whose default shell is `pfsh` creates a $HOME`/.dtprofile`, the file has no effect because the personal `.dtprofile` is never looked at. In this way, individual users or roles are prevented from specifying commands in the `.dtprofile` that are not specified in the account's profile.Either the default `/usr/dt/config/sys.dtprofile` or a version modified by the security administrator in `/etc/dt/config/sys.dtprofile` is used instead of $HOME`/.dtprofile` file.

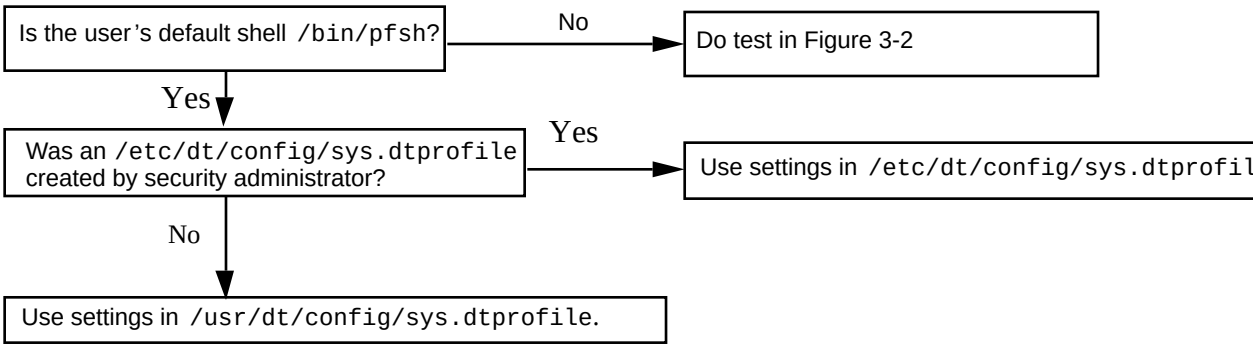


Figure 3–3 How `$HOME/.dtprofile` is Bypassed for Users with a Default Shell of pfsh(1MTSOL)

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 3–3 Startup Files Read at Shell Initialization

Shell	Startup File
C shell	`$HOME/.cshrc`
	`$HOME/.login`
Bourne shell	`$HOME/.profile`
Korn shell	`$HOME/.profile`
	file specified with ENV variable
Profile shell (only in Trusted Solaris systems)	`$HOME/.profile`

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 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 90.The same entry must be in a file named .Xdefaults-*hostname* in the home directory SLD at every sensitivity label at which the account works. To set up the copying or linking of .Xdefaults-*hostname* into all SLDs, see “Using .copy_files and .link_files” on page 79.

Note - The default .profile file for all roles has a function to alias the adminvi(1MTSOL) to vi(1), but the alias 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 104.

Other Shell Startup Files

One startup file that is useful to have in a home directory is .mailrc, which is often used to specify the user’s Mail folder, inbox and mail aliases, among other things, as shown in the following figure.

CODE EXAMPLE 3–2 .mailrc Example

```
set folder=/home/roseanne/Mailcd
set MBOX=$HOME/Mail/inbox
alias pubs janer@think monicap@owl jstearns@auburn
```

To give another example, the .newsrc file is consulted to determine which news groups to bring up whenever a user brings up a news viewer. The .Xdefaults and .Xdefaults-*hostname* files are also frequently modified to control the behavior of windows.

Administering Skeleton Directories

The system administrator role defines the Skeleton Path directory for an account in the User Manager Home dialog box, shown in the following figure.

Figure 3–4 User Manager: Home Directory Dialog Box

As shown in Figure 3–4, the default Skeleton Path directory is `/etc/skel`. If the system administrator accepts the default, certain default 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, which are shown in the `/etc/skel` directory listing in the following figure, are copied for normal user accounts.

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

```
trusted1% cd /etc/skel
trusted2% 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 77.)

The /etc/skel files are copied into the SLD that corresponds to an account's minimum sensitivity label when the account's home directory is created. The home directory is created when the administrator role creates an account using the User Manager.

Role Startup Files in /etc/skel/tsol

Home directories for roles are created in /etc/security/tsol/*rolename,* as shown in the following figure.

```
trusted1% cd /etc/security/tsol
admin/    install/    /oper     /secadmin
```

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.

```
trusted1% cd /etc/skel/tsol
trusted2% ls
role.link_files role.profile
```

When setting up a new administrative role, the security administrator should specify the /etc/skel/tsol in the Skeleton Path field in the User Manager.

Changing Skeleton Files

The system administrator role may:

- 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
- Create an alternate skeleton directory that contains site-specific skeleton files

  If using an alternate skeleton directory location, the system administrator can specify the pathname of the alternate directory in the HOME dialog box when setting up a user's account using the User Manager.
- Set up master copies of the shell startup files in one skeleton subdirectory for each shell and then specify in the User Manager the appropriate skeleton directory for each user's default shell.

  In this way, only the correct startup files for the user's default (login) $SHELL would be copied from the skeleton file directory into the SLD the first time a workspace is created at the account's minimum sensitivity label. See "To Separate the Shell Initialization Files for Each Shell " on page 91 in this chapter for how to do separate setup for each shell, including the pfsh.

The copying of files from /etc/skel (or whatever skeleton path is used) is done as it would be in other UNIX systems, but the fact that almost all home directories are MLDs complicates the process. In the Trusted Solaris system, files are copied from

`/etc/skel` *only* into the SLD created on the account's behalf at the account's minimum label.

After the workspace is created at the account's minimum sensitivity label, the account is responsible for renaming any file or files copied from the skeleton directory that are appropriate to his or her default shell and modifying the files to suit the account's needs. For example, if a user's default shell is a C shell and minimum sensitivity label is PUBLIC, the first time the account goes to a workspace whose sensitivity label is PUBLIC, the user would rename the `local.cshrc` automatically copied from `/etc/skel` to `.cshrc`, and modify the `.cshrc` as desired.

Because the initialization files are only copied from the skeleton directory into the home directory SLD created at the account's minimum sensitivity label, more work needs to be done to propagate these or any other files to any SLDs that may be created at other labels. For any initialization files to be copied or linked into other home directory SLDs created at other labels, either the user or administrator needs to create either a `.copy_files` or `.link_files` or both, as described in "Using `.copy_files` and `.link_files`" on page 79.

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, Solstice AdminSuite), the MANPATH environment variable should include all the directories shown in the following table.

TABLE 3–4 Man Directories for Trusted Solaris Bundled Products

Man Directory
`/usr/man`
`/usr/openwin/man`
`/usr/dt/man`

Note - Solstice AdminSuite man pages do not have a separate man directory but are included in `/usr/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` (or alternate skeleton directories) 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 account's minimum-sensitivity-label SLD. 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 minimum-sensitivity-label 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.

Whenever a workspace is created at a new label, dtsession(1) runs the updatehome(1MTSOL) 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(1MTSOL) command consults the copy and link-control files and performs the actions shown in the following table:

TABLE 3–5 What updatehome Does and When

When	Action
During a login session whenever a new workspace is created at a new sensitivity label	Copies the files in copy_files and links the files in link_files from the account's minimum sensitivity label SLD to the SLD at the new label.
At the start of a login session	Copies the files in copy_files and links the files in link_files from the account's minimum sensitivity label SLD to all of the existing home directory SLDs.

If `.copy_files` is Used to Copy Files Between SLDs:

After a file is copied, the copy can be modified to be different in each SLD, if desired. After any file is copied from the minimum sensitivity label SLD into a subsequently-created SLD at another sensitivity 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 91.

Following the steps in the procedures mentioned above, the system administrator role creates a skeleton directory (or uses the default `/etc/skel`) and puts into it 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 system administrator 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.

If `.link_files` is Used to Link Files Between SLDs:

A change made to one initialization file in one SLD is made to all the files at all labels in all SLDs to which the file is linked.

So, for example, if a change is made to the linked `.cshrc` file in a Confidential workspace, the change would apply to all the `.cshrc` files in all other SLDs at all other labels where the file has been linked. See “To Propagate Startup Files to Everyone’s Home Directory SLDs” on page 91.

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 3–6 Planning Worksheet for Copying and Linking Startup Files Between SLDs

Common Startup Files	**List to be Copied** **(for** `.copy_files`**)**	**List to be Linked** **(for** `.link_files`**)**
`.bugtraqrc`		
`.cshrc`		
`.dtprofile`		
`.login`		
.Xdefaults		
.Xdefaults-*hostname*		
`.mailrc`		
`.newsrc`		
`.profile`		

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

cron(1MTSOL) is the clock daemon that executes commands at specific times. This section gives a brief overview of what `cron` does and describes how administering `cron` and its associated commands is different in the Trusted Solaris system. See the Solaris 2.x *System Administration Guide, Vol. II* for basic `cron` information. For Trusted Solaris modifications see also the modified man(1) pages for at(1TSOL), atq(1TSOL), atrm(1TSOL), cron(1MTSOL), and crontab(1TSOL).

Background

`cron` maintains an internal time-ordered list of events for all scheduled jobs. Each event represents a job with the information necessary to execute it. `cron` runs two types of jobs:

- Single execution jobs (at_jobs) scheduled by `at` to be executed once at a specified time in the future
- Single execution jobs (at_jobs) scheduled by `batch`, which is a front-end script to `at(1)` that submits jobs to be executed right away, as soon as the system load level permits
- Periodic execution jobs (cron_jobs) scheduled by `crontab`, to be executed repetitively at specified intervals

  cron_jobs and at_jobs are scheduled by `cron` from reading `crontab` and `atjob` files in their respective spool directories only at the following times:

- During `cron`'s own process initialization or
- After change is made to a `crontab` or an `atjob` file.

`crontab` Files

The `crontab` file is generated by a user or role account using the crontab(1TSOL) command (which, in the Trusted Solaris system, must be in one of the account's execution profiles). A `crontab` file consists of commands, one per line, that execute automatically at the time specified at the beginning of each command line. Each command line is referred to as a *cron_job*. A `crontab` file may contain multiple cron_jobs. The spool directory for the `crontab` files is `/var/spool/cron/crontabs`.

`atjob` Files

The `atjob` file is generated by a user or role account using the at(1TSOL) or batch command (either of which, to be used in the Trusted Solaris system, must be in one of the account's execution profiles). The user's current process environment when the command is executed is saved as part of the file. Each file is referred to as an *at_job*. The spool directory for the `atjob` files is `/var/spool/cron/atjobs`.

Supporting Jobs at Multiple Labels in the Spool Directories

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 sensitivity labels within the `atjobs` directory.

Determining Whether the Profile Shell is Used by a Job

Caution - If the profile shell, pfsh(1MTSOL), is specified to execute a job, the security administrator 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 `pfsh` if either of the following is true:

- The login shell in the account's passwd(4) entry is the `pfsh` or
- The $SHELL environment variable is set to `/bin/pfsh`

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

  Because a user can use `at` with the -c (for `csh`), -k (for `ksh`), -s (for `sh`), or `-p` (for `pfsh`) options to specify the shell with which the job should be run, 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 login shell in the account's `passwd` entry is the `pfsh` or
- The $SHELL environment variable is the `pfsh` 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 no matter who executes the command is not usually consistent with a site's security policy, so the security administrator 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 execution profiles using the Profile Manager and
- Specify that the job is executed with the profile shell, as described in "Determining Whether the Profile Shell is Used by a Job" on page 83

  See "How Privileges Are Assigned to Commands and Actions" on page 504 and "To Give Forced Privileges to a Command" on page 533, or "Giving Inheritable Privileges to a Command or Action" on page 506 for more information. See also, "To Write a Profile Shell Script that Runs Privileged Commands " on page 534, which uses a cronjob in the example.

Using a UNIX Domain Socket for Communications

The communication mechanism between crontab(1TSOL), at(1TSOL), atrm(1TSOL) and cron(1MTSOL) in the Trusted Solaris system is a UNIX domain socket. Using the UNIX domain socket, the trusted networking interfaces provided by the Trusted Security Information Exchange (TSIX) library allow `cron`'s clients to communicate their message and security attributes to `cron` in a trusted way. See also the man(1) page for libt6(3NTSOL).

cron(1MTSOL) 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.

The clock daemon turns on the net_mac_read privilege in its effective set to make a multilevel port, so it can receive messages at different sensitivity labels.

When a user creates, modifies, or removes a `crontab` or `atjob` file, the `crontab`, `at`, or `atrm` command sends a message (filled in with the appropriate data) along with the command's processes' sensitivity label to notify the clock daemon. Upon receipt of a message and the associated sensitivity label, the clock daemon interprets the message and performs the necessary operations. The submitting processes' sensitivity label is used to create or remove the `crontab` or `atjob` file at the corresponding SLD.

Ancillary Files

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.:

Access to `at` and `cron`

Administrators can control access to `at` and `cron` by means of `*.allow` and `*.deny` files in the `/etc/cron.d` directory.

The two files that define the users permitted or forbidden to use `at` are:

- `at.allow`
- `at.deny`

The two files that define the users permitted or forbidden to use `cron` are:

- `cron.allow`
- `cron.deny`

If the `*.allow` file exists, then the `*.deny` file is not checked to determine the user's access. If no `*.allow` file exists, the `*.deny` file is checked, and if it exists but is empty then anyone is permitted to submit jobs. In Trusted Solaris, if neither an `allow` or `deny` file exists, no user is allowed to submit a job.

The Trusted Solaris system is delivered without `cron.allow` and `at.allow` files. The default `cron.deny` and `at.deny` files contain the following account names:

```
daemon
   bin
   smtp
   nuucp
   listen
   nobody
   noaccess
```

Allowing Access to Jobs Owned by Others

The default Trusted Solaris security policy does not allow users to access jobs owned by other users, but the security administrator can configure users to bypass this restriction. To allow certain users to access jobs belonging to other users, the security administrator can use both of the following together:

- The `at.admin` and `cron.admin` files in `/etc/cron.d`
- Assigning `at` and `cron`-related authorizations

at.admin and cron.admin Files

As shipped, the at.admin and cron.admin files in /etc/cron.d contain the names of special system accounts that are used by system processes. The default list, which is the same in both files, is as shown here:

```
bin
adm
lp
smtp
uucp
nuucp
listen
```

As shipped, the default crontabs ADMIN_LOW SLD contains crontabs for adm, sys, and uucp. There are no other default crontabs in the crontabs MLD or default at_jobs in the atjobs MLD. Any site can add additional crontabs and can also create at_jobs to run on the behalf of any of the special system accounts. Because no one logs into those accounts, which have no passwords assigned, creation of or any kind of modification to cron_jobs or at_jobs run on behalf of these accounts would otherwise not be possible without this feature.

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. account has the *modify at admin* authorization in an execution profile.
4. If neither of 1 or 2 is true, the invoking account must have the *modify at user* 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 *modify cron admin* authorization.
4. If neither of 1 or 2 is true, the invoking account must have the *modify cron user* authorization in an execution profile.

Changes to `crontab`(1TSOL)

The Trusted Solaris modifications to the crontab(1TSOL) command are described in the following table.

TABLE 3–7 crontab(1TSOL) Options

Option	Comments
e	Create or modify a `crontab` file at the sensitivity label that matches the invoking processes' sensitivity label. A user can edit another user's crontab file only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85 are met. If the user's `passwd` entry does not specify `/bin/pfsh`, the `crontab -e` command invokes the editor defined by the VISUAL environment variable. If VISUAL is null, the editor defined in the EDITOR environment variable is used; if neither is defined, `cron` uses the default editor `ed`(1). When the user's `passwd` entry specifies `/bin/pfsh`, if the environment variable is set to `vi`, then `adminvi` is used. If the environment variable is set to be `dtpad`, then the `TSOLdtpad` editor is used. If neither variable is set, `cron` uses the default editor `adminvi`.
l	Display the `crontab` file at the invoking processes' sensitivity label for the current user. A user can display another user's `crontab` file only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85 are met.
r	Remove a user's `crontab` file at the invoking processes' sensitivity label from the `crontabs` directory. A user can remove another user's `crontab` file only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85 are met.

The access control remains the same except that when neither `cron.allow` nor `cron.deny` exists, no user is allowed to submit a job.

Changes to the `at` CommandThe following table shows modified standard at(1TSOL) options and the new `-p` option for `at`.

TABLE 3–8 Trusted Solaris 2.5 `at`(1) options

Option	Comments
l	Display information about all at_jobs owned by the current user at the invoking processes' sensitivity label, if no at_job number operands are specified. If at_job number operands are specified, report information only for those jobs. If the at-job is not owned by the current user, display its job information only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85 are met.
r	Remove the at_jobs with the specified at_job number operands that were previously scheduled. If the specified at-job is not owned by the current user, it is removed only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85are met.
p	The profile shell is used to executed the job.

The access control remains the same except that when neither `at.allow` nor `at.deny` exists, no user is allowed to submit a job.

Changes to the `atq` Command The following table shows atq(1TSOL) changes.

TABLE 3–9 Trusted Solaris 2.5 `atq`(1) Changes

Argument	Comments
none	Display at_jobs owned by the invoking user at the invoking processes' sensitivity label. Other user's at_jobs are also displayed only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85are met.
user	Display all at-jobs owned by the specified *user*, if *user* is the invoking user. If *user* is not the invoking user, display jobs only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85 are met.

Changes to the `atrm` CommandThe following table shows atrm(1TSOL) changes.

TABLE 3–10 Trusted Solaris 2.5 `atrm`(1) Changes

Argument	Comments
user	Remove all at-jobs owned by the specified *user* at the invoking processes' sensitivity label, if *user* is the invoking user. If *user* is not the invoking user, remove jobs only if the conditions described in "Allowing Access to Jobs Owned by Others" on page 85 are met.
at_job #	Remove at_jobs with the specified at_job number operands at the invoking processes' sensitivity label.
option **a**	Remove all at-jobs owned by the invoking user at the invoking processes' sensitivity label.

Miscellaneous

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

Trusted Solaris 2.5.1 is delivered with the following crontab files:

- At the ADMIN_LOW 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 clock daemon at ADMIN_HIGH sensitivity label. The clock daemon logs its internal messages in this log file.

User Setup Procedures

▼ 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 Login and Assume an Administrative Role” on page 15 if needed.

2. **Use the file manager or commands in a terminal emulator to copy** `sys.dtprofile` **from** `/usr/dt/config` **to** `/etc/dt/config`**.**

   Create the destination directory if it does not already exist.

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

3. **Use the Admin Editor action to open the** `sys.dtprofile` **file for editing.**

   See “To Use the Admin Editor Action to Edit a File” on page 29, 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 sensitivity label and then list the `.Xdefaults-`*hostname* in either `.copy_file`s or `.link_files`, as described in “Using `.copy_files` and `.link_files`” on page 79. See also `updatehome`(1MTSOL).

1. **Go to your home directory.**

```
trusted4% 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 *User Accounts, Printers and Main Administration* manual for Solaris 2.5.**

The procedure tells you how to create three shell-specific skeleton directory names that you will enter in the Skeleton path field in the User Account 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.

- **As admin, create a `skelP` subdirectory (for a specialized version of the `.profile` to be installed into the home directories of users whose default shell is `pfsh`).**

- **As admin, enter the correct `skel`*X* 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.

---

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

   See "To Login and Assume an Administrative Role" on page 15 if needed.

2. **Go to `/etc/skel` or whatever skeleton directory location you wish to use.**

Code Example 3–4 shows a skeleton directory created for startup files for the C shell, `/etc/skel/skelC` (according to the procedure “To Separate the Shell Initialization Files for Each Shell ” on page 91).

CODE EXAMPLE 3–4 Changing to a Skeleton Directory Created for C Shell Startup Files

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

3. **Put generic copies of startup files into the skeleton directory.**

   Code Example 3–5 shows startup files for the C shell, for the mailer and for `dtterm`

CODE EXAMPLE 3–5 Startup Files in `/etc/skel/skelC`

```
$ ls
   .cshrc
   .login
   .mailrc
   .Xdefaults
   .Xdefaults-hostname
```

4. **Create a `.copy_files` or modify it, if it already exists, to list any files you want to have copied to all home directory SLDs.**

   Remember that copied files may be changed by users to be different in each SLD.

5. **Create a `.link_files` or modify it, if it already exists, to list any files you want to have linked to all home directory SLDs.**

   Remember that all linked files are identical in each SLD.

6. **When setting up user accounts, enter the pathname of the modified skeleton directory in the Home dialog box on the User Manager.**

CHAPTER 4

Managing Roles

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

Differences Between Role Accounts and User Accounts

Role accounts have all the characteristics of user accounts, with the following exceptions:

- The system does not allow role account to log in.

  Administrative work must be done only by known users who have previously identified themselves to the system and who have been authenticated. This requirement is based on the need for all users and especially for administrators to be accountable for their actions. Administrative actions performed by a user whose identity is not known are not attributable through the auditing mechanism.
- Instead of logging into a role, already-identified and authenticated users assume a role from the Trusted Path menu.
- A role cannot assume another role.

  If you identify an account as a role account in the User Manager Identity dialog box, the Role button dims on the Navigator dialog box and if you then click on the Role button you get an error message.
- Each role has a role workspace that gets added to the front panel when the user assumes the role.

Differences Between Administrative and Non-Administrative Role Accounts

Two types of roles are available:

- Non-administrative
- Administrative

Non-administrative Roles

Only one non-administrative role exists in the default system:

- System Operator (oper)

The operator role exists to do backups and manage printers, and none of its functions requires the Trusted Path Attribute or any of the other special characteristics described under “Administrative Roles” on page 95.”

When to Create a Non-administrative Role

When deciding whether to create a non-administrative role compared to simply creating a new non-role execution profile and assigning it to users, you should think about the advantages of creating the non-administrative role described here.

Non-administrative roles are able to share ownership of and access to files between multiple users, and all users who can assume the same role are able to share the same environment in a single shared workspace:

- Any files created by the role are owned by the role, so any employee allowed to assume the role can have the same access to the files as the creator.
- Everyone who assumes the role gets the same home directory and environment set up in the role workspace.

For example, you can set up several employees who are able to assume the operator role (account name *oper*) on various shifts. Each user acting in the operator role has the same home directory ($HOME/`oper`), and can use the same tools. Any files created by the role are owned by `oper`.

Administrative Roles

Two main administrative roles in the default Trusted Solaris system are:

- Security administrator (secadmin)
- System administrator (admin)

A third optional administrative role, root, exists for installing software or doing other actions where a real UID of 0 (root's UID) is required. The root role should ordinarily be assigned to the same account that has the secadmin role, because while root's capabilities are strictly limited, the root role's main function of adding software is security-sensitive

Administrative roles have the following unique characteristics:

- An administrative role is automatically assigned the sysadmin group 14 as one of its supplementary groups. Without this group ID the role cannot run administration tools.
- The administrative role workspaces have the Trusted Path Attribute that is also required for using the administration tools.

When to Create a New Administrative Role

Sites may create a new administrative role if they have a need for an additional role to use commands or applications that check for the Trusted Path Attribute. A site might create a new administrative role, for example, if it wanted to combine the responsibilities of the three default roles into one. See "Dividing the Tasks of Managing User and Role Accounts " on page 97.

Things that Need the Trusted Path Attribute

Table 4–1 shows the commands and applications that require the Trusted Path Attribute, and which therefore must be run in an administrative role's workspace.

TABLE 4–1 Commands and Applications Requiring the Trusted Path Attribute

audit(1MTSOL)
auditd(1MTSOL)
login(1TSOL) `-d` and `-f` options
lpsched(1MTSOL)
runpd(1MTSOL)
sendmail(1MTSOL) options `-ba`, `-bd`, `-bi`, `-bs`, `-bt`, `-bv`, `-M`, `-X`, `-d`, and `- q`
`Solstice AdminSuite tools:`
`Database Manager dbmgr`
`Group Manager groupmgr`
`Host Manager hostmgr`
`Printer Manager printmgr`
Profile Manager `prfmgr`
`Serial Manager serialmgr`
`User Manager usermgr`

For example, the Solstice AdminSuite tools, which are used for most administration tasks, check for the Trusted Path Attribute, so if the security administrator creates a new role that needs to access any of the Solstice AdminSuite tools listed in Table 4–1, it would need to be created as an administrative role.

Dividing the Tasks of Managing User and Role Accounts

As described in Chapter 3," in the default Trusted Solaris system, the security administrator and system administrator roles each configure a prescribed list of characteristics for user accounts. As illustrated in Figure 4–1, the security administrator role is also responsible for configuring the security relevant characteristics of normal user accounts and the accounts for all other roles, while, because no role can configure itself, the system administrator is responsible for configuring the same characteristics of the security administrator role as he or she configures for user accounts.

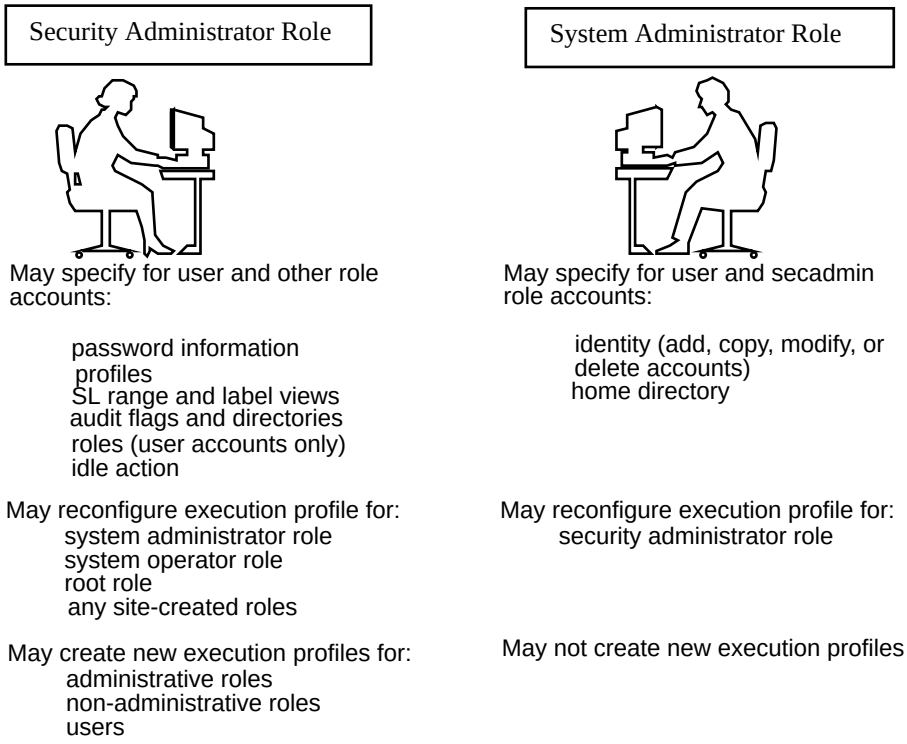


Figure 4–1 Division of Account and Profile Configuration Responsibilities Between Security Administrator and System Administrator

Authorizations for Access to Account Management Tasks

Figure 4–2 shows the User Manager Navigator (Add) menu for adding a user or role account. Because no information has yet been specified, the Account Name, UID, and Comment fields are empty. The labels on the eight buttons below the Comment field indicate the types of information that is specified for each account. Each button brings up a dialog box in which the named type of account information is specified. These buttons appear on the User Manager: Navigator whether the navigator is being used to add or to modify a user account.

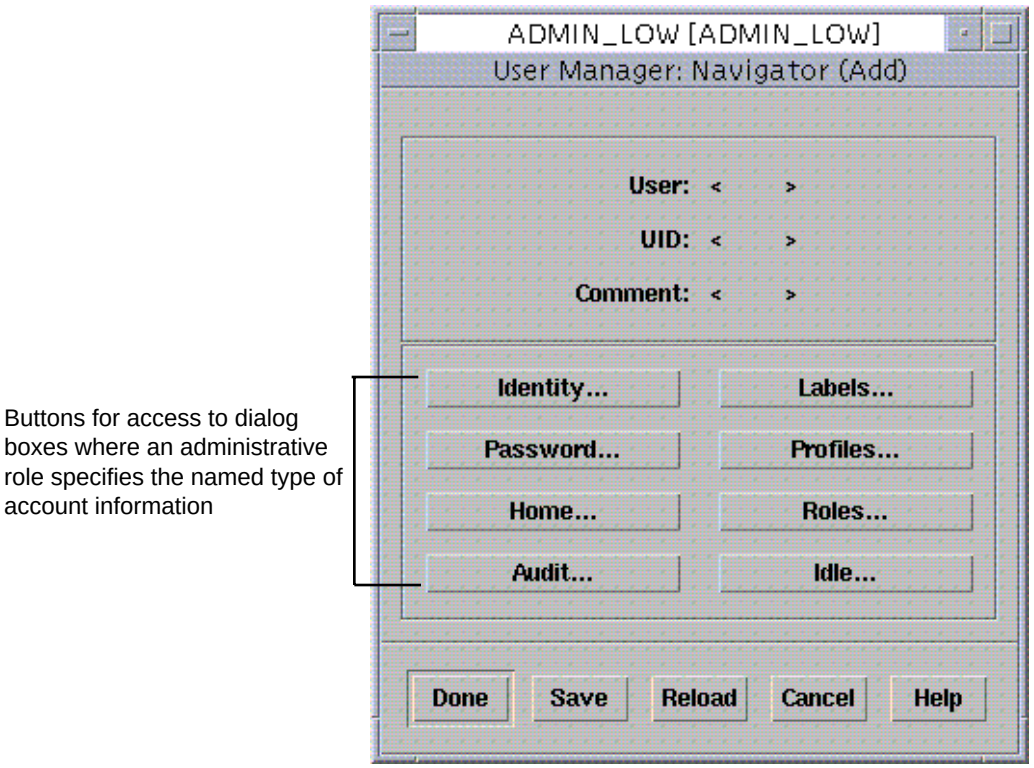


Figure 4–2 User Manager: Navigator

Each button requires an authorization. The authorizations are divided between the default system administrator and security administrator roles to maintain the separation of duties that is a requirement for two-person control of account administration.

Note - If any buttons are grayed out when an administrative role brings up the User Manager, this means that the authorization needed for that button is not in the role's execution profile.

The authorization required for each button, and the information that can be specified on the corresponding dialog box is shown in Table 4–2. See also the auth_desc(4TSOL) man page.

TABLE 4–2 Authorizations For Specifying Types of User Information

User Manager Button	Authorization Name	Number	Scope of Authorization
Identity	set user identity	17	Allows an administrator to set security information related to the account's identity via the User Manager: the user name, primary group, secondary groups, comment, and login. Needed to add, copy, or delete a user.
Password	set user password	18	Allows an administrator to set password information pertaining to an account via the User Manager, including the password, type of password, life time, expiration date, warning days, and the permission to set up the credentials table.
Home	set attributes related home directories	25	Allows an administrator to determine such things as location, permissions, and initial contents of a user's home directory via the User Manager.
Labels	set user labels	20	Allows an administrator to set various label-related pieces information associated with a particular user via the User Manager: the user's minimum login label, clearance, label view, and label translation attributes.
Profiles	set user profiles	22	Allows an administrator to assign profiles to a user via the User Manager.
Roles	set list of assumable roles	24	Allows an administrator to select via the User Manager which roles a user may assume.
Idle	set idle time	23	Allows an administrator to set the idle time and determine which command to execute when a workstation has been idle for the specified idle time via the User Manager.

Note - It is important to understand that the default authorizations shown in Table 4–2 allow the role to modify the appropriate information for every other user and role account except that of the role itself. For example, the set user password authorization allows the security administrator to configure password options for all other accounts except that of the security administrator role.

Authorization for Specifying Information for One's Own Role

The *permit self-modification* authorization allows a role who has that authorization to specify the authorized information for the role's own account.

By default, neither of the administrative role profiles have this authorization.

The permit self-modification authorization allows a role only to partially bypass the restriction on modifying one's own role. This authorization therefore actually allows the role to configure for itself only those areas where the role already has the required authorizations. For example, if the system administrator adds the permit self-modification authorization to the execution profile for a security administrator—who has the authorizations that give access to the security relevant dialog boxes in the User Manager, the security administrator role could then only specify the security-relevant aspects of its own account, and still would not be able to enter information in any other fields in the dialog boxes for which the role was not authorized, such as those on the Identity dialog box.

Note - The permit self-modification authorization does not allow a role to edit any profiles assigned to that role account.

Alternatives to Two-Role Administration

Your site may choose not to use two role administration and may decide to create a single combined administrative role to do administrative tasks, if your site's security policy allows.

Note - Do not attempt to combine the capabilities of secadmin and sysadmin in the root role because the root role cannot be added to the NIS+ admin group, and the root role cannot run Solstice from any other host than the NIS+ master. As a result, a combined root role would not be able to run any of the Solstice administration tools unless directly logged into a the NIS+ master.

Note - If your site wishes to have a single combined administrative role, make sure that the combined role includes all authorizations that are in both role's profiles along with the authorization *permit self modification*, which is described in "Authorization for Specifying Information for One's Own Role" on page 100. You may also wish to add the commands, authorizations, and actions assigned to the system operator role.

Creating a New Role

In Trusted Solaris 2.5 system, each role must have its own individual account, so the system administrator and security administrator must set up an account for a new role. The security administrator may also need to create a new profile for the role.

Before creating a new profile for a role, the security administrator should analyze whether any of the commands required by the role need privileges in order to do the tasks the role is assigned to do. See Chapter 16," and priv_desc(4TSOL). See the man pages for guidance about how to determine *required* privileges and *override* privileges that may be required by the command.

Required Privileges

When a man page that has not been modified for the Trusted Solaris system states that super-user is required to execute a certain command or option, remember that one or more privileges are required instead. When a command or one of its options needs a privilege in order to succeed, that privilege is a *required* privilege, which is sometimes called an *operational* privilege. Required privileges are indicated on a Trusted Solaris man page with the words "must have" as shown in this sentence: "The ifconfig(1MTSOL) command must have the sys_net_config privilege to modify network interfaces." If the required privilege is not given to the command in a user's execution profile by the security administrator, the command will not work.

Override Privileges

Override privileges are needed for a Trusted Solaris command that is designed to work within security policy when the command fails because certain DAC or MAC checks are not passed. Like required privileges, override privileges may be assigned at the security administrator's discretion, after making sure that the override privileges do not give the user powers beyond his or her level of trust or allow actions beyond the scope of the task for which the profile is designed. On man pages, the names of privileges that may be used to override access restrictions are given in the ERRORS section. The override privileges that may be given to bypass DAC or MAC restrictions on files or directories are given below.

DAC Override Privileges

The DAC override privileges are file_dac_read and file_dac_write. If a user does not have DAC access to a file, the security administrator may assign one or both of these privileges to a command, depending on whether read or write access or both are desired.

MAC Override Privileges

The MAC override privileges are file_mac_read and file_mac_write. If a user doesn't have MAC access to a file, the security administrator may assign one or both of these privileges to the command, depending on whether read or write access or both are desired.

Options for Avoiding the Need for Privilege

Besides being able to assign an override privilege, the security administrator has other options. For example, to avoid the use of privilege the security administrator may specify that the command executes with another user's ID (usually the root ID 0) or group ID, one that allows access to the file or directory based on its permissions or its ACL.

Also, when configuring an administrative role, the security administrator needs to decide whether a command needs to run at a certain label.

Verifying the Use of Security Attributes Within Security Policy

After assessing which privileges and other security attributes a command needs in order to do its work, the security administrator needs to ascertain whether the command and the role may be trusted to use the privileges and other security attributes without violating the site's security policy.

Example: Using the Man Page when Configuring `mount` in a Profile

The mount(1MTSOL) command is specified in the System Maintenance profile assigned to the system administrator role. It is a fairly-simple example of how specifying security attributes is done. In Solaris and other UNIX operating systems, `mount` without any options displays the name of the mounted filesystems to an ordinary user. However, when run as root, `mount` may be used by the administrator (super-user) to mount filesystems from the command line or to mount filesystems specified in the vfstab(4TSOL) table. The mount(1MTSOL) command checks for UID=0.

Customizing the Execution Profiles for the Default Roles

The four custom role execution profiles, which are provided for customizing the execution profiles for roles, are shown in the following table.

TABLE 4–3 Custom Role Profiles

Custom Admin Role
Custom Root Role
Custom Secadmin Role
Custom Oper Role

Using these profiles ensures that each site always has a record of what changes have been made to the role profiles, which can be especially helpful when debugging problems or making service requests. The following constraints apply to which role can modify which profile:

- The security administrator role can customize the custom role profiles for every role except the security administrator role.
- Only the root role can customize the security administrator role.

  If compatible with the site’s security policy, the root role can add the *permit self modification* authorization to the Custom Secadmin Role profile, so the security administrator can customize the security administrator role.

See “To Customize an Administrative Role” on page 244<![%XRefs; [“To Customize an Administrative Role” on page 254]]>.

▼ To Configure a New Role

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

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

3. **Decide if the role needs to have a new execution profile, and if so, create one for it.**

   How to use the Profile Manager to create a new role profile is described in Chapter 8<![%XRefs; [Chapter 8, “Managing Execution Profiles for Users and Roles]]>.” See Appendix A<![%XRefs; [Appendix A, “Profile Summary Tables]]>” for lists of the existing profiles and their commands, actions, and authorizations.

4. **Create an account for the role.**

   The background for this step and the procedures to create a role account are in Chapter 5<![%XRefs; [Chapter 5, “Using the User Manager to Configure User and Role Accounts]]>.”<![%EditorsComments; [roles: managing <$endrange>; roles: creating]]>

Aliasing `vi` to `adminvi`

When roles or other user accounts have the adminvi(1MTSOL) command assigned instead of the vi(1) command in one of their profiles, they get the following error in `dtterm(1)`, which is the role’s default terminal, if they accidentally type the shorter name, `vi`, instead of the longer name, `adminvi`.

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

The default profile(4) file for all roles in the `/etc/security/tsol/home/`*role_name* directories has the following function to alias `vi` to `adminvi`:

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

The alias does not work because `dtterm` does not read the `.profile` file unless the following entry is also made by the account in the `$HOME/.Xdefaults-`*hostname* file 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<![%XRefs; [Chapter 3, “Managing User Accounts]]>” under “Managing Startup Files in a Trusted Solaris System” on page 71<![%XRefs; [“Managing Startup Files in a Trusted Solaris System” on page 77]]>. For the procedure to make the supporting entry in `.Xdefaults-`*hostname*, see “To Force `dtterm` to Launch New Shells as Login Shells ” on page 90<![%XRefs; [“To Force dtterm to Launch New Shells as Login Shells” on page 101]]>.

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 execution 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 105<![%XRefs; [“To Assign the trusted_edit Editor to a Role” on page 119]]> for the procedure.

▼ To Assign the `trusted_edit` Editor to a Role

1. **If the `trusted_edit` command is not in one of the role’s profiles, do the following:**

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

      See “To Login and Assume an Administrative Role” on page 15<![%XRefs; [“To Login and Assume an Administrative Role” on page 15]]>, if needed.

   b. **Add the `trusted_edit` command to one of the account’s execution profiles.**

      See “To Customize an Administrative Role” on page 244<![%XRefs; [“To Customize an Administrative Role” on page 254]]>, if needed.

      i. **Add the `/usr/dt/bin/trusted_edit` script.**

ii. Give the script the proc_audit_tcb privilege.

Note - Either the security administrator or the role assigned the `trusted_edit` script can do the rest of the setup.

---

2. If desired, alias the `vi` command to `trusted_edit` by doing the following:

a. Search for the `vi` function in the `.profile` file in the `/etc/security/tsol/home/`*role_name* 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:

```
Dtterm*LoginShell: true
```

Note - Because `trusted_edit` launches an editing window, it cannot be used for command line editing, which may be the only option available in a remote login session. For this reason, do not assign `trusted_edit` as the only editor for a role, unless you know the role is not going to be doing remote editing on the command line.

CHAPTER 5

Using the User Manager to Configure User and Role Accounts

As discussed in Chapter 3 and in Chapter 4 the security administrator and system administrator roles share the responsibility for entering information into the various fields of the User Manager to create and maintain accounts for users and roles.

The following two main sections in this chapter detail the types of information needed when specifying accounts for users and roles and then shows the steps for using the User Manager to add or modify user and role accounts.

Note - The terms *User Manager*, *user name,* and *user ID* are retained for compatibility with earlier versions of the Solstice User Manager, even though the Trusted Solaris User Manager not only configures accounts for users but also for roles. The *user name* is actually an account name, which is a string of characters by which a user *or a role* is identified to the system, and the *user ID* is actually the account ID that can apply to either user or role accounts.

Understanding the Information Entered in the User Manager Dialog Boxes

Account information for users and roles is specified by the security administrator and system administrator roles in a series of dialog boxes launched from buttons on the User Manager. See Chapter 3" where the division of responsibilities for

configuring user accounts and the authorizations needed to access each button are described in more detail.

The buttons that launch the dialog boxes are shown in Figure 5–1.

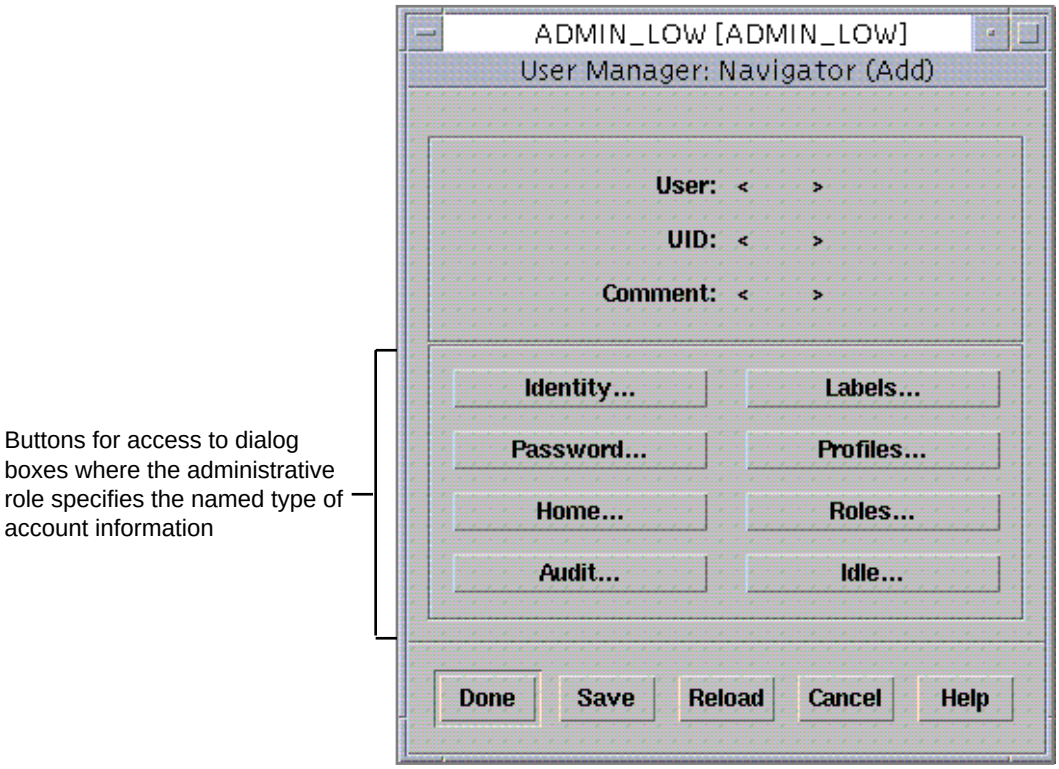


Figure 5–1 User Manager: Navigator

The sections listed here describe the type of information required on each of the dialog boxes and discuss choices and trade-offs among the various options you have when setting up an account.

- "Identity " on page 110
- "User Name, User ID, Group Name(s) and Group Id(s)" on page 110
- "Decisions to Make Before Starting" on page 111
- "Comment" on page 111
- "Decision to Make Before Starting" on page 111
- "Shell" on page 112
- "Using the Profile Shell To Enable Accounts" on page 112
- "Using the Profile Shell To Restrict Accounts" on page 112
- "Decision to Make Before Starting" on page 112

Identity

The system administrator enters the following information for an account in the identity dialog box:

- The name and number that identifies the user or role
- The user or role's group and (optional) supplementary groups
- A comment
- A default shell (command interpreter)
- Which type of account is being created, whether it is for a normal user, a non-administrative role, or an administrative role

User Name, User ID, Group Name(s) and Group Id(s)

Each account name needs to have a unique name (called the username) and a unique number (user ID or UID).

Caution - Make sure never to reuse user names or user IDs over the lifetime of the system. Ensuring that user names and UIDs are unique prevents possible confusion over which user did what or which user owns which files when archived files are restored or audit records are analyzed.

Whether the account is for a user or a role, the account's user name, user ID number, group name, and group ID number are used in these ways:

- They all are associated with each process that executes a command on behalf of the account.
- They all are associated with any files and directories created by the account.
- They all are used in DAC decisions to determine whether the user has access to files and directories created by other users based on standard UNIX file and directory permissions and access control lists (ACLs).
- For a normal user account, the user name is requested at login as part of the identification and authentication process.
- For a role account, the user name of the role account is requested when the user assumes a role through the trusted path.
- For a normal user account, the UID becomes the audit ID, which is stored in audit records that are generated by any auditable actions taken by that user. The audit ID stays the same throughout a login session, even if a user assumes a role and begins work under the role's UID. The audit ID allows administrators to trace audited actions back to individual users.

Decisions to Make Before Starting

♦ **Decide the account's user name based on your site's convention.**

- Account names cannot be longer than 8 characters.
- Account names must be unique in the network.
- Account names must not be reused during the life of the system.

♦ **Decide the account's user ID (UID) number, which is used along with the user name to identify the account on the system.**

- UIDs must be unique in the network.
- UIDs must not be reused during the life of the system.

♦ **Decide the primary group for the user and if the user's should belong to a secondary group or groups.**

You need to have the group names and GID numbers available to assign to the new account.

Where This Information is Entered Into the User Manager

See Step 6 on page 137.Step 6 on page 137, Step 6 on page 138, Step 6 on page 138, and Step 6 on page 138.

Comment

For user accounts, the Comment, which is entered in the GCOS field in the passwd(4) entry, contains the first and last name of the user, and perhaps the job title and work phone number. Some informal organizations allow employees to decide what goes into the comment, and sometime the entry is used for humorous purposes, such as this example for a user who writes manuals: "Roseanne Sullivan – Manual Laborer." The text you enter for the user in the Comment: field appears in the From: line when the user sends email, and is part of the information sent about the user if someone enters finger(1), among other uses:

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

For a role account, enter whatever comment is appropriate.

Decision to Make Before Starting

♦ **Provide a comment consistent with your site's usage.**

Where This Information is Entered Into the User Manager

See Step 6 on page 137.Step 6 on page 137.Step 6 on page 138.

Shell

The system administrator chooses a default shell for the user or role account, either Bourne, Korn, C, Profile or other. 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.

Each time any account executes a command in the profile shell, the shell consults the profile database to find whether the command is in any of the account's execution profiles and whether any privileges are specified to be inherited by that command. Any inheritable privileges specified in the database for the command and any authorizations in any of the account's execution profiles are made available for use while the command is executing.

Using the Profile Shell To Restrict Accounts

The profile shell can be used to *restrict* accounts in what they can do, because the profile shell, unlike other shells, may be used to limit the account to a specified set of commands.

Using the Profile Shell To Enable Accounts

Accounts can be *enabled* by the profile mechanism: Some accounts can be given authorizations and access to commands that may be run in the profile shell with inheritable privileges not be available to all users.

Note - The profile shell is *essential* for a user when any of that user's execution profiles specify a command that needs to run with inheritable privilege, because the profile shell is the only shell that allows a process executing a command to inherit privilege. No matter what default shell has been assigned to an account, every account can invoke the profile shell on the command line in another shell.

Decision to Make Before Starting

- **Identify the account's shell.**

Where This Information is Entered Into the User Manager

See Step 6 on page 137.Step 6 on page 137.Step 6 on page 138.

Account Type

The system administrator role identifies whether the account is being created for a normal user, a non-administrative role or an administrative role. Choose non-administrative role or administrative role only when you are configuring a new role account as described in Chapter 8.

The administrative role does its work in an administrative workspace and runs with the *Trusted Path Attribute*. Being able to work in a role workspace and to have its processes inherit the Trusted Path Attribute, also called the Trusted Path Process Flag, are the only characteristics that are different in an administrative role compared to a non-administrative role. The Trusted Path attribute allows administrative roles to do certain things, like run privilege debugging, that are not permitted to non-administrative roles.

A non-administrative role does its work in a regular user's workspace. Being able to configure non-administrative roles allows your site to assign roles to certain users. The roles restrict the users to a controlled sets of commands, actions, and attributes to perform limited tasks, without allowing them to do all the things allowed to administrative roles. If a non-administrative role is assigned to multiple users either at the same time or one after another, each time anyone assumes the role, that person works in the same environment, shares the same home directory, and access the same files. The operator role in the default system is a non-administrative role. See Chapter 4, for more about this topic.

Decision to Make Before Starting

- **Choose "Normal User," if you are setting up a regular user account.**

You may later specify in the Role dialog box whether this user may or may not be allowed to assume any of the *existing* roles.

- **Choose "Admin. Role" if you are setting up an account for a *new* administrative role.**

- **Choose "Non-Admin. Role," if you are setting up an account for a *new non*-administrative role.**

Where This Information is Entered Into the User Manager

See Step 6 on page 137.Step 6 on page 137.Step 6 on page 139.

Password

The security administrator does the following in the password dialog box:

- Creates a password for the account or selects among other options
- Specifies how long before the user *can* change the password
- Specifies how long before the user *must* change the password
- Specifies a length of time before the password expiration date to send a warning
- Chooses a password generation method to be presented to the user when the user changes the password through the Trusted Path
- Open the account, leave it closed, or specify it as “Always Open”
- Specifies whether NIS+ credentials are to be used

Background About Creating a Password or Selecting Other Password Options

In most cases, when setting up an account for a user or role, the security administrator creates a password for that user or role. The security administrator gives the initial password to the new user to be used at first login. It is up to the user or the site to decide how soon, if ever, the account is allowed or required to change the password for the account. The password must be eight characters in length. The account for the security administrator role and for users allowed to assume that role should be specified as Always Open. The password for a role account should not be subject to password aging constraints.

When a user is allowed to assume an administrative role, the security administrator role gives the initial password for the role to that user. The role password is never used to log in directly to a role, but it is used to authenticate the user who is assuming the role through the Assume Role option in the Trusted Path menu.

Both normal users and roles use their passwords when reauthenticating themselves if the screen automatically locks after the specified idle maximum time has elapsed. The security administrator may later create another new password for a user if the user has lost the password, or for other reasons. The security administrator role can change passwords for all other accounts but cannot change that of the security administrator role.

Note - Neither the passwd(1TSOL), nor the yppasswd(1) or nispasswd(1) commands are used in the Trusted Solaris environment, either by users or administrative roles. The security administrator can change an account’s password through the User Manager. For each account, all password changes must be done through the Change Password option on the Trusted Path Menu after the account logs in and is authenticated by the system. Normal users change passwords from the Trusted Path menu in a user workspace. Users in roles change the role passwords from the Trusted Path menu in a role workspace.

When either of the Type in or Choose from list are selected from the Password menu, the administrative role is immediately prompted to create a password for the user by the method selected.

Note - Only the "Type in" and "Choose from List" options on the Password menu are recommended because those are the only two options consistent with Trusted Solaris security policy for user identification, authentication, and accountability. Any of the other options should be used only in specialized circumstances where a knowledgeable security administrator ascertains that the option is both needed and allowable within the site's security policy— keeping in mind that using it makes the system both more vulnerable and harder to maintain.

TABLE 5–1 Password Creation Options, Descriptions and Recommendations

Option	Description
Account is locked[1]	When this option is selected, the account is locked with an invalid password and can be unlocked by assigning a new password. With this option, the account can own files but cannot log in until the account is unlocked.
No password – setuid only	This option sets up a specialized account that can own files but that can never be logged into, and it is never selected for user or role accounts. Accounts are set up with this option only for use by programs that use `setuid` to change their identity to the ID of the account so that they can access files owned by the account. Existing accounts of this type on the default system include `lp` and `uucp`. Being able to create no-password accounts helps reduce the risk of intruders deducing the passwords for these accounts for the purpose of breaking into the system.
Type in	Selecting this option brings up the Password Add dialog box for the administrative role to enter the account's password.
Choose from list	Selecting this option brings up the Password generation dialog box with a list of automatically generated passwords from which the administrative role chooses the account's password.

1. This option is not the same as the method used in Trusted Solaris 1.x for locking an account until the security attr >2.x, the "Closed" option in the Status menus used instead. See Table 5–3.

Decision to Make Before Starting

- **Decide whether to create a password for the account by typing it in or choosing from an automatically-generated list.**

Where This Information is Entered Into the User Manager

See Step 7 on page 140.Step 7 on page 140. Step 7 on page 142, Step 7 on page 142, and Step 7 on page 143.

Background on the Password Duration and Warning Fields

The password change options limit how long intruders could potentially access the system if they were able to guess or steal passwords. Establishing a minimum length of time to elapse before change stops users who have just been given new passwords from reverting immediately to their old passwords. The passwords for role accounts should not be subject to any password aging restraints.

Decisions to Make Before Starting

- **Decide how long before the user *may* change the password,**

- **Decide how long before the user *must* change the password, which is specified by either:**
  - Entering a number of days, weeks or months after “Max Change” or
  - Choosing an Expiration Date from the Days, Months, and Years menus.

- **Decide the number of days, weeks, or months or a date before the password is to expire for a warning to be sent to the user.**

Where This Information is Entered Into the User Manager

See Step 7 on page 140.Step 7 on page 143. Step 7 on page 143, Step 7 on page 144, Step 7 on page 144, Step 7 on page 144, and Step 7 on page 140.Step 7 on page 145.

Background About Selecting a Method for Password Generation

The method of password generation selected in the Change by menu determines whether the manual or automatic password generator dialog box is presented whenever the account changes the password through the Trusted Path.

The rules shown in Table 5–2 are enforced by the software when an account chooses the Change Password option from the trusted path menu.

TABLE 5–2 Password Rules for Manually Created Passwords

Rules
The password must be eight characters in length.
The password must contain at least two alphabetic characters.
The password must contain at least one numeric or special character.
The password must differ from the user's login name and any reverse or circular shift of that login name. (For this comparison, upper case letters and lower case letters are considered to be equal.)
A new password must have at least three characters different from the old. (For this comparison, upper case letters and lower case letters are considered to be equal.)

Note - The rules for passwords created using the User Manager, and the passwords created by the password generation software are slightly different from those shown in Table 5–2. For example, the automatically-generated passwords do not have numbers. Although generated passwords and the User Manager-created passwords do not fully conform to the rules for manually-generated passwords, these passwords are also accepted by the system. The security administrator is strongly advised to create passwords for user accounts that conform to the requirements shown in Table 5–2. See also "Security Requirements" on page 36.

Decision to Make Before Starting

Decide whether the account can pick its own password or if it must choose one from an automatically-generated list.

Where This Information is Entered Into the User Manager

See Step 7 on page 140.Step 7 on page 146.

Background on the Account Status Menu Options

The Status menu options are new to the Trusted Solaris version of the User Manager. See Table 5–3 for descriptions and recommendations for each of the status options.

TABLE 5–3 Status Menu Options, Descriptions, and Recommendations

Option	Description and Recommendation
Closed	In effect until the account has been completely specified. Unlike account locking, which is specified in the Password menu near the top of the Password dialog box (see Table 5–1). this option does not affect the account's password. After three (3) failed login attempts, the status of the account is automatically changed to closed. [The word "locked" is entered into the "lock" field in the user's entry in the tsoluser(4TSOL) password file.] The security administrator role then must use the User Manager to reset this value to Open.
Open	Use after the account has been completely specified or, using discretion, reset to this value after the account has been locked. The security administrator role chooses this option after setting up the security-relevant characteristics of the user account., or if the account has been locked after 3 failed login attempts, resets this value after verifying that the system has not been compromised. This value is not accepted by the User Manager until the account is fully specified.
Always Open	Use only when your site's security permit an account to stay open even if the limit on failed logins is reached. This option may be used at site's where the need for protection against the denial of service (which would occur if a malicious user attempted multiple logins with the attention of closing accounts) is greater than the need to limit attempts to penetrate the system by repeated attempts to guess passwords. It is recommended that at least one user who is able to assume the security administrator role be given the Always Open status.

Note - By default, an account will be closed after three failed login attempts on a local host. (If needed, see "Changing the Maximum Number of Bad Password Entries" on page 41.)The security administrator can set up trusted accounts to be exempt from this restriction by choosing the Always Open option for the account.

Decision to Make Before Starting

- **Decide the account status.**

Where This Information is Entered Into the User Manager

See Step 7 on page 140.Step 7 on page 147.

Background About Checking NIS+ Credential Table Setup

Clicking the toggle button next to the Cred. Table Setup field, adds the NIS+ principal's public and private keys to the `cred` table, which establishes the account as a NIS+ principal and adds the accounts password to the NIS+ databases. (See

Chapter 5, “Administering NIS+ Credentials,” of the *NIS+ and FNS Administration Guide - Solaris 2.5.)*

Decision to Make Before Starting

- **Decide whether to check the Cred. Table Setup box for NIS+ credential setup.**

Where This Information is Entered Into the User Manager

See Step 7 on page 140.Step 7 on page 147.

Home

The system administrator sets the following in the Home dialog box:

- Whether or not the home directory is automatically created
- The home directory’s pathname
- The name of the server for the home directory
- The skeleton path for the shell initialization files
- The desired home directory DAC permissions
- The name of the account’s mail server

Note - Because you are prompted to specify a server for the account’s home directory, the server for the account’s home directory must be configured before you create the account.

Why Say Yes to Automatic Creation of Home Directories?

If the system administrator does not tell the system to automatically create the home directory, then the system administrator must set up the home directory manually as an MLD before the user can log in. Because the system can automatically copy start-up files to the initial SLD within the multilevel home directory the first time an account logs in, it is advisable to have the system do the set up.

Skeleton Path Considerations

When you are prompted to specify a skeleton path for shell startup files, if you specify `/etc/skel` as the skeleton path location without making any changes, the supplied `local.cshrc`, `local.login` and `local.profile` start-up files for all the shells are automatically copied into the account’s home directory. The account is

responsible for renaming the appropriate copied file and removing the files that do not apply to her or his shell. For example, if the user's login shell is the C shell, the user would remove the `local.profile` and would rename and make any desired modifications to the `local.cshrc` file and `local.login` files.

Note - Without administrative intervention, neither the `.login` or `.profile` are looked at when the window system comes up unless the account's default shell is pfsh(1MTSOL). Also, files in a skeleton directory are copied only into the first home directory SLD created at the account's initial sensitivity label, and the administrator or user must ensure that startup files are copied to all subsequent SLDs created at other sensitivity labels for the account. The needed background and procedures related to startup files are described in "Managing Startup Files in a Trusted Solaris System" on page 71 in Chapter 3."

Controlling the Use of Shell Initialization Files

When a user's login shell is specified as either Bourne, Korn or C shell, keep in mind that, by default, shell initialization files are not looked at during login because a variable in a `*.dtprofile` file is commented out. Other comments in the `*.dtprofile` files encourage the administrator or individual users to modify the initialization files before removing the comment if they wish the appropriate initialization file for a shell to be sourced, and to make sure that the initialization file does not do anything that requires a terminal emulator or user interaction while the window system is coming up.

A personal $HOME`/.dtprofile` file could be modified in such a way that it would allow the user to launch commands while the window system is coming up and before the user's profiles are in effect. To support the special purposes of the profile shell, pfsh(1MTSOL), it is important that users not be allowed to specify any commands that run before the user's shell is in effect. So *when a user's login shell is specified as the profile shell*, the `.profile` file *is* looked at during login, but the profile-shell-user's $HOME`/.dtprofile` file is never consulted.

For more details on these topics, see "To Make `.login` or `.profile` Looked at During Login" on page 89 and see "Managing Startup Files in a Trusted Solaris System" on page 71 in Chapter 3."

Decisions to Make Before Starting

- **Decide the account's home directory server.**

- **Decide whether to allow the automatic creation of a multilevel home directory (MLD) for the account.**

- **Decide what to put in `/etc/skel`, 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 91 in Chapter 3.**

- **If a Trusted Solaris administrative role has created a shell-specific subdirectory in `/etc/skel` for each of the shells, prepare to enter the correct `skel`*X* subdirectory name into the Skeleton path field in the User Manager, based on the user's default shell.**

- **Decide the default permission bits for files the user creates (for setting the `umask`).**

- **Decide the mail server for the account.**

Where This Information is Entered Into the User Manager

See Step 8 on page 149.Step 8 on page 149, Step 8 on page 150, Step 8 on page 150, Step 8 on page 150, Step 8 on page 150, Step 8 on page 150, and Step 8 on page 151.

Labels

The system administrator specifies the following in the Labels dialog box:

- The clearance
- The minimum SL
- Whether the account is allowed to view administrative labels or should be shown alternate labels within the user accreditation range
- Whether the user can view SLs
- Whether the user can view ILs

Background on the Clearance and Minimum Label

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

The account's minimum sensitivity label defines the lowest sensitivity label at which the account can work. Administrative role accounts have a minimum sensitivity label of ADMIN_LOW. The minimum sensitivity label is the sensitivity label of the first workspace that comes up for the account after the first login.

During the first session or during any subsequent session, an account can change the sensitivity label of any workspace and can specify an alternate workspace so that any workspace at any sensitivity label within the user's personal label range may be first workspace that comes up during subsequent sessions.

Note - Both the account's clearance and minimum sensitivity label must be dominated by the highest sensitivity label defined in the user accreditation range and must dominate the minimum clearance defined in the user accreditation range in the label_encodings(4TSOL) file.

Decisions to Make Before Starting

- **Decide a clearance for the account that dominates the minimum clearance set in the** `label_encodings` **file.**

- **Decide the account's minimum sensitivity label, which is the minimum sensitivity label at which the account is permitted to work.**

Where This Information is Entered Into the User Manager

See Step 9 on page 152, Step 9 on page 152., Step 9 on page 152.Step 2 on page 155.

Background on Displaying Labels

The system administrator specifies how labels are displayed for the account in the lower part of the Labels dialog box. In the following discussion, the term CMW label is used to refer to the IL and the SL when they are shown together in the standard format, with the long name of the information label and any words and the short name of the sensitivity label and of any words.

```
INFORMATION LABEL [SL]
```

Label View

The label view allows the security administrator to determine whether the names for administrative labels are ever displayed for the account, because at some sites the names of administrative labels are considered to be classified information. ADMIN_HIGH and ADMIN_LOW are the default names for the administrative labels, but because the security administrator role is able to specify alternate names for administrative labels in the label_encodings(4TSOL) file, your site could have other names assigned. The administrative labels appear in sensitivity labels, information labels, and clearances.

Note - See *Trusted Solaris Label Administration* for how the security administrator role specified alternative names.

A system-wide default label view is set in the `label_encodings` file. During installation, the default setting of INTERNAL is either accepted or changed by the site's security administrator. When each user or role's account is being set up, the security administrator chooses one of the following for the account:

- Internal
- External
- Sys Default

Internal View

The INTERNAL view allows the account to see the *names* of the administrative labels, which are either the strings "ADMIN_HIGH" and "ADMIN_LOW" or their administratively set names.

External View

Accounts with the External option are not exposed to the names of the administrative labels. If the label view for an account is set to EXTERNAL, the *minimum* valid label of the same type in the User Accreditation Range is shown instead of the ADMIN_LOW label or its site-specified equivalent. Also, when the account's label view is EXTERNAL, the *maximum* valid label of the same type in the User Accreditation Range is shown instead of the ADMIN_HIGH label or its site-specific equivalent.

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 it is displayed to hide its real name.

Sys Default

If the Sys Default option is selected for an account, whatever value is specified in the label_encodings(4TSOL) file for the "DEFAULT LABEL VIEW" key word (EXTERNAL or INTERNAL) in the file applies to the account.

Example of the Effects of the Label View

Here is an example of how the default label view affects what the user sees. Remember that the IL of a newly opened file is always ADMIN_LOW because it is

empty, while its SL is the label of the process that created it. So, if a user begins to edit a file in a Text Editor at TOP SECRET at a site where the default administrative label names are being used, the CMW label is:

```
ADMIN_LOW [TS]
```

In the example, the lowest valid IL in the user accreditation range is UNCLASSIFIED. If the account has the INTERNAL label view set, the CMW label displays with the administrative label ADMIN_LOW in the trusted frame of the Text Editor as: ADMIN_LOW [TS].

If the account has the EXTERNAL label view set, the CMW label displays with the label UNCLASSIFIED replacing the administrative label, as: `UNCLASSIFIED[TS]`.

Note - Ultimately, whether the IL portion and the SL portion of administrative labels or their substitutes display, or whether the administrative labels display at all, is determined by the values set in the SL and IL display menus below the label view menu.

Decision to Make Before Starting

- **Decide whether the user is allowed to see the names of administrative labels or if 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.**

Where This Information is Entered Into the User Manager

See Step 9 on page 152.Step 3 on page 156.

Background on Showing or Hiding SLs and ILs

When the complete CMW label, the SL, or the IL are displayed, they appear in the following locations, among others:

- The trusted path indicator at the bottom of the screen,
- At the top of window frames, and
- In the title bar on window icons.

When the IL and SL are displayed together in the form of a CMW label, SLs display in short form inside square brackets ([]). When the SL displays alone, it displays in long form within the square brackets.

Decision to Make Before Starting

- **Decide if the account can view the complete CMW label, the Sensitivity Label alone, or the Information Label alone.**

Where This Information is Entered Into the User Manager

Step 9 on page 152.Step 4 on page 157, and Step 5 on page 157.

Profiles

The security administrator role assigns execution profiles to accounts. The available execution profiles are displayed in a scrolling list in the Profiles dialog box, along with brief descriptions. See Appendix A" for tables that list the actions, authorization, and commands defined for each profile.

Note - The default administrative roles provided with Trusted Solaris have the needed execution profiles already assigned. If you create a new role, you may create a new profile for that role; if so, create the new profile before you configure the role account, so that you can assign the profile to the role's account through the User Manager.

If you are configuring a user account and plan to assign a role to that user, *you do not need to and should not assign a corresponding role profile to the user's account*. It's important to realize that the role accounts have their own execution profiles, which come into effect only when the user assumes the role and in effect, changes his or her identity to that of the role.

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.

The *order of profiles* is also 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. This is similar to how the system uses the first instance of a command that it finds in any directory in the user's PATH variable: for example, if you invoke the ps(1) command by entering `ps` on the command line, either `/usr/bin/ps` or `/usr/ucb/ps` will run depending on how your PATH is set up, and each version has its own set of options.

Similarly, if the user executes a command or action that is defined with differing attributes in more than one execution profile (for example, with differing privileges or with `setuid` turned on in one but not the other), the *command or action runs as it is defined in the first profile in which it is found in the profile list*.

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

Also, it is possible to take advantage of the order of profiles to allow certain accounts to run certain commands and actions with privileges or with an effective UID or GID or at a certain sensitivity label and run all other commands and actions with none of these special security attributes. Do this by assigning a set of profiles that explicitly names any commands or actions that the account needs to run with the specific security attributes and then, at the end of the list of profiles, give that account the All profile that permits the account to run any other command or action in the system without privilege. In this way, the account can use the explicitly named and configured commands and actions with the specified attributes and use all other commands without any special attributes.

For more about execution profiles see “Review of Terms” on page 198 in Chapter 8.”

Decisions to Make Before Starting

- **Decide on at least one execution profile to assign to the account.**

- **Decide the order in which execution profiles should be listed**

Where This Information is Entered Into the User Manager

Step 8 on page 158.Step 8 on page 158 and Step 8 on page 159, and Step 8 on page 158.Step 8 on page 159 and Step 8 on page 159.

Roles

The security administrator role assigns roles to normal user accounts. The available roles are displayed in a scrolling list in the Roles dialog box, along with brief descriptions.

Role accounts cannot assume other roles, and the User Manager does not allow access to the Role dialog box when a role account is being configured. However, you may decide to assign multiple roles to a single user if it is consistent with your site’s security policy.

Decisions to Make Before Starting

- **Decide which roles, if any, the account is allowed to assume.**

Where This Information is Entered Into the User Manager

Step 9 on page 160.

Idle

The security administrator role specifies whether an action is taken if the workstation remains idle for a specified amount of time. The two action choices (log out the user or lock the screen) are described in the following table.

TABLE 5–4 Idle Action Choices

Log Screen	Locks the screen after the specified period of idleness has passed. The account must then supply a password to regain access to the session. Moving the mouse or pressing a key causes the dialog box shown in Figure 5–2 to display so that the user can enter the password.
Log out	Logs the user out of the system entirely when the specified period of idleness has passed. The user must log in again to regain access.

Figure 5–2 Lockscreen Password Dialog Box

The security administrator role also chooses from the Idle Time menu an amount of time before either action, either 1, 2, 3, 4, 5, 10, 15, 30, 60, or 120 minutes or Forever.

The Forever menu option essentially prevents any action from being taken when the workstation is idle.

Decisions to Make Before Starting

- Decide on the consequences, if any, of a workstation sitting idle when a user is logged on but not active, if the screen should be locked, or the user's session should be terminated and the user logged out.
- Decide what length of time must pass without activity on the workstation before the specified action will be taken.

Where This Information is Entered Into the User Manager

Step 10 on page 163.

Setting Up or Modifying a User or Role Account

The following procedures show how to use the various options on the User Manager to specify account information.

▼ To Launch the User Manager

Note - The admin role adds a new account and specifies Identity and Home directory information. The secadmin opens the account and specifies security-related information.

---

1. **Log in as a user who is able to assume the appropriate role and assume the role.**

   See "To Login and Assume an Administrative Role" on page 15 if needed.

2. **Access the Solstice_Apps folder in the Application Manager, and click the User Manager icon.**

   See Figure 5–3. The User Manager: Load dialog box displays as shown in Figure 5–4 and the main User Manager window displays with no users.

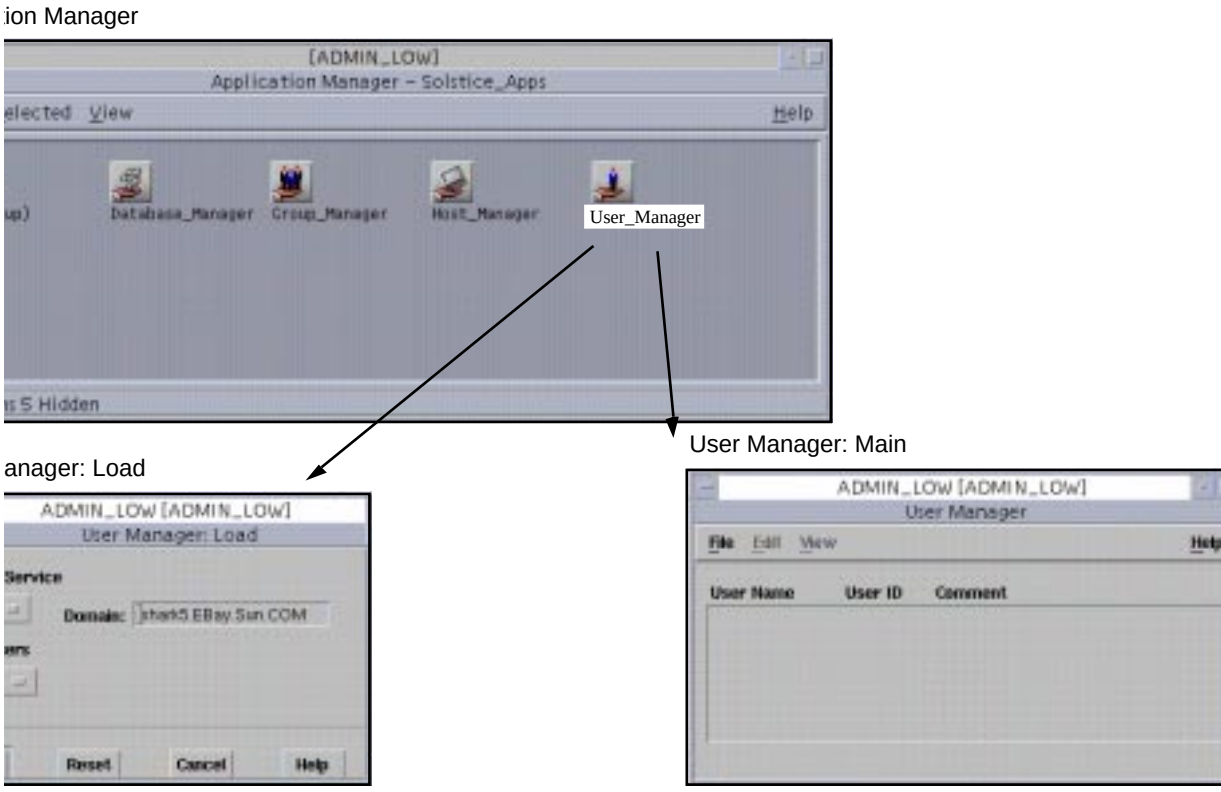


Figure 5–3 Launching the User Manager

▼ To Load a List of User and Role Accounts Using the Load Dialog Box

1. **From the Naming Service menu, choose the NIS+ option.**

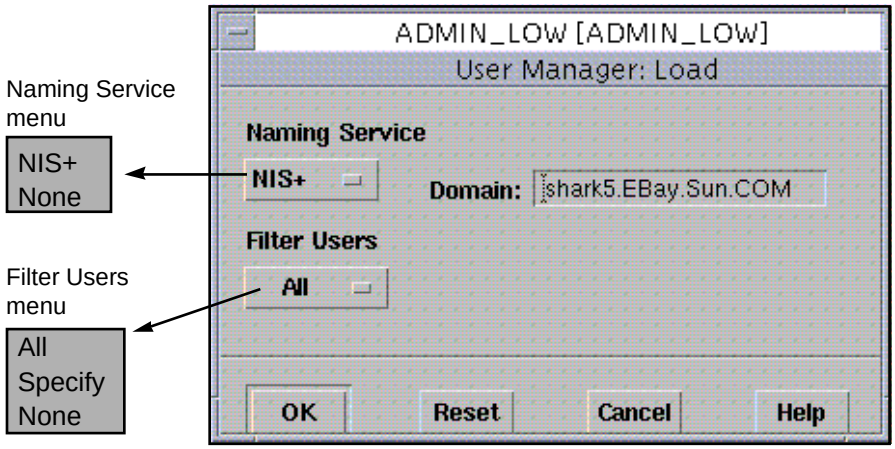


Figure 5–4 User Manager: Load Dialog Box with Filter Users Menu

---

Note - NIS+ is the recommended option in the Trusted Solaris system, because it supports the centralized administration of all the workstations and servers together in one a single distributed system, which is important for both user accountability and trusted administration. The None option should only be used in specialized circumstances where a knowledgeable security administrator decides that local accounts are both needed and allowable within the site's security policy— even though they can make the system both harder to protect and harder to maintain.

---

2. **Accept the domain name shown or enter another domain name in the text field next to Domain.**

3. **Choose an option from the Filter Users menu.**
   a. **Choose All to display a list of names of already-configured accounts along with their associated user IDs and comments.**

   b. **Choose Specify to specify a user or set of users.**
      Next to the Specify option, do either Step 3 on page 130 or Step 3 on page 130.

      i. **Enter a specific username.**

      ii. **Enter a regular expression to filter the usernames.**

a. **Choose None to cause the main User Manager: Load dialog box to be displayed with no account names.**

4. **Click OK.**

▼ To Load Users or Exit (optional)

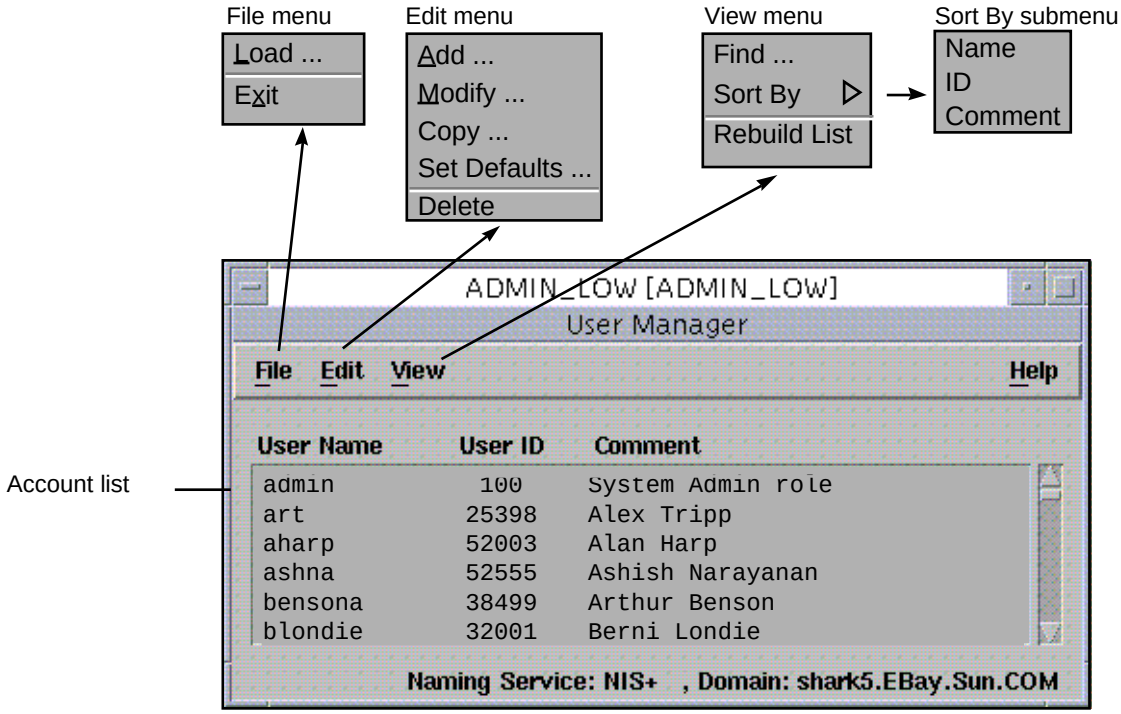


Figure 5–5 User Manager: Main Window and Menus

1. **Choose the Load option from the File menu to bring up the Load dialog box (see Figure 5–5).**

   See "To Load a List of User and Role Accounts Using the Load Dialog Box" on page 129 and Figure 5–4 for how to use the Load dialog box.

2. **Choose the Exit option to exit from all User Manager windows.**

▼ To Find or Sort Accounts

1. **Choose an option from the View menu.**

   The View menu options are shown in Figure 5–6.

   a. **Choose Find to enter a target string (containing zero or more wildcards) for locating a user in the list.**

      i. **Enter either a username, user ID, or comment.**

         See Figure 5–7. You can only specify one type of target at a time.

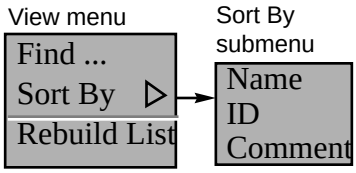


Figure 5–6 View Menu with Sort By Submenu

      ii. **Click Apply or OK.**

         OK highlights the next matching user on the User Manager list, if any, and then the User Manager: Find dialog box disappears. Apply highlights the next match, if any, and the Find dialog box continues to display. If there are multiple matches, click Apply again on the User Manager: Find dialog box to go to the next match.

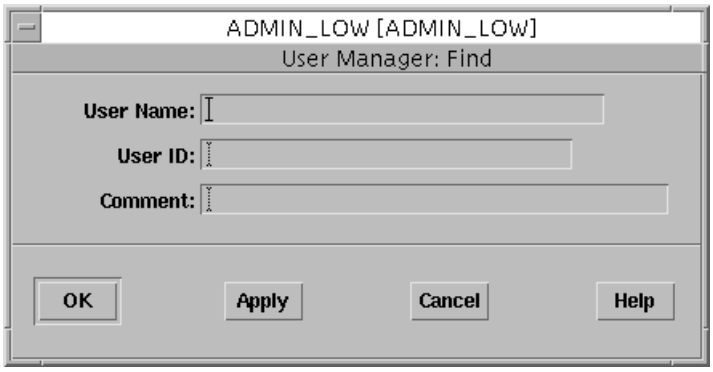


Figure 5–7 User Manager: Find Dialog Box

a. Choose Sort By to display a submenu and select one of the three fields to sort by: Name, ID, or Comment (see Figure 5–6).

b. Choose Rebuild List to display the current list of accounts with any additions, deletions or modifications made since the list was last displayed.

▼ To Add, Modify, or Delete Accounts

As stated in Chapter 3," the responsibilities for setting up accounts are divided between the security administrator role, which handles the security aspects of the user's record, and the system administrator role, which handles the general aspects. A role cannot modify its own account. The system administrator role first adds a new account, and then specifies the information for which the role is authorized, and the security administrator role then specifies the rest of the information and sets the account status to Open

1. Choose Add, Modify, Copy, Set Defaults, or Delete from the User Manager Edit menu shown in Figure 5–8.

Figure 5–8 User Manager Edit Menu

The admin role can use any of the options on the Edit menu. After an account is created by the admin, the secadmin role can use Modify; secadmin can also use Set Defaults.

Selecting any of the options from the Edit menu displays one of the Navigator dialog boxes. The Navigator (Modify) dialog box is shown in Figure 5–9. The buttons in the Navigator give access to additional dialog boxes for entering the named types of information.


[ADMIN_LOW]
User Manager: Navigator (Add)
User: < >
UID: < >
Comment: < >
Identity...
Labels...
Password...
Profiles...
Home...
Roles...
Audit...
Idle...
Done
Save
Reload
Cancel
Help

Figure 5–9 User Manager Options for the Security and System Administrators

---

Note - When buttons are grayed out when an administrative role brings up the User Manager, the authorization needed for that button is not in any of the role's execution profiles.

Note - On all the user manager dialog boxes that are invoked from the Navigator, clicking the Apply button saves the data and leaves the dialog box displayed, and clicking the OK button registers the data, and closes the dialog box that overlays the Navigator.

Note - The Audit button is not enabled in the 2.5.1 release of the Trusted Solaris operating environment. See the *Trusted Solaris Audit Administration* manual for how to set up auditing for individual users.

2. **To set defaults to apply to all new accounts, choose Set Defaults, and then click the button that allows you to update the desired type of information (optional).**

   The defaults you can set on each of the dialog boxes are shown in Table 5–5. The defaults you cannot set are grayed out.

TABLE 5–5 Defaults You Can Set on the User Manager Dialog Boxes

Dialog Box	Settable Defaults	For Instructions, Go to:
Identity	Primary Group	Step 6 on page 137
	Secondary Group	
	Log in Shell	
	User Type	
Password	All	Step 7 on page 140
Home	All except Path	Step 8 on page 149
Labels	All	Step 9 on page 152
Profiles	All	Step 8 on page 158

TABLE 5–5 Defaults You Can Set on the User Manager Dialog Boxes *(continued)*

Dialog Box	Settable Defaults	For Instructions, Go to:
Roles	All	Step 9 on page 160
Idle	All	Step 10 on page 163

3. **When adding a new account, choose Add or Copy, and then go to Step 6 on page 137.**

   Copy lets you create a new account by editing the fields already specified for an existing account. This option allows you to keep the entries you want from another user account while changing only the information that is different for the new account.

4. **When modifying an existing account, choose Modify, and then go to Step 6 on page 137.**

5. **When removing an account, choose Delete, and then go to Step 11 on page 164.**

6. **Enter a new account's identity Information, or change it for an existing account if desired.**

   See "Identity " on page 110, if necessary, to review the guidelines on what to specify in the Identity dialog box.

   a. **Click on the Identity button in the User Manager Navigator.**

      The Identity dialog box displays (see Figure 5–10).

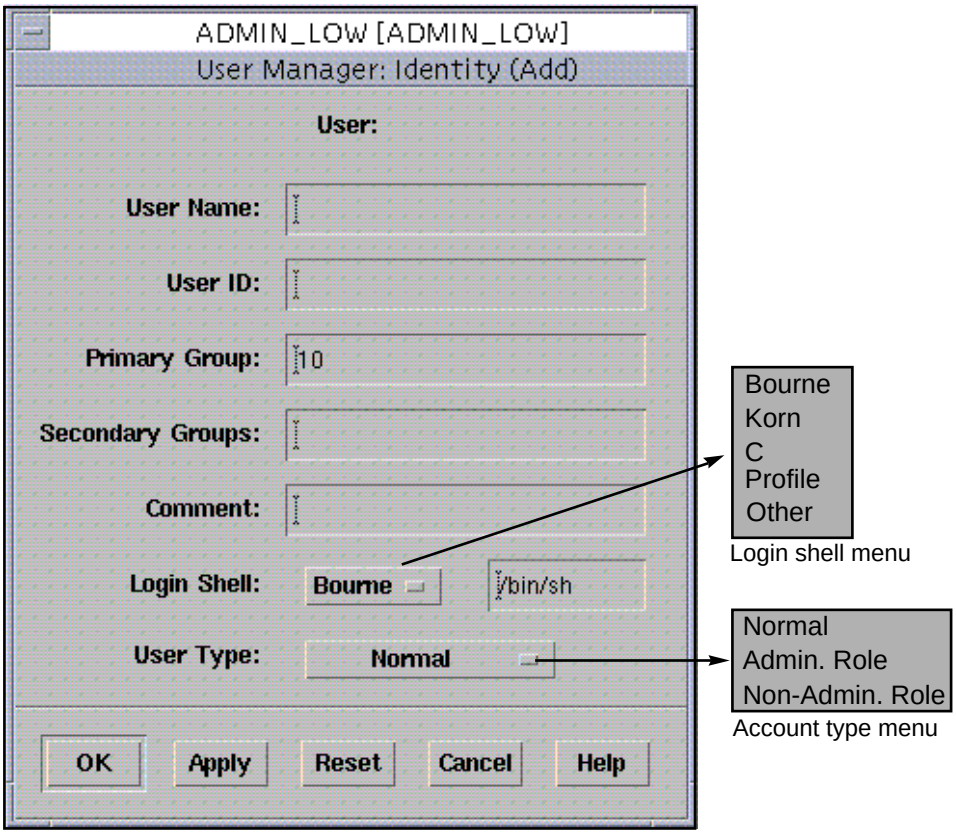


Figure 5–10 User Manager: Identity Add Dialog Box

b. Enter the name that identifies the user or role.

The name cannot be longer than eight characters. The name must be unique in the network and must never have been used before.

c. Enter the numeric ID for the user or role.

The ID must be unique in the network and must never have been used before.

d. Enter the user or role's group and (optional) supplementary groups.

e. Enter a comment.

f. Choose a default shell from the Login Shell menu: Profile, Bourne, Korn, C, or another type that you specify.

g. If you select other, enter the pathname of the shell.

Note - The profile shell should *always* be selected as the login shell for role accounts. Normal users may be configured to have the profile shell if you want use it to take advantage of its enabling or restricting capabilities.

h. Choose the type of account from the User Type menu.

Choose either normal user, administrative role, or non-administrative role.

i. Store your changes by clicking Apply or OK at the bottom of the Identity dialog box. (See Figure 5–11).

Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

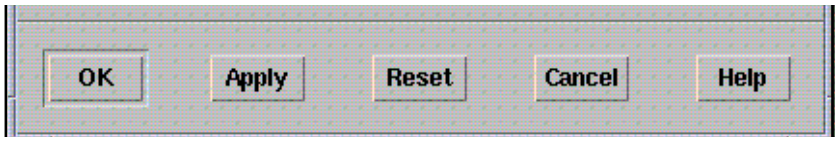


Figure 5–11 Controls on the User Manager: Identity Dialog Box

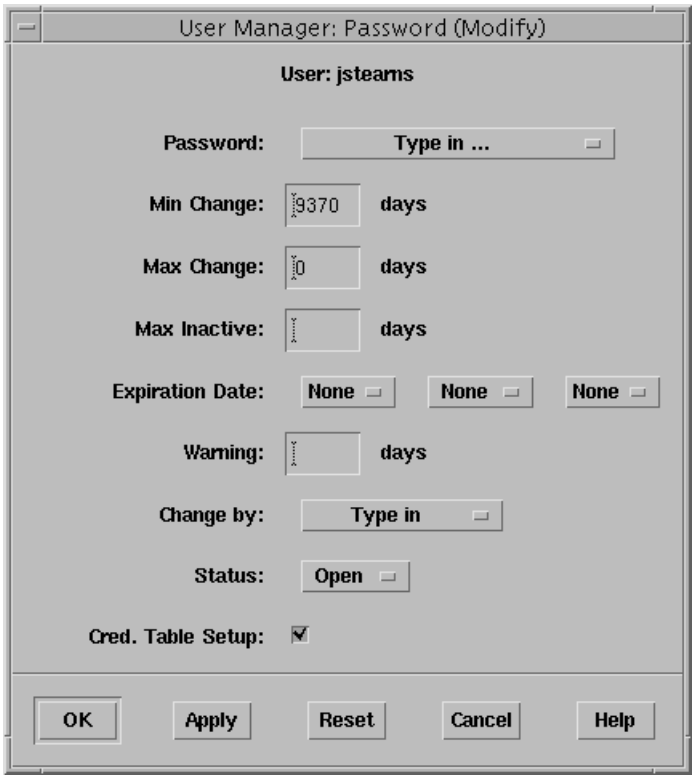
j. If you are done entering information for a new account or to update an existing account, go to Step 11 on page 164.

7. Enter a new account's password information, or change it for an existing account, if desired.

See "Password " on page 114, if necessary, to review the guidelines on what to specify in the Password dialog box.

a. Click the Password button in the User Manager Navigator.

The Password dialog box displays (see Figure 5–12).



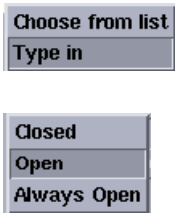


Figure 5–12 User Manager: Password Dialog Box

b. Choose either `Type in` **or** `Choose from list` **from the Password menu (see Figure 5–13).**

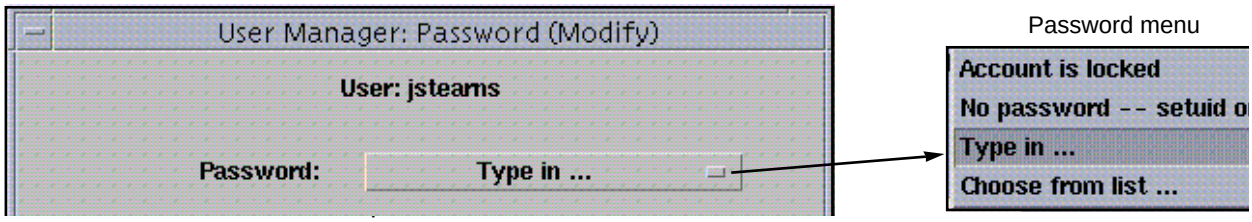


Figure 5–13 Password Dialog Box: Password Menu

Caution - The security administrator is responsible for protecting the password and ensuring that only the password's owner ever gets to see it.

If you choose Type in, the dialog box shown in Figure 5–14 displays. If you choose the Choose from list option, the dialog box in Figure 5–16 displays.

Figure 5–14 User Manager: Set Password

i. For a typed in password, enter the password in the Enter Password text field, press the Tab key to go to the Verify Password text field, and enter the password again (see Figure 5–14).

ii. To choose a password from a generated list in the Password Generator dialog box, click the circle to the left of the desired password, and then enter the password you selected into the text entry field below to confirm (see Figure 5–15).

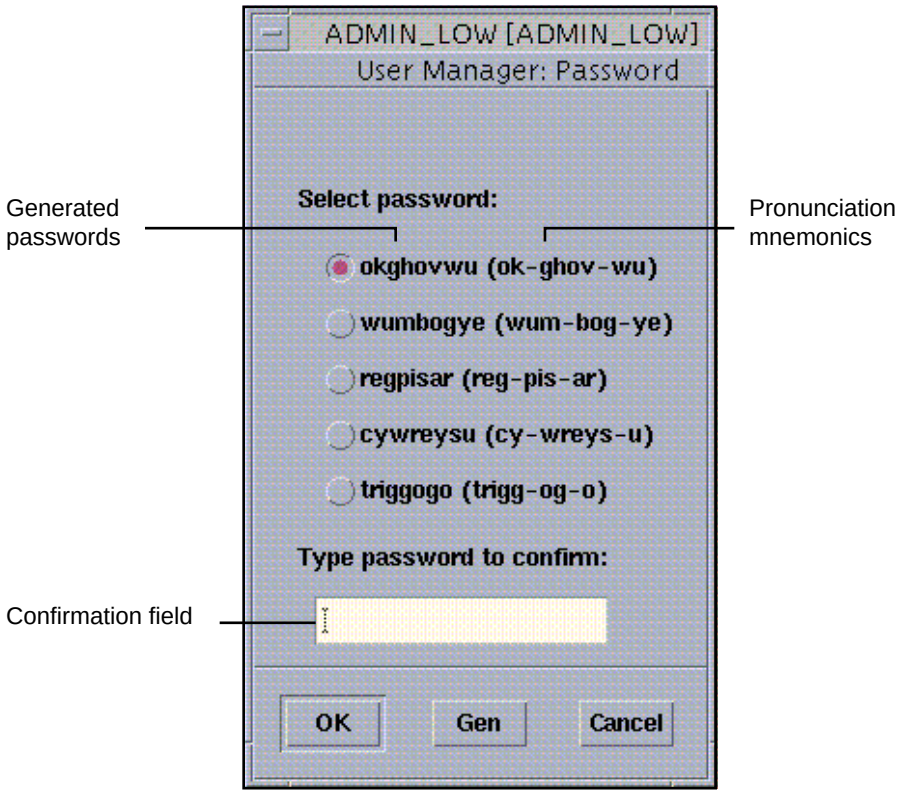


Figure 5–15 Password Generator Dialog Box

The Password Generator dialog box gives a choice of five system-generated passwords. The pronunciation mnemonic shown in parentheses to the right of each password divides the password into syllables to make it easier to remember.

iii. If you do not find a generated password that you like, click Gen to generate five new passwords, and select one, as described in Step 7 on page 142.

iv. Either click OK or press Enter to store the password.

a. Specify the duration of password validity (see Figure 5–16) (optional).

i. In the Min Change field, enter a minimum number of days that must elapse after a password change before the user may change the password on the account again.

ii. To have the password expire by a certain number of days after its last change, put that number of days in the Max Change text entry field.

iii. In the Max Inactive field, enter a maximum number of days that an account can be inactive before the user account is closed.

iv. To have the password expire on a certain date, enter the date in the Expiration Date field.

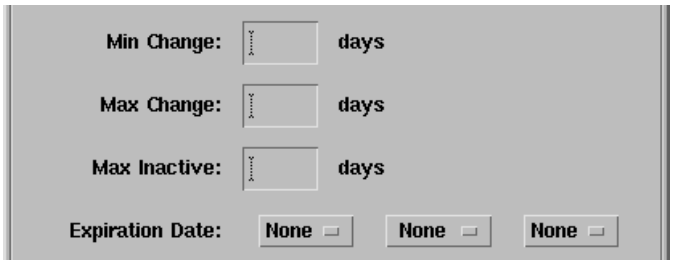


Figure 5–16 Password Dialog Box: Password Duration and Expiration Date Fields

a. If a password validity duration or expiration date is set in Step 7 on page 143, type into the Warning text entry field a number of days before the password expires for a warning to be sent (see Figure 5–17.)

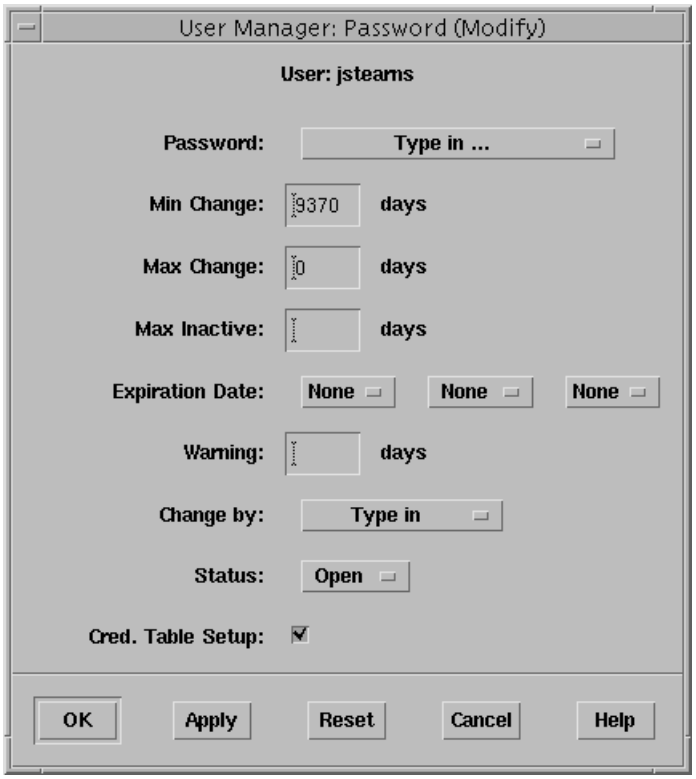


Figure 5–17 Password Dialog Box: Warning Field

Note - Entering meaningless combinations such as 1-none-2000 generates errors.

---

b. Set the password generation type for the user

The Generation menu in the Password dialog box lets you specify whether the user or the system is to supply the password (see Figure 5–18).

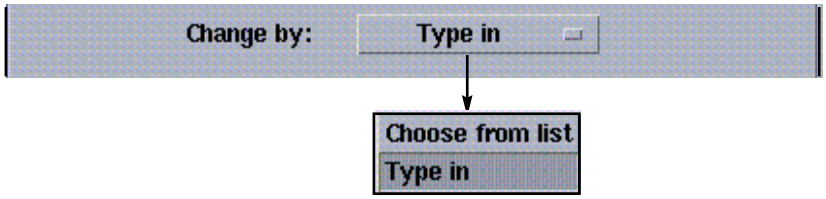


Figure 5–18 Password Dialog Box: Generation Field and Menu

If you choose the Choose from list option, the Password Generator dialog box displays whenever the user chooses the Change password option from the Trusted Path menu. (To see the Password Generator dialog box, see Figure 5–15).

c. Choose from the Status Menu to Open the account, leave it Closed, or set it Always Open. (See Figure 5–19).

A new account is closed until the security administrator opens it. The status of an open account automatically changes to closed after three failed login attempts under the account's user name. An always-open account is always accessible; it is never closed as a result of multiple failed logins.

Note - When an existing account is closed after three failed login attempts, the security administrator should investigate the situation before reopening the account.

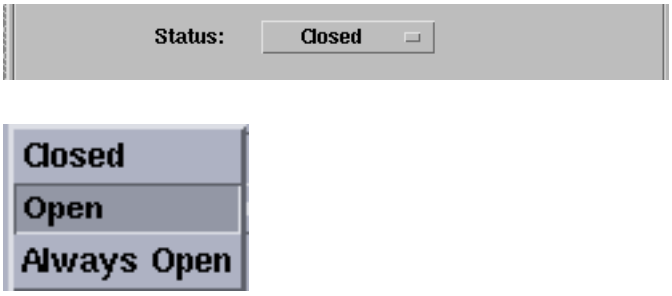


Figure 5–19 Password Dialog Box: Status Field and Menu

d. Click the toggle box next to Cred. Table Setup to add the NIS+ principal's public and private keys to the cred table. (See Figure 5–20.)

Figure 5–20 Credential Table Setup Check Box

Note - The recommended option is to check the Cred. Table Setup box when creating a new account so that the account is added to the NIS+ cred table and the account's password is added to the NIS+ databases, and to leave the toggle as already specified when modifying an account, unless you change the password.

---

e. Store your changes by clicking Apply or OK at the bottom of the Password dialog box (see Figure 5–21).

Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

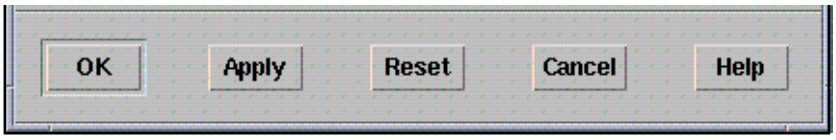


Figure 5–21 Controls on the User Manager: Password Dialog Box

f. If you are done entering information for a new account or to update an existing account, go to Step 11 on page 164.

8. Enter a new account's home directory information, or change it for an existing account, if desired.

See "Home " on page 119, if necessary, to review the guidelines on what to specify in the Home dialog box.

a. Click Home on the User Manager Navigator.

The Home dialog box displays (see Figure 5–22).

b. Check the Create Home Dir check box to have the account's home directory created as an MLD (optional).

c. Type in the pathname for the home directory next to Path.

d. Type in the name of the home directory's server next to Server.
If the home directory is mounted on a local partition on the account's primary workstation, enter the name of the primary workstation. Otherwise, enter the name of the NFS server for the account's home directory,

---

Note - The server must be configured prior to creation of the user account.

---

e. Check the appropriate permissions boxes under Permissions to specify the read, write, and execute permissions for the home directory by owner, group, and other.

f. Specify the mail server (optional).

g. Check the AutoHome Setup box to have the home directory be automounted automatically.

Figure 5–22 User Manager: Home Directory Dialog Box

h. Store your changes by clicking Apply or OK at the bottom of the Home dialog box (see Figure 5–23).

Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

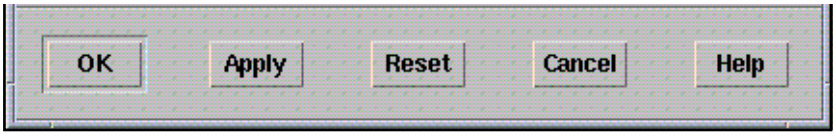


Figure 5–23 Controls on the User Manager: Home Dialog Box

i. If you are done entering information for a new account or done updating an existing account, go to Step 11 on page 164.

9. Enter a new account's label information, or change it for an existing account, if desired.

See "Labels" on page 121, if necessary, to review the guidelines on what to specify in the Labels dialog box.

a. Click Labels on the User Manager Navigator.

The Labels dialog box displays (see Figure 5–24).

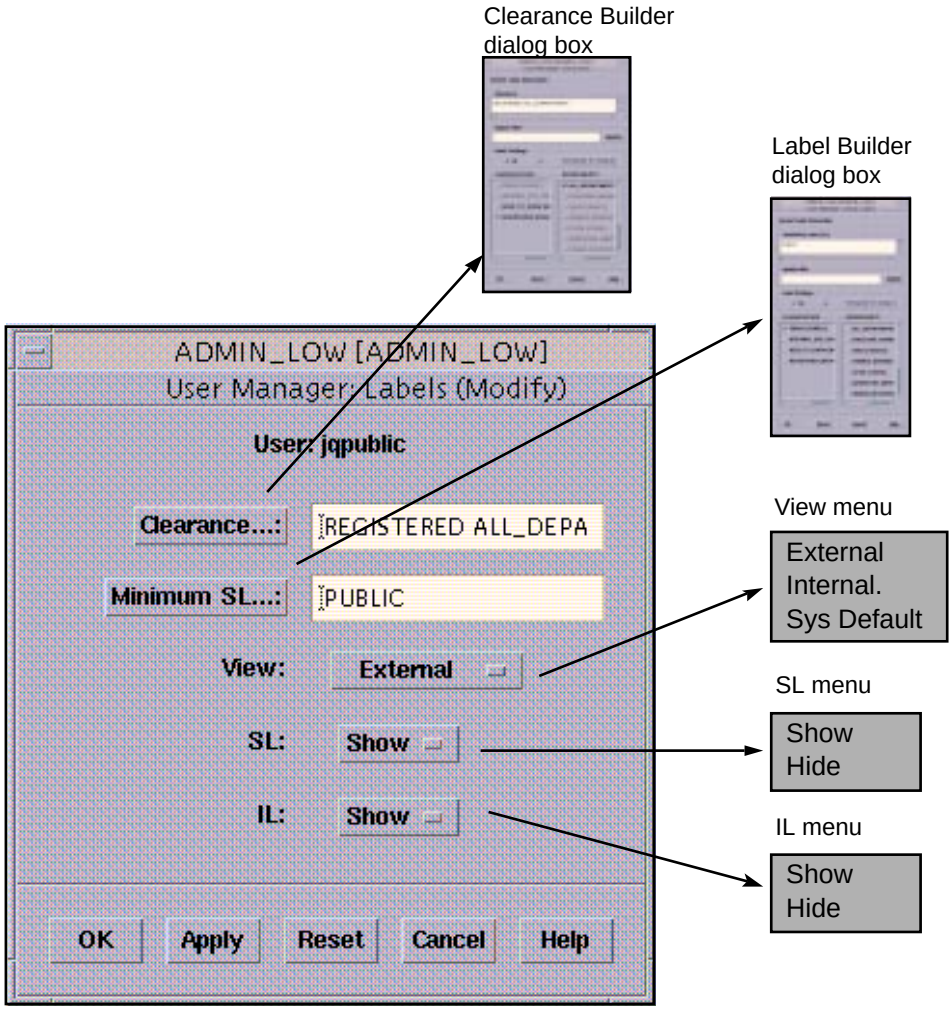


Figure 5–24 User Manager: Labels Dialog Box

1. **Click the Clearance button to bring up the Clearance Builder dialog box and enter the account's clearance.**

   The User Manager: Clearance label builder dialog box displays (see Figure 5–25.

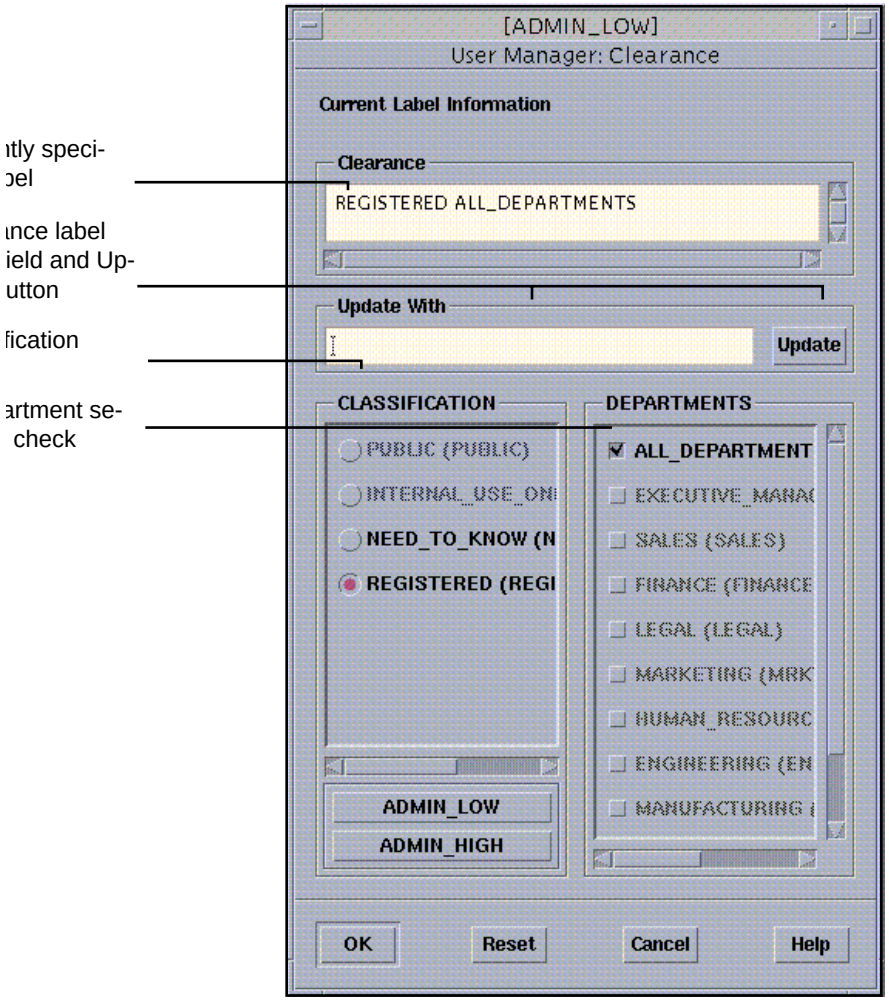


Figure 5–25 Label Builder for Setting the Account's Clearance

a. To type in the clearance, use the text entry field under Update With and select the Update button when you are done.

b. To use the mouse to build the clearance, click a button next to the desired classification in the menu and click none or more boxes next to the desired compartment names.

c. Click OK to close the label builder dialog box.
The clearance displays in the Clearance field in the Labels dialog box.

Note - The account's clearance must be dominated by the maximum clearance and must dominate the minimum clearance specified in the label_encodings(4TSOL) file.

---

2. **Click Minimum to enter the user's minimum sensitivity label.**

   The Minimum Sensitivity Label Builder dialog box displays (see Figure 5–26).

   a. **To type in the minimum sensitivity label, use the text entry field under Update With and select the Update button when you are done.**

   b. **To use the mouse to build the minimum sensitivity label, check a button next to one of the classifications in the menu and check none or more boxes next to the desired compartment names by using the left mouse button.**

   c. **Click OK to close the label builder dialog box.**

      The minimum sensitivity label displays in the Minimum SL field in the Labels dialog box.

Note - The minimum SL must be dominated by the maximum SL and must dominate the minimum SL defined in the `label_encodings` file.

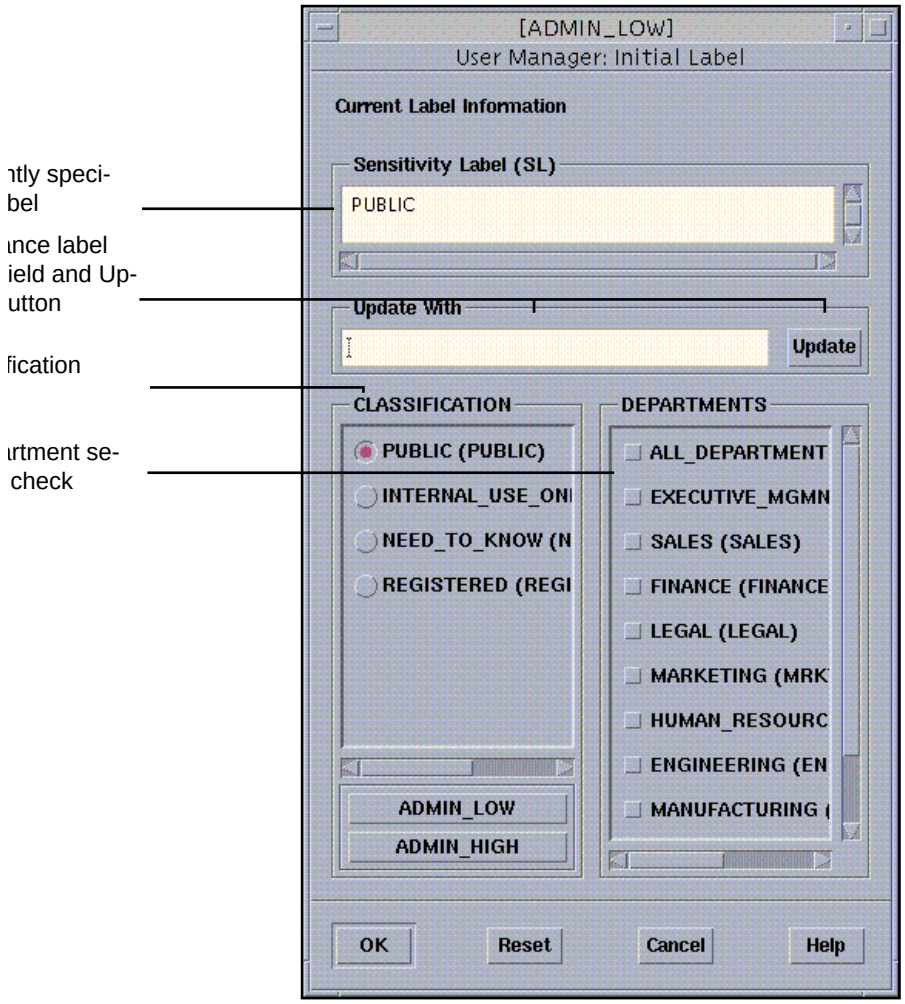


Figure 5–26 Label Builder for Setting the Minimum SL

3. **Choose the account's view of administrative labels from the options on the View menu.**
   a. **Choose External to substitute the maximum and minimum labels that are within the User Accreditation range instead of showing the ADMIN_HIGH and ADMIN_LOW administrative labels.**

   b. **Choose Internal to allow the user to see administrative labels ADMIN_HIGH and ADMIN_LOW or any site-configured alternative names.**

c. **Choose Sys default to use the system default label view that is specified in the label_encodings(4TSOL) file.**

Note - The display options in the View have no effect on SLs if you do not configure the account to be able to see SLS and they have no effect on ILs if either ILs are configured not to display system-wide, or if ILs are configured to display system-wide but you do not configure the account to be able to see ILs.

---

4. **Choose whether SLs are displayed for the account from the SL menu.**
   a. **Choose Show to allow SLs to be displayed.**
   b. **Choose Hide to hide SLs.**

5. **Choose whether ILs are displayed for the account from the IL menu.**
   a. **Choose Show to allow ILs to be displayed.**
   b. **Choose Hide to hide ILs.**

6. **Store your changes by clicking Apply or OK at the bottom of the Labels dialog box (see Figure 5–27).**
   Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

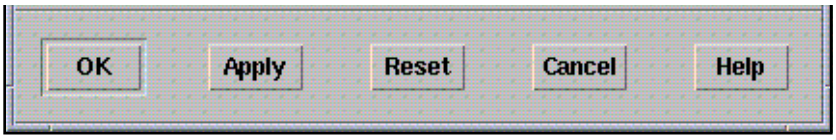


Figure 5–27 Controls on the User Manager: Labels Dialog Box

7. If you are done entering information for a new account or done updating an existing account, go to Step 11 on page 164.

8. Specify a new account's profile or profiles, or change the profiles selected for an existing account, if desired.

See "Profiles " on page 125, if necessary, to review the guidelines on what to specify in the Profiles dialog box.

a. Click Profiles on the User Manager Navigator.

The Profiles dialog box displays (see Figure 5–28).

The list at the left of the dialog box displays the available execution profiles that have not been selected. The list at the right of the dialog box contains the execution profiles that have been selected for this account.

b. Assign an execution profile to the account either by double-clicking its name in the Available list, or by single-clicking it and then clicking the right arrow to move it into the Selected list.

c. Remove execution profiles from the Selected list, if desired, either by double-clicking the name of the profile or single-clicking it and then clicking the left arrow to move it to the Available list.

d. To change the order of an execution profile in the Selected list, single-click on the name of the execution profile and then click the up- or down arrows to move it around in the list.

Note - As mentioned under “Profiles ” on page 125, keep in mind that the order of profiles can affect which command definitions are used, and take care to have the order reflect your true intention for a user’s use of a command.

---

e. Store your changes by clicking Apply or OK at the bottom of the Profiles dialog box.

Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

f. If you are done entering information for a new account or done updating an existing account, go to Step 11 on page 164.

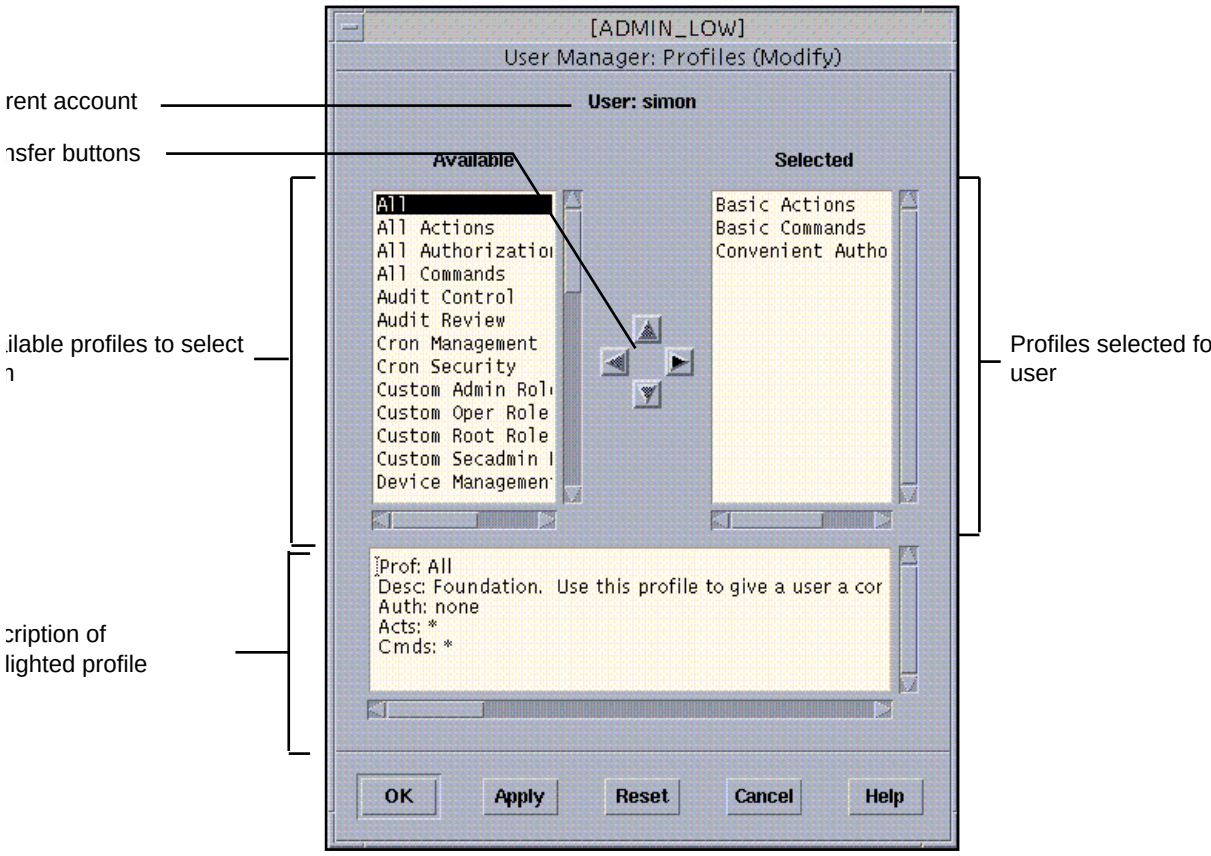


Figure 5–28 User Manager: Profiles Dialog Box

9. Specify any role(s) for the account in the Roles dialog box.

See "Roles" on page 126, if necessary, to review the guidelines on what to specify in the Roles dialog box.

a. Click Roles on the User Manager Navigator.

The Roles dialog box displays (see Figure 5–29).

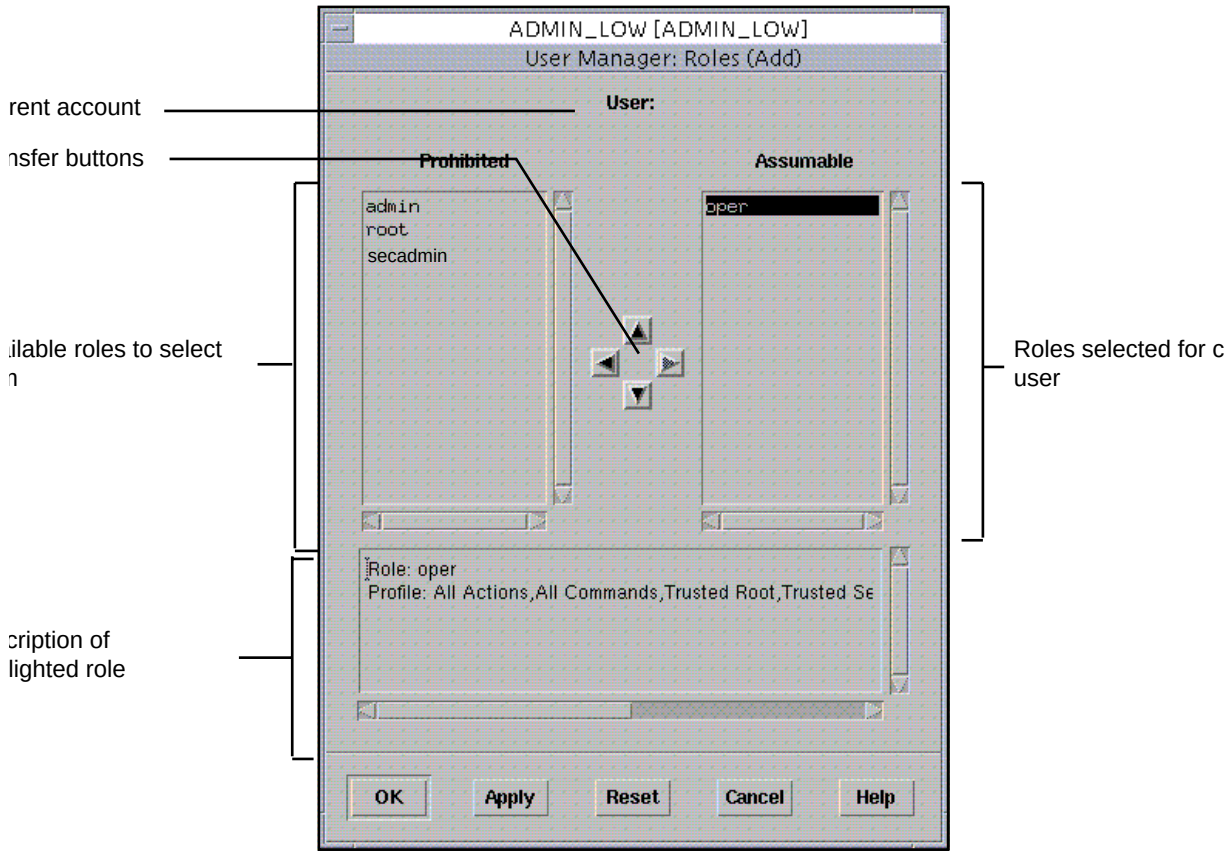


Figure 5–29 User Manager: Roles Dialog Box

Note - Roles cannot assume roles, so if the Account Type field in the Identity dialog box identifies this as a role account, the Roles button is dimmed on the navigator.

The list at the left of the dialog box displays a list of roles that have not been selected. The list at the right of the dialog box under Assumable contains the roles that have been selected for this account.

b. Assign a role to the account either by double-clicking its name in the Prohibited list, or by single-clicking it and then clicking the right arrow to move it into the Assumable list.

c. Remove roles from the Assumable list, if desired, either by double-clicking the name of the role or single-clicking it and then clicking the left arrow to move it to the Prohibited list.

d. Store your changes by clicking Apply or OK at the bottom of the Roles dialog box (see Figure 5–30).

Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

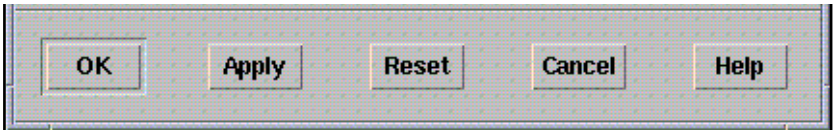


Figure 5–30 Controls on the User Manager: Roles Dialog Box

e. If you are done entering information for a new account or done updating an existing account, go to Step 11 on page 164.

10. Specify any idle limits and actions for the account in the Idle dialog box (optional).

See “Idle ” on page 127, if necessary, to review the guidelines on what to specify in the Idle dialog box.

a. Click Idle on the User Manager Navigator.

The Idle dialog box displays (see Figure 5–31).

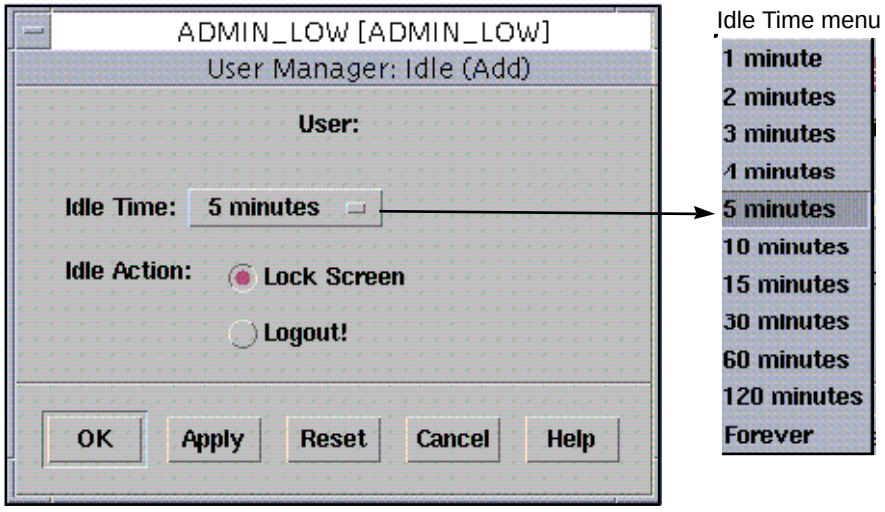


Figure 5–31 User Manager: Idle Dialog Box with Idle Time Menu

b. Choose from the Idle Time menu a length of time for the workstation to remain idle before an action is taken.

c. Click on either Lock Screen or Logout!.

Caution - If the logout action is selected, processes running in the user’s session are killed.

d. Store your changes by clicking Apply or OK at the bottom of the Idle dialog box (see Figure 5–32).

Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

Figure 5–32 Controls on the User Manager: Idle Dialog Box

e. If you are done entering information for a new account or done updating an existing account, go to Step 11 on page 164.

11. Click Done or Save on the User Manager Navigator.

Clicking Done in the Navigator saves what you've entered in all of the dialog boxes and closes the Navigator. Clicking Save saves what you've entered and leaves the Navigator open.

Figure 5–33 Controls on the User Manager Navigator

Note - If no password information has been saved, the account stays closed.

---

12. Choose the Exit option from the User Manager File menu to exit from all User Manager windows.

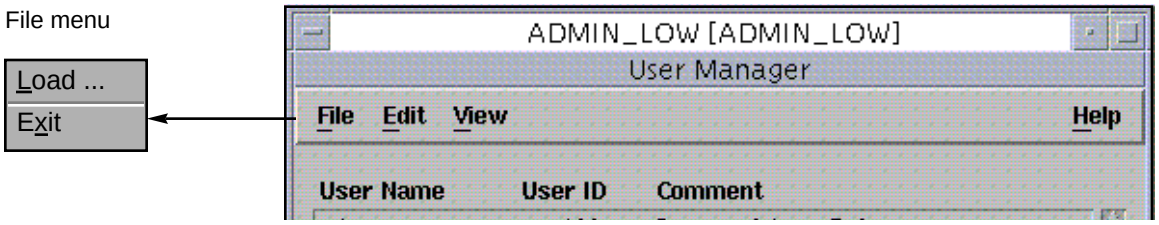


Figure 5–34 User Manager: Main Window and File Menu

CHAPTER 6

Managing Mail

Because mail is essentially the same in the Trusted Solaris environment as it is in the Solaris environment, the system administrator role sets up and administers mail servers according to instructions in Chapter 9 of the Solaris 2.5 *User Accounts, Printers, and Mail Administration* manual. The differences in how to administer Trusted Solaris mail are described here and on the sendmail(1MTSOL) man page. This chapter covers the following topics:

This chapter contains the following procedures:

- "To Configure Mail Delivery Options for Mail Below Users' Minimum Labels" on page 186
- "To Substitute an Alternate Mail Application in the Front Panel for All Users " on page 188
- "To Create a Multilevel Action for the Alternate Mail Application " on page 191
- "To Install an Alternate Mailer in the Front Panel" on page 193

Overview of Trusted Solaris Mail Features

In the Trusted Solaris environment, mail may be sent and received at multiple sensitivity labels by each user, and the sensitivity labels and information labels of mail messages are preserved. The following are the highlights of how mail is handled differently from the Solaris environment.

- MLDs are used to store mail messages queued for delivery and to store unread mail messages.

  See "Multilabel Directories for Outgoing and Incoming Mail" on page 169" and "Mailboxes in Multilabel Directories" on page 170.

- Sendmail is modified to enforce sensitivity labels, to float information labels, and to route mail to a user only if the sensitivity label of the incoming mail is within the clearance and minimum sensitivity label of the user.

-

- Because users should not be able to list queued mail sent by other users, the `restrictmailq` option is set by default in the `sendmail.cf` file, and 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 176.

- The `p` option in the `sendmail` configuration file, `sendmail.cf`, has been extended with `Optsol`... privacy option settings that allow administrators to specify what sendmail(1MTSOL) should do with any mail that arrives at a sensitivity label below the user's minimum sensitivity label.

  See "How Sendmail Handles Mail Below the Recipient's Minimum SL" on page 185 and "To Configure Mail Delivery Options for Mail Below Users' Minimum Labels" on page 186.

- An X-Sender-Information-Label line is added to the header of each message by `sendmail` to show the information label of the original sender.

  If an organization chooses to configure the system without information labels and, as a result, the `tsol_enable_il` switch is off in `/etc/system`, no X-Sender-Information-Label displays. [See system(4).]

Note - All the fields in the message header, including the X-Sender-Information-Label are vulnerable to manipulation or spoofing by an unscrupulous user, so the information label in the X-Sender line is not strictly reliable.

- A CDE action for an alternate mail application can be substituted for the default `dtmail` action in the front panel mail subpanel.

  See "Substituting an Alternate Mail Application" on page 187.

Multilabel Directories for Outgoing and Incoming Mail

The `/var/spool/mqueue` directory is an MLD where messages at varying sensitivity labels are stored until they can be delivered. Code Example 6–1 shows that when a user changes to `/var/spool/mqueue`, the user is transparently redirected into `.SLD.2`, which is the SLD whose sensitivity label is the same at the current process. The mail messages shown in `.SLD.2` are the ones waiting to be sent at the current sensitivity label.

Within the mqueue MLD, a listing of all the SLDs shows:

- `.SLD.1: Permission denied` (because the label of `.SLD.1` strictly dominates the current sensitivity label))
- `.SLD.0` is empty, and
- Messages in `.SLD.2` are waiting to be sent at the current sensitivity label.

CODE EXAMPLE 6–1 `/var/spool/mqueue` MLD and its Contents at Different Sensitivity Label

```
trustworthy% cd /var/spool/mqueue
trustworthy% mldpwd
/var/spool/.MLD.mqueue/.SLD.2
trustworthy% ls
dfNAA00212
dfNAB00212
dfNAC00212
trustworthy% cd /var/spool/.MLD.mqueue
trustworthy% ls -R .SLD.*
ls: .SLD.1: Permission denied
.SLD.0:
.SLD.2:
dfNAA00212
dfNAB00212
dfNAC00212
```

The Trusted Solaris software also makes `/var/mail` a multilabel directory (MLD) to store mail in mailboxes for all accounts. See "Mailboxes in Multilabel Directories" on page 170" for additional information.

Mailboxes in Multilabel Directories

The sendmail(1MTSOL) program creates mailbox files within the `/var/mail` MLD to store mail received for accounts. For each account, a separate mailbox is created at each of the sensitivity labels of any mail the account has received. As a result, each account may have multiple mailboxes: if user roseanne has received mail labeled PUBLIC, NEED TO KNOW ENGINEERING and NEED TO KNOW MARKETING, she has three separate mailboxes each at a different sensitivity label in a separate SLD. The mailboxes are given the name of the account. All of user roseanne's mailboxes are called *roseanne*. Role accounts have their own mailboxes, like all user accounts. Any mailboxes created for the role account secadmin are called *secadmin*.

The first time mail is sent to any user or role on a host at a particular sensitivity label, the `sendmail` program creates an SLD in `/var/mail` at the sensitivity label of the mail and stores the mail in a mailbox in the new SLD. Subsequent mail at the same sensitivity label whether it is for another user or the same user is put into the same SLD. When a user removes all the mail from a mailbox at a specific sensitivity label, the mailbox (`/var/.MLD.mail/.SLD.?/`*username*) is deleted.

For example, if the first mail received on the system has a sensitivity label of NEED TO KNOW ENGINEERING and is addressed to user roseanne, `/var/.MLD.mail/.SLD.0` is created at sensitivity label NEED TO KNOW ENGINEERING, and the mail is put into the `roseanne` mailbox in `.SLD.0`. If the next time mail is received on the system, it is also has the NEED TO KNOW ENGINEERING sensitivity label and is addressed to user jhoman, this second piece of mail is put into `.SLD.0` in the `jhoman` mailbox. A third piece of mail to roseanne at PUBLIC causes `.SLD.1` to be created; the mail is put into the `roseanne` mailbox in `.SLD.1`. The following figure shows that mail sent to roseanne and jhoman at NEED TO KNOW ENGINEERING is being stored in the `roseanne` and `jhoman` mailboxes in `.SLD.0`, while mail sent to user roseanne at PUBLIC is in another `roseanne` mailbox in `.SLD.1`.

CODE EXAMPLE 6–2 Mailboxes in SLDs at Different Sensitivity Labels

```
$ cd /var/.MLD.mail/
$ ls -R .SLD.*
.SLD.0
roseanne
jhoman
.SLD.1
roseanne
```

Mail Notification

Accounts are notified about mail arrival through the Mail subpanel on the Front Panel in whatever workspace is active. As illustrated in Figure 6–1, the Mail subpanel displays a mail icon for each of the following:

- For each sensitivity label at which mail has been received by the logged-in user account

- For each sensitivity label at which mail has been received by any role that the user has assumed during the current session.

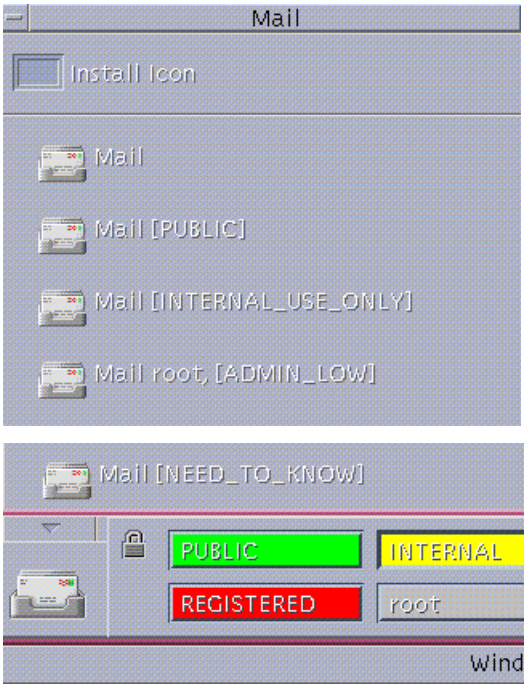


Figure 6–1 Mail Subpanel with Mail at Multiple Labels

Figure 6–1 shows the Mail subpanel displayed in a workspace with the INTERNAL_USE_ONLY sensitivity label. The Mail subpanel shows mail icons at PUBLIC, INTERNAL_USE_ONLY, and NEED_TO_KNOW for the user. The

The Mail subpanel in Figure 6–1 also shows mail at ADMIN_LOW for the root role, which the user has assumed during the session—as indicated by the presence of a root administrative role workspace button at the lower right of the figure.

The mail icon at the top of a subpanel is not identified with a sensitivity label. If clicked, it displays with the sensitivity label of the current workspace, just like the mail icon in the Front Panel.

Access to mail at various sensitivity labels is not constrained by the sensitivity label of the current workspace, because it does not violate the Trusted Solaris security policy for any user to be able to read and send mail at any sensitivity label for which the user is cleared. Therefore *a mail reader may be launched at any of the sensitivity labels of any of the icons displayed in the mail subpanel, no matter what the sensitivity label is of the current workspace.*

Reading of Mail

When the recipient of the mail clicks on a mail icon, a mail reader is displayed at the sensitivity label of the incoming mail. When a complete CMW label is displayed, the short name of the sensitivity label is displayed in brackets. Because the sensitivity label of the mail shown in the following figure is INTERNAL_USE_ONLY, the mail reader shows a sensitivity label of [INTERNAL] to the right of the PUBLIC information label.

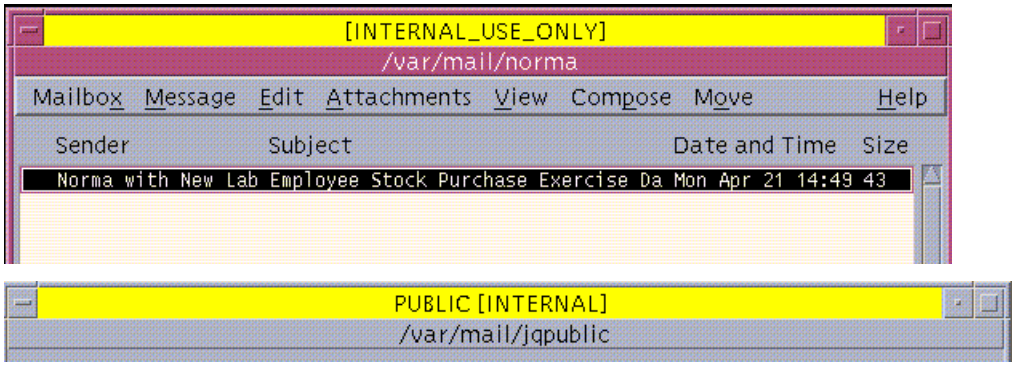


Figure 6–2 Window Label on a Mail Reader Launched at a Sensitivity Label of INTERNAL_USE_ONLY when ILs are Enabled

When information labels are disabled, the long name of the sensitivity label displays within brackets. The mail reader shown in Figure 6–3 displays with a sensitivity label of [INTERNAL_USE_ONLY].

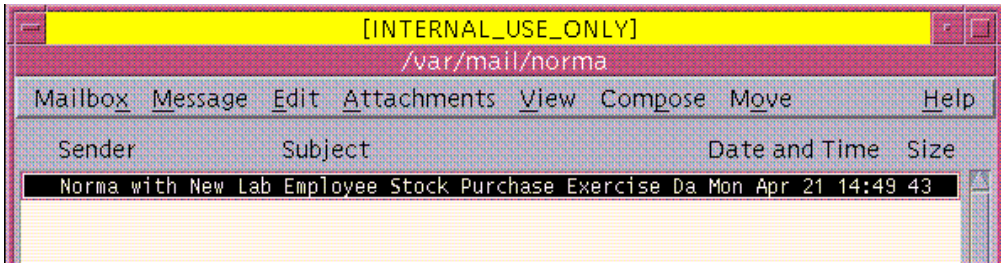


Figure 6–3 Window Label on a Mail Reader Launched at a Sensitivity Label of INTERNAL_USE_ONLY when ILs are Disabled

How Mail Gets its Sensitivity Label

The sensitivity label of the mail is determined by one of the following:

- For mail coming from a Trusted Solaris or from another type of labeled host configured with one of the labeled host options in the Trusted Networking

databases, the sensitivity label is the sensitivity label of the creating process (the sensitivity label of the mailer that sent the mail).

- For mail coming from an unlabeled machine (one that does not recognize labels and is configured as an unlabeled machine in the Trusted Networking databases), it is the default sensitivity label assigned to the host in the Trusted Networking databases.

Changing Mail Aliases

See the Solaris 2.5 "Mail Administration Guide" for how to administer mail aliases, along with the following sections that explain the exceptions that apply in the Trusted Solaris environment.

Enabling the Use of .mailrc Files in Home Directory MLDs

As in Solaris systems, individual users may put local copies of a `.mailrc` file into their home directories to create personal mail aliases. However, because Trusted Solaris home directories are almost always MLDs, the system administrator or individual user has to do some setup so that `.mailrc` files (and possibly other startup files) can be copied or linked into every SLD created within the user's home directory at each sensitivity label. See also "Managing Startup Files in a Trusted Solaris System" on page 71."

To partially automate the process of propagating `.mailrc` files (and other startup files) the system administrator can make use of standard skeleton directories and two new Trusted Solaris files explained below:

- `.copy_files` and
- `.link_files`

.copy_files

Create a `.copy_files` file and list the `.mailrc` file in it if you want the `.mailrc` to be copied to each SLD. (Copied startup files can be different in each SLD.) Copying the `.mailrc`, for example, would allow users to potentially have differing mail aliases in each SLD.

.link_files

Create a .link_files file and list the .mailrc file in it if you want the .mailrc to be linked to each SLD. Linked startup files are the same in each SLD. Linking the .mailrc, for example, would mean that any change made to the .mailrc in any SLD would be made in all SLDs.

Using the .copy_files and .link_files Along with Skeleton Directories

If the system administrator specifies a skeleton path directory for a user account, all the files in that skeleton directory are copied into each home directory SLD as it is created for the user. The system administrator can take advantage of this feature by putting into a skeleton directory (such as /etc/skel) a basic .mailrc along with a .copy_files that lists the .mailrc for copying or a .link_files file that lists the .mailrc for linking. Once this has been set up, and once the .copy_files or .link_files has been copied into the home directory SLD created at the user's minimum sensitivity label:

- If the .mailrc file is listed in .copy_files, it is copied into each subsequently-created SLD
- If the .mailrc files is listed in .link_files, it is linked into each subsequently-created SLD

The user may also create a .mailrc, .copy_files, or a .link_files or edit any existing files that have been propagated to any of their home directory SLDs after the system administrator's intervention.

Note - Do any setup for the .mailrc using .copy_files or .link_files you want to make immediately *before users have a chance to work at multiple sensitivity labels.* Otherwise, multiple home SLDs may be created without startup files being propagated.

"Controlling Which Startup Files Are Read by the Window System" on page 72 in Chapter 3.

▼ To Propagate a .mailrc to All Accounts' Home Directory SLDs

1. **Assume the system administrator role.**

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **In an ADMIN_LOW workspace, go to /etc/skel or whatever skeleton directory location you wish to use.**

3. **Put a basic .mailrc file into the skeleton directory.**

4. **To make the `.mailrc` copied to all home directory SLDs, create a `.copy_files` or modify it, if it already exists, to list the `.mailrc`.**

5. **To make the `.mailrc` linked to all home directory SLDs, create a `.link_files` or modify it, if it already exists, to list the `.mailrc`.**

6. **When setting up user accounts, if you want to support the use of personal mail aliases enter the pathname of the skeleton directory in the Home dialog box on the User Manager.**

Creating and Initializing New Local and NIS+ Managed Aliases

Aliases may be specified in NIS+ maps or in the `/etc/aliases` file on each host. The Database Manager is used to update aliases. The Naming Service menu on the Database manager allows the system administrator to choose between NIS+ or none when editing any database, including Aliases.

▼ To Edit Aliases

1. **Assume the system administrator role, go to an ADMIN_LOW workspace.**
   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **Bring up the Database Manager: Aliases Database**
   See "To Launch Solstice Administration Tools" on page 27, if needed.
   a. **Highlight Aliases on the Databases scrolling list.**
   b. **From the Naming Service menu, pick NIS+ to edit the NIS+ aliases map or pick None to edit the local version /etc/aliases file.**
   c. **Click the OK button.**
      The Database Manager: Aliases Database displays.

3. **To modify an alias, select the name of an existing alias and modify it.**
   a. **Highlight an alias name.**
   b. **Choose Modify from the Edit menu.**
      The Database Manager: Modify dialog displays.
   c. **Change the text in the Expansion text entry field as desired.**
   d. **Click the OK button when done.**

4. **To add an alias, select Add from the Edit menu.**
   a. **Enter the alias name in the Alias text entry field.**
   b. **Enter all addresses for the alias in the Expansion text entry fields.**
   c. **Click the OK button when done.**
5. **Choose Exit from the File menu of the Database Manager.**

Allowing Users to List the Entire Mail Queue

In the default configuration, users are not allowed to list queued mail sent by other users. In the Solaris versions of in the `/etc/mail/sendmail.cf` file, the `restrictmailq` option is available to be used but it is not set by default. In the Trusted Solaris implementation, `restrictmailq` is set by default in the `sendmail.cf` file, and only users in the same group as the mail queue may list jobs in the mail queue.

Note - Even if an administrator performs one of the steps described below to allow listing of the mail queue, only users who have the `mailq` or `sendmail` commands in one of their profiles can list the mail queue—by entering either `mailq` or sendmail(1MTSOL) with the -bp option. Also, these commands show mail only at labels dominated by the user's process.

▼ To Allow Listing of the Mail Queue

1. **To allow all users to list the mail queue, remove the `restrictmailq` option in the `/etc/mail/sendmail.cf` file.**
2. **To allow a group of users to list the mail queue:**
   a. **Create a new group.**
   b. **Add the desired set of users to the group.**
   c. **Use chgrp(1TSOL) to change the group of `/var/spool/mqueue` to the new group.**

3. **Add the** `mailq` **and** `sendmail -bp` **commands to a profile and assign the group of the** `/var/spool/mqueue` **to those commands.**

Tracing Sendmail's Activities

Multiple instances of `sendmail` are involved in local and remote mail delivery. Figure 6–4 shows how data flows through the `sendmail` processes. Any mailer used to send mail (the default is `dtmail`) starts an instance of `sendmail`. This instance of `sendmail` attempts to deliver 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 [¿ in Figure 6–4]. 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 a long time.

An instance of the `sendmail` program 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 Figure 6–4]. A third instance of `sendmail` periodically scans the mail queue and attempts to deliver any mail in the queue 2 and 4 in Figure 6–4]. 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..

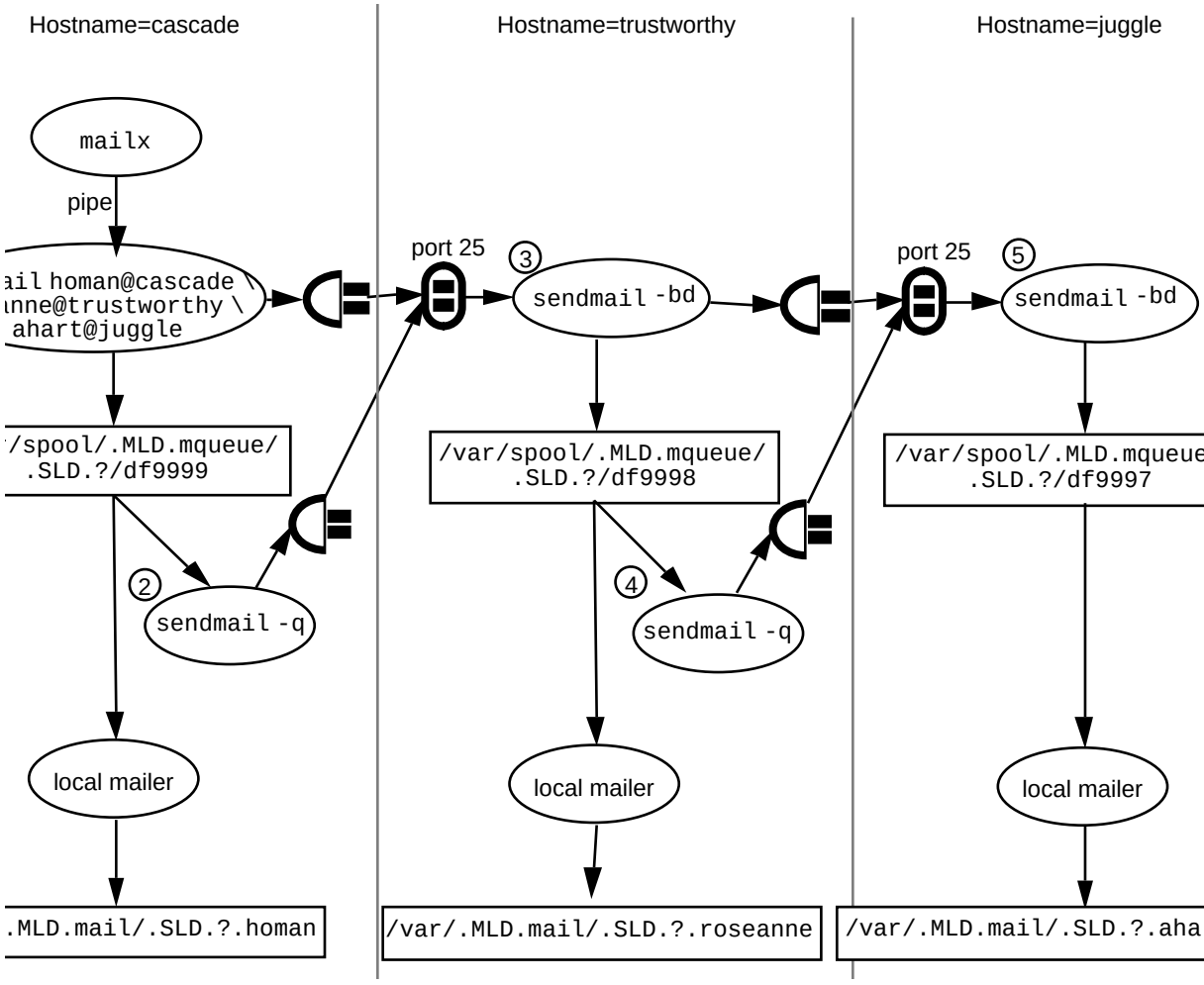


Figure 6–4 Sendmail Data Flow Example

When you send mail to *username@hostname* and *hostname* is the name of a remote host, `sendmail` forwards the message to port 25 of *hostname.* As shown in Figure 6–4, 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 `sendmail -d` is described in detail in the `sendmail` Nutshell Handbook put out 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. A new category, 75, selects Trusted Solaris debugging information.

▼ To Trace Sendmail for Trusted Solaris Information

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

   See "To Login and Assume an Administrative Role" on page 15 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` followed optionally by a dot (.) and a level, followed by a space and the address, followed by a message.**

   The message can either be included 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
```

Troubleshooting Mail Delivery Difficulties

Remember that sendmail(1MTSOL) 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 within that account's sensitivity label range (between the account's clearance and minimum sensitivity label). The account's sensitivity 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 10."

- Multilevel hosts with an accreditation range between ADMIN_LOW and ADMIN_HIGH can receive mail at all sensitivity labels,
- Multilevel hosts that have a restricted accreditation range can receive mail only between their maximum and minimum sensitivity labels
- Single-level (unlabeled) hosts can receive mail at only the single sensitivity label set up for them.

If a user is having trouble sending mail, use the following as guidelines for where to look:

♦ **Check the mail aliases.**

Remember that the `/etc/aliases` file and the `mail_aliases` NIS+ map are consulted by `sendmail` in determining where to deliver mail. For example, mail to `fred` from a system 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` for user fred.

♦ **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 180.

▼ To Check for a Properly Configured Network Connection for Sending Mail

1. **Send mail using** `mailx`**.**

```
# mailx -v somebody@somehost
   Subject: test1
   testl
```

.

Look at the messages from `mailx`. If you see a line that says, message accepted, go to Step 3 on page 181.

2. **While logged into the sending host or, if the mail server is not the same as the sending host, while logged into the mail server, at the sensitivity label at which the user needs to send mail, use the telnet(1) command to connect to port 25 of the receiving host.**

```
trustworthy% telnet hostname 25
```

If the connection is 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 you get an error message from `telnet`, the connection is not properly set up with the correct labels in the trusted networking databases; go to Step 5 on page 181 and following and do the step that applies to the type of host you are trying to debug. If the connection seems to be set up properly, go to Step 3 on page 181.

3. **At the sensitivity 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.**

Check the list to see if the mail is stuck on the mail server.

```
# mailq | more
```

4. **Try the procedure under "To Trace Sendmail for Trusted Solaris Information" on page 179.**

5. **If the destination host is a Trusted Solaris 2.x system, 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 by using the User Manager.**

   b. **Note the account's minimum label and clearance in the Labels dialog in the User Manager.**

c. Make sure that the sensitivity label of the mail being sent is dominated by the recipient's maximum label and dominates the recipient's minimum label.

i. If the sensitivity label of the mail being sent is not in the recipient's account label range, if you can find a mutually-acceptable sensitivity label for the sender and the recipient, change the sensitivity label to one within the destined recipient's label range and try again.

ii. If the mail goes through, instruct the sender to send mail to that recipient at the mutually-acceptable label.

a. Make sure that the sensitivity 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(4TSOL) file.

`sendmail` does not deliver mail if the sensitivity label of the mail is outside the System Accreditation Range.

If the sensitivity 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 5 on page 182.

b. If the mail is below the minimum label of the destined user, do any of the following steps, if it is consistent with your site's security policy.

See "How Sendmail Handles Mail Below the Recipient's Minimum SL" on page 185 and "To Configure Mail Delivery Options for Mail Below Users' Minimum Labels" on page 186.

i. To enable all users on the system to receive mail that is below their minimum labels, either make sure that `sendmail` is automatically upgrading the mail by specifying `tsolotherlowupgrade` (the default), or make sure the mail reader delivers the mail at the incoming label, which allows the upgrade if the account has the needed authorization, by specifying `tsolotherlowaccept`.

ii. To allow the normal user accounts of all employees who are able to assume administrative roles to receive mail from system processes outside the User Accreditation Range, make sure that the `tsoladminlowupgrade` or `tsoladminlowaccept` options are set in the `sendmail` configuration file.

iii. To enable one administrative role to receive mail from system processes outside the User Accreditation Range, use the Profile Manager to make sure that the administrative role has the *use all defined labels* authorization.

(The default administrative roles have that authorization in their profiles.)

6. **For a destination host running the Trusted Solaris 2.5 software, on the sending host make sure that the `tnrhdb`/`tnrhtp`(4TSOL) entries for the receiving host are configured properly so the TS2.x system can communicate with that host over the network.**

Note - You can use the tninfo(1MTSOL) 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(4TSOL) database and that the template in the tnrhtp(4TSOL) file correctly defines the host's type as `sun_tsol`.**

   b. **Check that the minimum and maximum sensitivity label set in the assigned template in the tnrhtp(4TSOL) allow communications at the sensitivity 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 2 on page 181 on Step 2 on page 181 and run telnet(1) to make sure.**

7. **For a destination host running the Trusted Solaris 1.x software, on the sending host make sure that the `tnrhdb/tnrhtp` entries for the receiving host are configured properly so the TS2.x system can communicate with that host over the network.**

   a. **Check that the destination host has the correct template name assigned to it in the tnrhdb(4TSOL) database and that the template in the tnrhtp(4TSOL) file correctly defines the host's type as `msix`.**

   b. **Check that the minimum and maximum sensitivity label set in the assigned template in the tnrhtp(4TSOL) allow communications at the sensitivity 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 2 on page 181 on Step 2 on page 181 and run `telnet` 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/tnrhtp` entries for the receiving host are configured properly so the TS2.x system can communicate with that host over the network.**

a. **Read the tnrhtp(4TSOL) 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 10."

b. **Create a template or copy an applicable one in the tnrhtp(4TSOL) and make sure that the correct template is assigned to the host in the tnrhdb(4TSOL) database to identify it with the appropriate host type.**

c. **Check that the minimum and maximum sensitivity label set in the assigned template in the tnrhtp(4TSOL) allow communications at the sensitivity 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 2 on page 181 on Step 2 on page 181 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 so the TS2.x system can communicate with that host over the network.**
   a. **Check that the destination host has the correct template name assigned to it in the tnrhdb(4TSOL) database that the template in the tnrhtp(4TSOL) file correctly defines the host's type as** `unlabeled`**.**

   b. **Check that the sensitivity label defined as the default sensitivity label for the unlabeled host in the assigned template in the tnrhtp(4TSOL) allows communications at the sensitivity 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 2 on page 181 on Step 2 on page 181 and run** `telnet` **to be sure.**

Configuring Trusted Solaris Mail Delivery Options for Mail Below Users' Minimum Labels

The security administrator must make sure that the Trusted Solaris-specific privacy options in the `sendmail` configuration file `sendmail.cf` have values consistent with the site's security policy. See the sendmail(1MTSOL) man page, especially the TRUSTED SOLARIS DIFFERENCES section, and see the paragraphs that follow for a description of what the options do.

How Sendmail Handles Mail Below the Recipient's Minimum SL

Two types of actions may be specified for `sendmail` to take when mail is received at a sensitivity label that is below the recipient's minimum sensitivity label. The `Optsol`... privacy option settings in the `sendmail` configuration file specify the actions to take, depending on whether the mail is at ADMIN_LOW or at another label below the recipient's sensitivity label. 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 is SECRET or NEED TO KNOW.

These options give sites the discretion to decide which response is consistent with their site's security policy. The defaults (as set in `/etc/mail/sendmail.cf`)automatically upgrade mail sent at ADMIN_LOW and return mail sent at other labels below the user's minimum sensitivity label.

Mail Handling Options

Three mutually exclusive options have the prefix *tsoladminlow* and the three other mutually exclusive options have the prefix *tsolotherlow*. Each option has an action portion in its name to specify what `sendmail` should do when it receives the mail at ADMIN_LOW or some other sensitivity label below the user's minimum sensitivity label. *upgrade* means to deliver the message at the recipient's minimum sensitivity label. *accept* means to deliver the message at the message's sensitivity label. *return* means to return the message to the sender.

TABLE 6–1 Trusted Solaris Mail Handling Options

Option Name	Effect
`tsoladminlowupgrade`	Default setting. Accepts ADMIN_LOW mail and delivers it at the user's minimum SL.
`tsoladminlowaccept`	Accepts ADMIN_LOW mail and delivers it at ADMIN_LOW. User may upgrade it and read it only if one of the user's execution profiles has the SYS_ACCRED_SET authorization.
`tsoladminlowreturn`	Returns ADMIN_LOW mail to the sender
`tsolotherlowupgrade`	Upgrades mail received at a SL below the user's minimum SL to the user's minimum SL. Because the mail is upgraded by `sendmail`, the user does not need the SYS_ACCRED_SET authorization to receive it.
`tsolotherlowaccept`	Accepts mail below the user's minimum label and delivers it. User may upgrade and read it only if one of the user's execution profiles has the SYS_ACCRED_SET authorization.
`tsolotherrlowreturn`	Default setting. Returns mail below the user's minimum label 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 Login and Assume an Administrative Role" on page 15, if needed.

2. **Use the Set Mail Options action to open the `sendmail.cf` file for editing.**

   See "To Use the Admin Editor Action to Edit a File" on page 29, if needed.

3. **Search for the lines that begin `Optsol`, and change either of the two existing default settings.**

   See Table 6–1, "Table 6–1" for names and descriptions of the tsol privacy options.

```
# TSOL actions for mail received below recipient min label
   #    options are:
   #         tsoladminlowupgrade - upgrade to user min label (default)
   #        tsoladminlowaccept  - accept at delivered label
   #        tsoladminlowreturn  - return to sender
   #        tsolotherlowupgrade - upgrade to user min label
   #        tsolotherlowaccept  - accept at delivered label
   #        tsolotherlowreturn  - return to sender (default)
```

```
Optsoladminlowupgrade
Optsolotherlowreturn
```

Substituting an Alternate Mail Application

By default, `dtmail` is the mail application that is launched from the mail panel the Trusted Solaris Front Panel. The Trusted Solaris system allows the substitution of an alternate mail application, but only the system administrator 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 sensitivity label of the current workspace. However, since mail monitoring for mail at multiple sensitivity 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 sensitivity label.

The system administrator 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 188”).
- 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 191”).

For an alternate mail action to be installed in the front panel, the mail application must have an action defined. The example screens in “To Substitute an Alternate Mail Application in the Front Panel for All Users ” on page 188 show the substitution of the OpenWindows `mailtool` for `Dtmail`. The OpenWindows `mailtool` action is defined in the `/usr/dt/appconfig/types/C/sunOW.dt` file as shown in Code Example 6–3.

CODE EXAMPLE 6–3 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
}
```

Tip

Note - Difficulties may arise if mail arrives while you are installing an alternate mail icon or deleting the default one. For that reason, it is a good idea to stop sendmail before you start and start sendmail again after you are done.

If all the mail icons disappear from the Front Panel, investigate the account's $HOME/.dt/fp.dynamics 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 would need to clean up the contents of directories created by the window system in every $HOME/.dt/fp.dynamics directory.

1. **Assume the administrator role, and go to an ADMIN_LOW workspace.**
   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **In a profile shell at ADMIN_LOW, go to the** /etc/rc2.d **directory and stop** sendmail**.**

```
$ cd /etc/rc2.d
  $ S88sendmail stop
```

3. **Log out.**

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

5. **Make sure an action is defined for the alternate mail action.**
   See "Substituting an Alternate Mail Application" on page 187."

6. **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 29, if needed.

7. 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
     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        FPOnItemMail
   HELP_VOLUME        FPanel
```

8. 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.

---

9. Remove all contents of the $HOME/`.dt/fp.dynamics` directory.

10. Restart the Workspace Manager from the workspace menu to see the changes to the `dtwm.fp` go into effect in the front panel.

11. Assume the administrator role and make sure you are in an ADMIN_LOW workspace.

12. In a profile shell, go to the `/etc/rc2.d` **directory and restart** `sendmail`**.**

```
$ cd /etc/rc2.d
   $ S88sendmail 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 Login and Assume an Administrative Role" on page 15, 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 29, 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
     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. **C*opy* the control text to a file whose name has the** `.fp` **extension, for example,** `mail.fp`**, and quit the** `dtwm.fp` **file.**
   Create the new file `mail.fp` file in a directory such as `/etc` or `/usr/bin` that is in every user's path.

```
:wq
```

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
   }
```

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        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 193, after testing the procedure yourself to see if the mailer shows up and works properly.

▼ To Install an Alternate Mailer in the Front Panel

1. Obtain the pathname of the correct alternate mail application's control file from the system administrator.

The system administrator must have completed some setup in order for the mailer to watch for and notify you about mail at all labels within your personal accreditation range. See “To Create a Multilevel Action for the Alternate Mail Application ” on page 191.

Note - Do not install an alternate mailer if the file does not end with a suffix of `.fp`.

Caution - Unless you have the approval of your site’s security administrator, do not install an alternate mailer from any of the application manager folders or if the file does not end with a suffix of `.fp`.

---

2. **Ask the administrator 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 administrator to restart** `sendmail`**.**

CHAPTER 7

User Manager Data Collection Worksheet

Note - In the User and Role Account Worksheet on the following page, the names in the left column correspond to the labels on the buttons on the User Manager Navigator.

User or Role Account Worksheet

Full Name and Function of Individual or Role	
Primary Host Name	
Domain Name	

Identity	Login name: UID: Primary GID: Secondary GIDs: >
	Comment:
	Shell: [ ]Bourne [ ]C [ ]Profile [ ]Other:
	User Type: [ ]Normal [ ]Administrative Role [ ]Non-administrative Role
Home	Create home dir automatically?
	Home directory path:

	Skeleton path:
	Default permissions:
	Mail server:
	AutoHome setup:
Password	[ ]Account is locked [ ]No password (setuid only) [ ]Type in … [ ]Choose from >
	Minimum days between password change:
	Maximum days before change must be made:
	Maximum days account can be inactive:
	Expiration date: Warning (days before expiration date):
	Change by (method used by account when changing password): [ ]Choose from list [ ]Type in
	Account status [ ]Open [ ]Closed [ ]Always Open
	NIS+ credential table setup?
Idle	Idle time : [ ]Minutes before idle action [ ]Forever
	Idle action: [ ]Logout [ ]Lock Screen
Labels	Clearance:
	Minimum sensitivity label:
	View : [ ]Internal (replace names of administrative labels with closest equivalents within the user accreditation range) [ ]External (show actual administrative labels) [ ]SysDefault
	Sensitivity Labels: [ ]Show [ ]Hide; Information Labels: [ ]Show [ ]Hide
	Information Labels: [ ]Show [ ]Hide
Profiles	See Appendix A” for names of profiles and the tools they contain.
Roles	[ ]Security Administrator [ ]System Administrator [ ]Operator [ ]Root

CHAPTER 8

Managing Execution Profiles for Users and Roles

This chapter describes how to modify or add execution profiles, which are used for defining the capabilities of individual users and administrative roles. It includes the following major topics.

This chapter includes the following procedures:

Review of Terms

This section reviews some Trusted Solaris concepts and terms related to execution profiles and introduces some new ones. The reviewed topics were introduced in the *Trusted Solaris User's Guide* and discussed in the *Trusted Solaris Administration Overview* and are included here for convenience.

Execution profiles

Execution profiles are a bundling mechanism for defining the capabilities of individual users and administrative roles. Each execution profile potentially contains a list of UNIX commands, CDE actions, and authorizations. An execution profile may also associate security attributes (an *effective UID* and *effective GID*, *inheritable privileges*, or *label constraints*) with each command and action. One or more execution profiles are assigned to each user and role. The order in which execution profiles are specified is significant. When multiple profiles are assigned to a user or role, the set of authorizations is combined, and whatever command or action appears first in the list of profiles for that account is used with the privileges, UID, and GID as specified for its first appearance. Execution profiles can be used to restrict users to a limited set of applications or commands. They also can be used to extend the capabilities of an account, especially when used in combination with the role mechanism. See *enabling attributes* and *restrictive attributes*.

Effective UID and GID

A process executing a command has an effective user ID (EUID) and effective group ID (EGID) equal to the process' real UID and real GID, unless the process has executed a command that has the set user ID (setuid) bit or set group ID (setgid) bit set. Setting an effective UID, an effective GID, or both an effective UID and GID, by means of the setuid or setgid bit is usually done when a command performs checks while it is running to ensure that it is being run by root (UID 0). Setting the setuid or setgid bit on a command or action allows a non-administrative user to do some things that only the superuser could do in the base system. To achieve the same

effect in the Trusted Solaris system, the security administrator role can assign an effective UID or effective GID to a command or action in an execution profile when the command or action must be run by a specific user or group, most often when the command must be run as root.

Actions

An action is a bundling mechanism that allows one or more commands to be specified for a particular task that cay be assigned to one or more users. An action can have a set of options and arguments specified along with each of the command(s) and may make use of a dialog box to prompt for additional arguments.

A set of administrative actions have been created to edit files that are not managed through NIS+. For example, to edit the `vfstab_adjunct` file, the security administrator uses an action called Set Mount Attributes. The Set Mount Attributes action includes a command (`adminvi`), an argument (`vfstab_adjunct`), a label range (ADMIN_LOW to ADMIN_HIGH, and an effective UID of root and an effective GID of sys. The label range is used to ensure that the `vfstab_adjunct` file is not edited at any other SL. The effective UID and effective GID ensure access to the file and ensure that the file's owner and group do not get changed in the process of being edited.

Each action usually has its own icon, is assigned its own set of security attributes, and may be specified in an execution profile. The icons in the Application Manager and most of the icons in the front panel correspond to actions.

Adding actions is described in the *CDE User's Guide*, and the *CDE Advanced User's and System Administrator's Guide*, but certain aspects of Trusted Solaris security policy affect who can add actions, and who can use the actions. Normal users can drag and drop actions from the Application Manager to the front panel to run at a single sensitivity label. The system administrator can register actions and put them in the `/etc/dt/appconfig/types/C` directory to make them available for everybody. See "Actions" on page 494 in Chapter 16," for more about actions and how to add them.

Enabling Attributes

The optional set of override privileges, effective user ID, and effective group ID that may be associated with commands and actions in execution profiles extends the capabilities of the command or action, enabling the account that invokes it to:

- Work outside of the Trusted Solaris security policy
- In a controlled way
- For a well-defined purpose

The UID, GID, and privileges are referred to as *enabling attributes.*

Restrictive Attributes

The label range is referred to as a *restrictive attribute*, because it limits the range of labels at which the command or action may be invoked. The default label range is the full range in the system accreditation range, from ADMIN_LOW to ADMIN_HIGH.

Privileges in Profiles

When a command or an action is assigned one or more privileges in an execution profile, the privileges become available to the command or action through *privilege inheritance*. Privileges are considered to be *enabling attributes*. Commands inherit privilege only when invoked in a profile shell, pfsh(1MTSOL). Actions inherit privileges through the window system, through the same database lookup used by the profile shell.

Using privilege inheritance allows finer controls on who can make use of privilege. Compare the use of inherited privileges to the use of forced privileges for a command. Forced privileges are associated with an executable file, so forced privileges are in effect for the command no matter who invokes the command and no matter what shell is used. In contrast, inherited privileges are available only when a user or role executes the command in a profile shell and only when one of that account's execution profiles has the command specified to run with privilege.

Background on Execution Profiles

Adding and modifying profiles is an advanced topic for sites that want to control what users can do and enhance what administrators can do—beyond the controls provided in the default system.

The Trusted Solaris default set of execution profiles includes:

- Basic profiles for normal users
- Three administrative role profiles and one non-administrative role profile

Divided among the default role profiles are all the commands, actions, authorizations, and privileges necessary to fulfill both the administrative responsibilities of the UNIX system administrator (root/superuser) and the added administrative responsibilities of managing a Trusted Solaris system.

The default execution profiles may be all your site ever needs to operate fully and with the full measure of trust. However, once the site's security administrator is thoroughly familiar with the execution profiles and how they work, the security administrator may decide to create new execution profiles, based on the site's answers to the following questions:

- What tasks need to be accomplished system-wide and by each individual?
- What is each individual capable of?
- What information and responsibilities can each individual be trusted with?

Note - Trusted Solaris documentation sometimes uses the term profile and execution profile interchangeably. The term execution profile is stressed to avoid any confusion with the installation profiles that may be created during installation and used as part of system jumpstart.

Before making any changes to execution profiles, especially before you create any new administrative role profiles or modify the administrative role profiles that currently exist, make sure you thoroughly understand how the current roles are configured—as described in tables in the *Trusted Solaris Administration Overview*—and understand what the security implications might be of any changes or additions you might decide to make. The tables list the commands, actions, and authorizations assigned to each profile, and the security attributes (UID, GID, privileges, and label range) that apply to each command and action.

Use of the Profile Manager to Create or Modify Execution Profiles

In the default configuration, the Profile Manager is assigned only to the security administrator role. The Profile Manager is used at ADMIN_LOW.

When the Profile Manager is launched (as described in “To Access the Profile Manager ” on page 233), the Profile Manager Load dialog box displays as shown in the following figure. The Profile Manager Load dialog box has menus for picking a naming service and filtering profiles. Filtering profiles allows you to start with an existing profile loaded into the Profile Manager or to start with the Profile Manager empty.

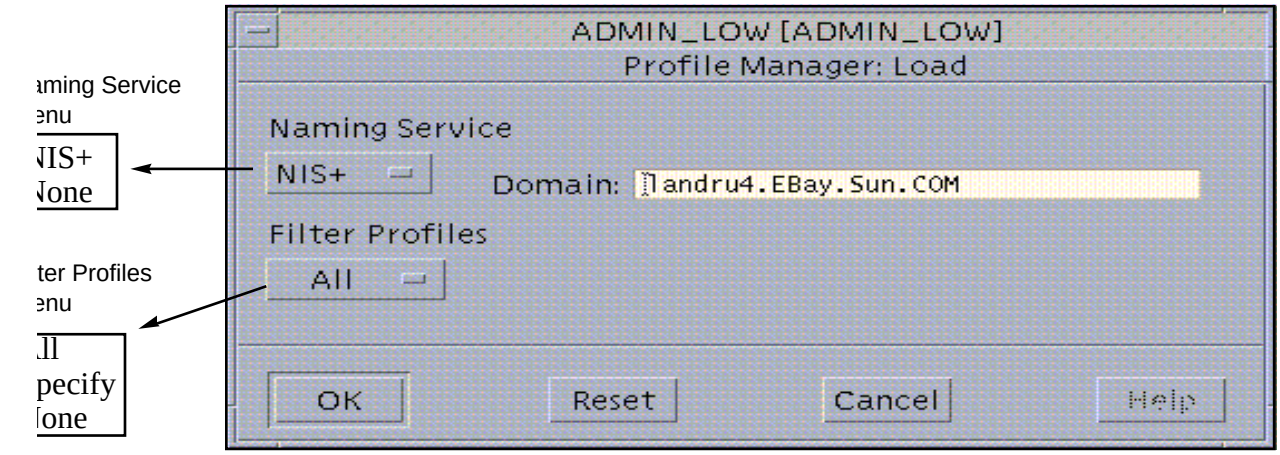


Figure 8–1 Profile Manager: Load Dialog Box profmgr.load.rs

Using the Control Buttons on the Profile Manager Dialog Boxes

The control buttons shown in Figure 8–1 are also found on the Set Privileges and Set UID/GID dialog boxes.

OK

OK saves any changes and closes the dialog box.

Reset

Restores the fields to their original state after the last save and leaves the dialog box open.

Cancel

Cancel closes the dialog box without saving any changes

Help

Help provides context sensitive help messages.

Picking a Naming Service

Like all other Trusted Solaris AdminSuite tools, the Profile Manager requires that you pick a naming service. The default is NIS+, which is the recommended mechanism for centralized distribution of user and host information. The None option should only be used in specialized circumstances where a knowledgeable security administrator decides that local profiles on individual hosts are both needed and allowable within the site's security policy. A standalone Trusted Solaris host may use either naming service.

If you choose NIS+ for the Naming Service, any modifications you make to the set of execution profiles are stored in the NIS+ `tsolprof` table for the specified domain when the Profile Manager Load dialog first displays. The current domain name appears in a text field next to the NIS+ option, as shown in the following figure. If the secadmin role has been authenticated for another domain, you may update the NIS+ profiles table in that other domain by replacing the domain name with that of the other NIS+ domain.

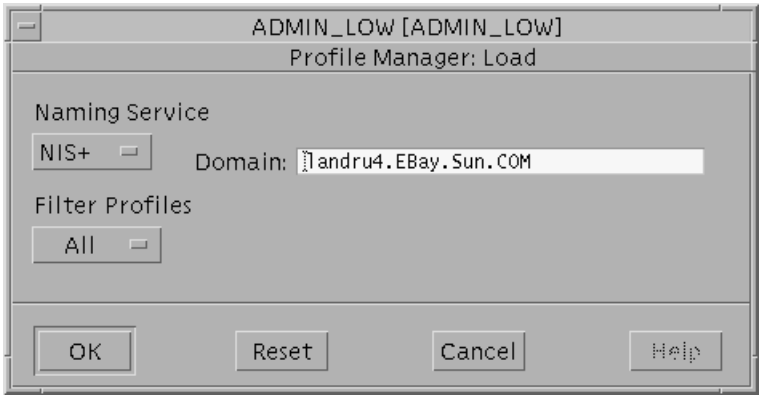


Figure 8–2 Profile Manager: Load, Naming Service NIS+ profmgr.load.NIS+.rs

If you choose None for the Naming Service, the name of the local host displays as shown in the following figure. You may change the name in the Host field. Any modifications you make to the set of execution profiles are stored in the `/etc/security/tsol/tsolprof` file on the specified host.

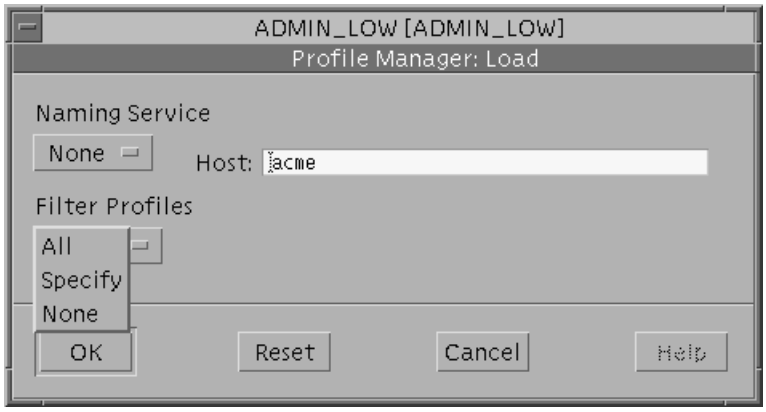


Figure 8–3 Profile Manager: Load, Naming Service None profmgr.load.none.rs??

Decision To Make Before Starting

- **Decide whether to use NIS+ or no naming service.**

Filtering Profiles

The Filter Profiles menu shown in Figure 8–4 allows you to specify whether the Profile Manager should come up empty or come up loaded with a profile that you specify. Which of the Filter Profile menu options you choose depends on whether you are adding a new profile or modifying an existing profile. See “When Adding a New Profile” on page 205” or “When Modifying an Existing Profile ” on page 206.

Note - Whichever option you choose to start, after the Profile Manager comes up you may display another profile or empty the display by using the Profiles menu, as described in “Bringing Up a Blank Profile Definition, Loading an Existing Profile, or Saving Changes Within the Profile Manager” on page 214.

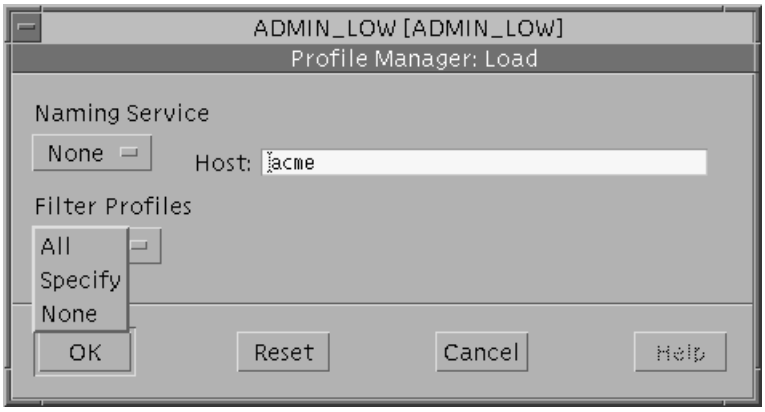


Figure 8–4 Profile Manager: Load, Profile Filter Choicesprofmgr.load.prof.filters.menu.rs.

Note - The Profile Manager initially displays in *action mode,* with lists of excluded and included actions. (Figure 8–6 shows one example of the Profile Manager in action mode.) You can use the Profile Manager View menu to switch among the modes for actions, commands, and authorizations, as described " To Specify Commands in the Profile Manager" on page 241.

Decision To Make

- **Decide whether you are adding a new profile and go to the appropriate section, either "When Adding a New Profile" on page 205" or "When Modifying an Existing Profile " on page 206.**

When Adding a New Profile

If you are adding a new profile, you can bring up the Profile Manager either of these two ways:

- Empty (as described under "Launching an Empty Profile Manager " on page 206)
- Loaded with an existing profile that you may then rename and modify (as described under "Launching the Profile Manager Loaded with an Existing Profile" on page 208).

Decision To Make Before Starting

- **Decide whether to copy and modify an existing profile or start with an empty Profile Manager.**

When Modifying an Existing Profile

To modify an existing execution profile, you launch the Profile Manager loaded with the profile (as described under “Launching the Profile Manager Loaded with an Existing Profile” on page 208) and you then make modifications, keeping the same name.

Launching an Empty Profile Manager

Choosing None from the Filter Profiles menu (as shown below) brings up an empty Profile Manager (as shown in the following figure).

Figure 8–5 Choosing None from the Profile Manager: Load, Profile Filter Menuprofmgr.load.filter.none.rs.

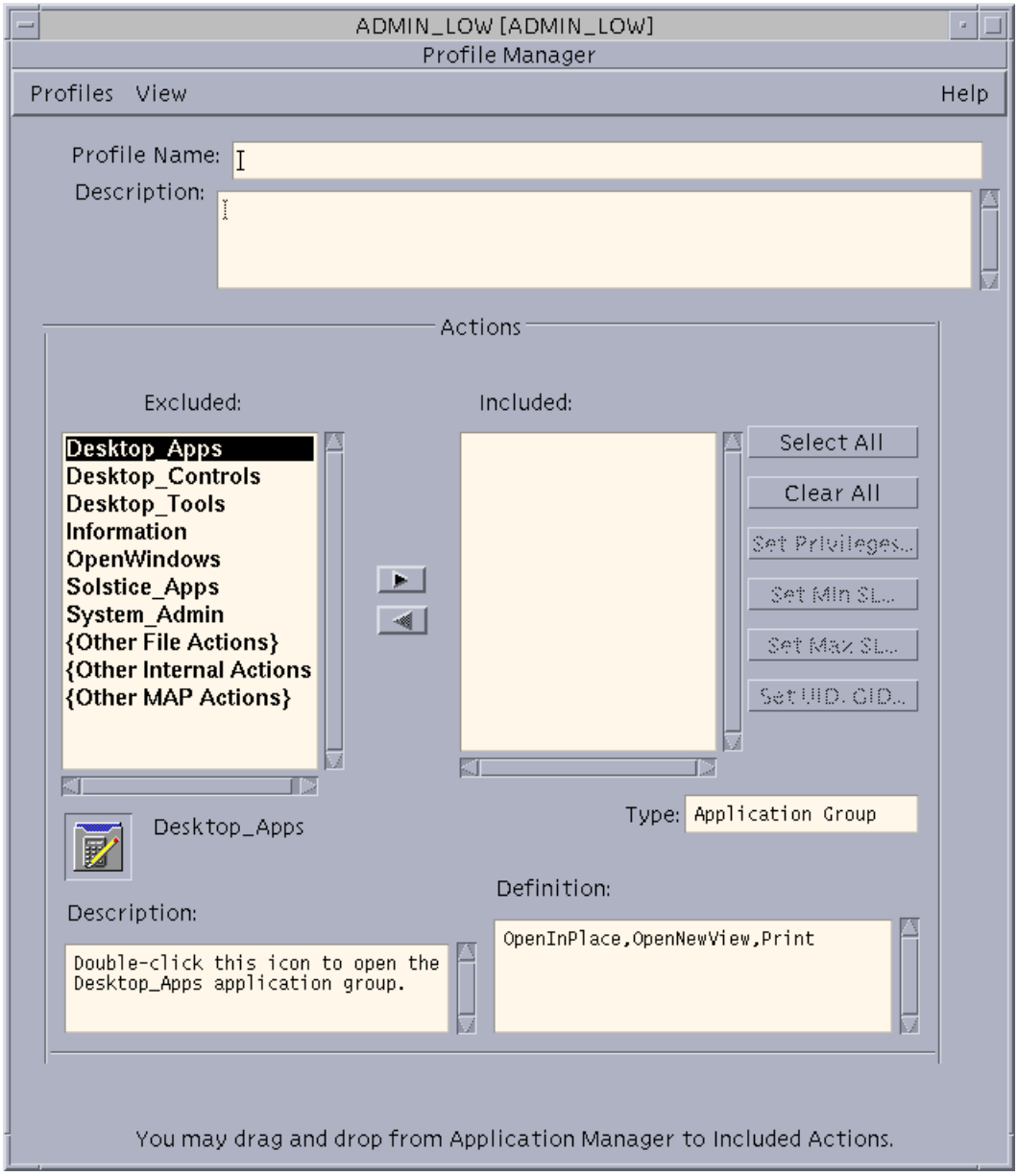


Figure 8–6 Empty Profile Manager in Action Mode profmgr.open.none.rs??

Launching the Profile Manager Loaded with an Existing Profile

Choosing either All or Specify from the Filter Profiles menu brings up a list from which you choose the name of a profile. When you click Load after selecting the name, the Profile Manager comes up with an existing profile loaded.

- If you want to select an existing profile name from the names of all existing profiles, see Step 1 on page 208.
- If you want to use a regular expression to search for the name of an existing profile, see Step 1 on page 211.Step 1 on page 212 on Step 1 on page 212.
- If you want to specify the name of an existing profile, see Step 1 on page 211.Step 1 on page 211 on Step 1 on page 211.

▼ Selecting existing profile names

1. **Choosing All from the Filter Profiles menu (as shown in the following figure) allows you to pick the name of a profile from a list of all profiles (as shown in the following figure).**

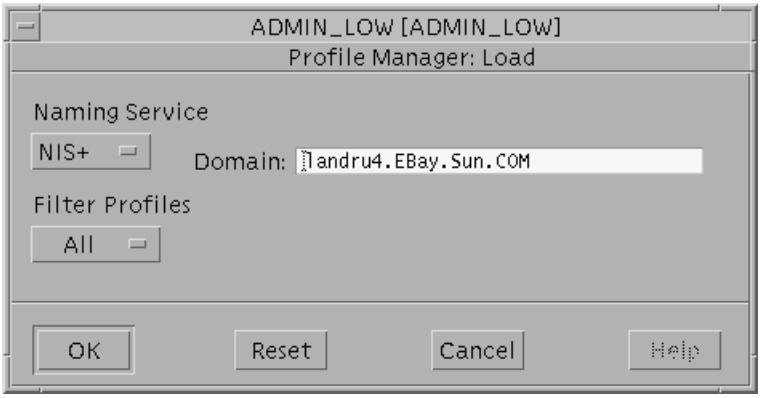


Figure 8–7 Choosing All from the Profile Manager: Load, Filter Profiles Menuprofmgr.load.filter.all.rs

Highlighting the name of the profile and clicking Load (as shown in Figure 8–8) opens the Profile Manager with the profile loaded (as shown in Figure 8–9).

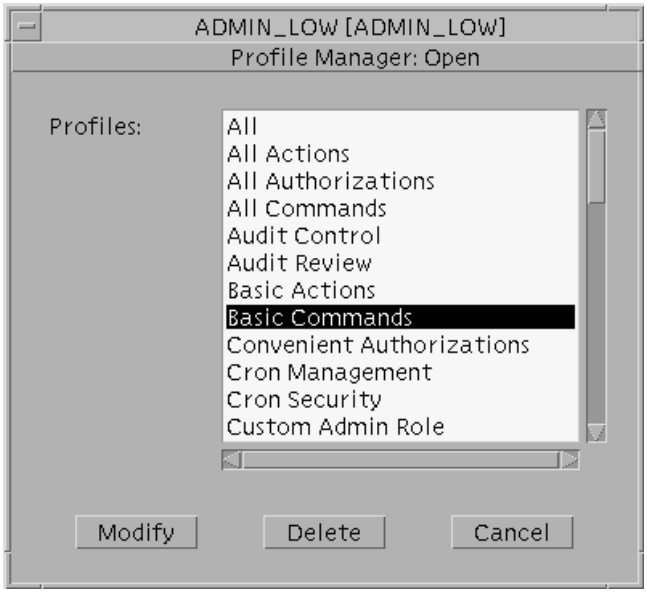


Figure 8–8 Profile Manager: Load, Highlighting a Profile Nameprofmgr.open.all.selected.rs

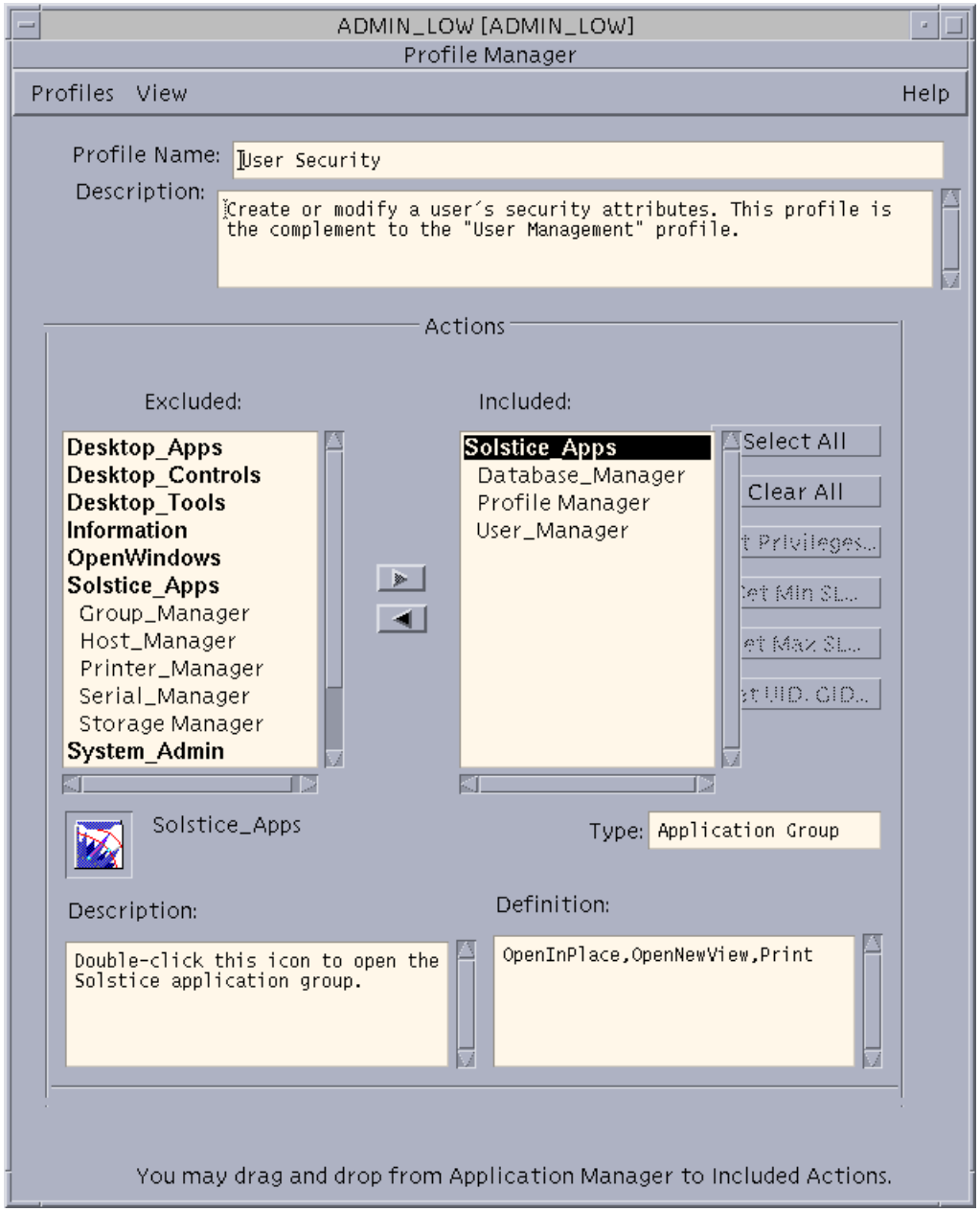


Figure 8–9 Profile Manager with a Profile Loadedprofmgr.user.security.selected.rs

▼ Specifying an existing profile name

1. **Choosing Specify from the Filter Profiles menu allows you to specify a profile name in a text field in one of the following two methods.**
   a. **Entering a regular expression creates a list of profiles to select among.**

Figure 8–10 Specifying a Profile to be Loaded in the Profile Manager by Using a Regular Expression profmgr.load.filter.specify.rs

The example in the figure above shows P* entered to locate any profile beginning with P. Figure 8–11 shows the result: the Privileged Shells profile displays in the Profile Manager: Open dialog. Highlighting the name of the profile and clicking Load opens the Profile Manager with the Privileged Shell profile loaded (as shown in Figure 8–12).

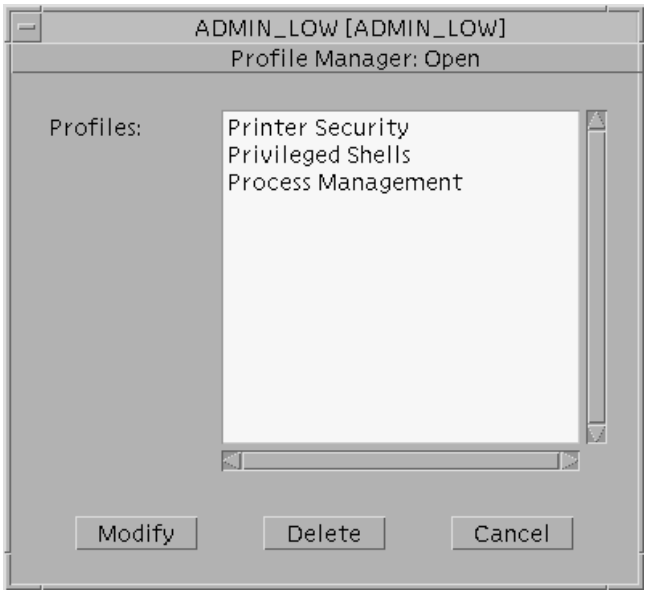


Figure 8–11 Privileged Shells Profile Listed in the Profile Manager: Open Dialog when P* is Specified in the Filter Profiles Text Field profmgr.open.specify.rs

b. Entering the name of a known profile causes the profile name to be displayed for your selection.

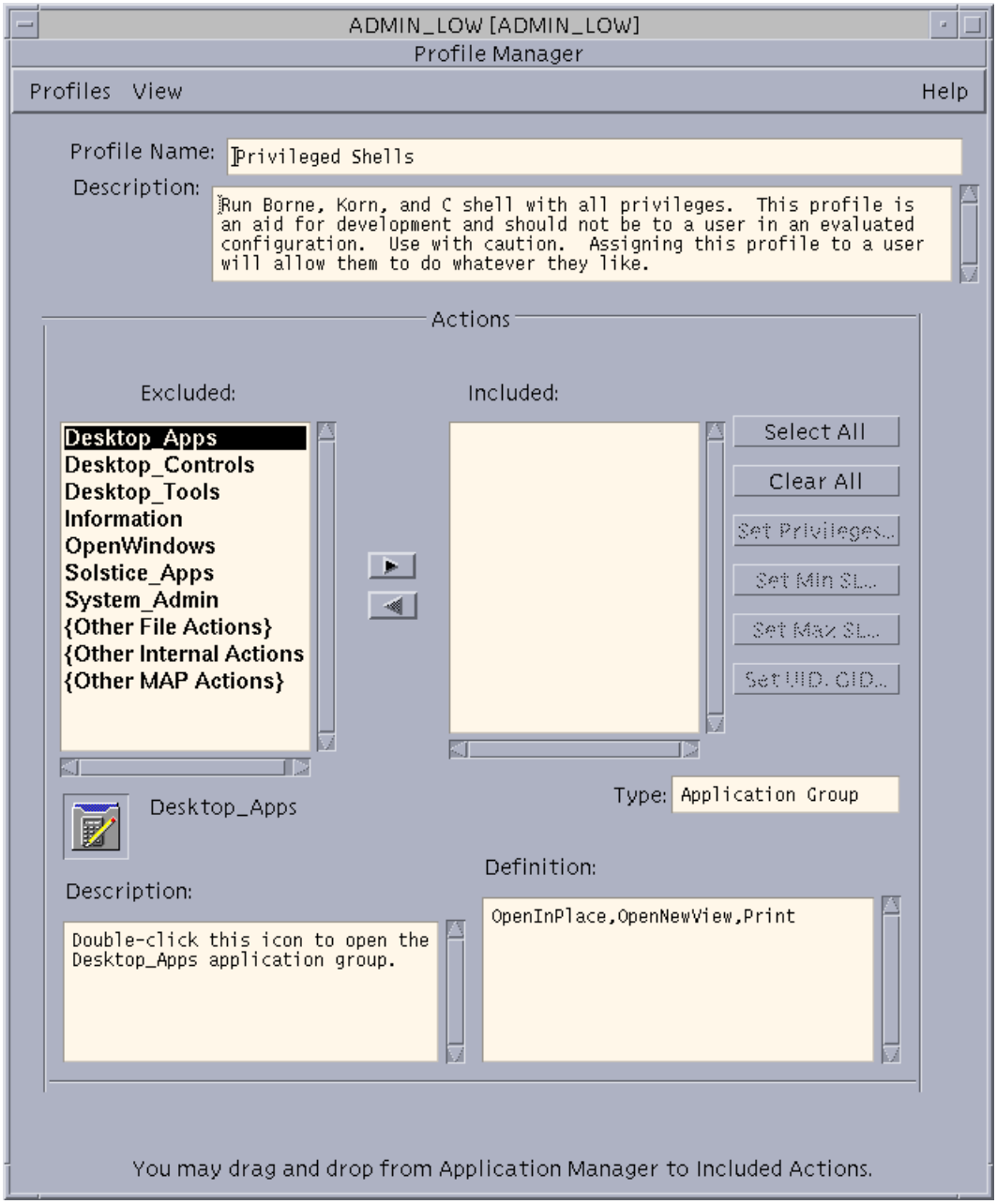


Figure 8–12 Profile Manager Loaded with the Privileged Shells Profileprofmgr.specified.profile.rs

Bringing Up a Blank Profile Definition, Loading an Existing Profile, or Saving Changes Within the Profile Manager

The following figure shows the options on the Profile menu.

Figure 8–13 The Profile Manager Profiles Menu for Opening, Saving, and Closing Profilesprofmgr.profiles.menu.rs

You can use the options in the Profiles menu to do the following:

- New Profile clears the existing profile description from the Profile Manager.
- Open Profile allows you to select another profile to load from the list of all profiles. (A Profile Manager: Open dialog displays with a list of all profiles for you to choose from, as shown in Figure 8–14.)
- Save Profile saves any changes you have made to the profile.
- Load loads a new set of execution profiles.
- Close closes the Profile Manager without prompting you or saving any unsaved changes.

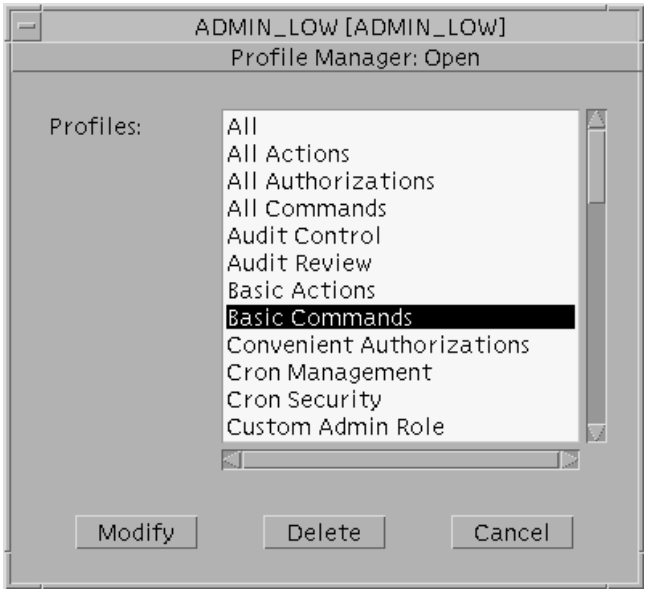


Figure 8–14 Profile Manager: Open, Highlighting a Profile Nameprofmgr.open.all.selected.rs

Entering or Changing the Profile Name or Description

If you brought up the Profile Manager without a profile loaded, you may enter a new profile name and description in the empty text entry fields. If you enter a name and description and use Profiles->Save, a new profile is created with the specified name.

When the Profile Manager is loaded with an execution profile, you may modify the existing name and description as shown in Figure 8–15 . If you modify an existing profile's name and use Profiles->Save, a new profile is created with the modified name.

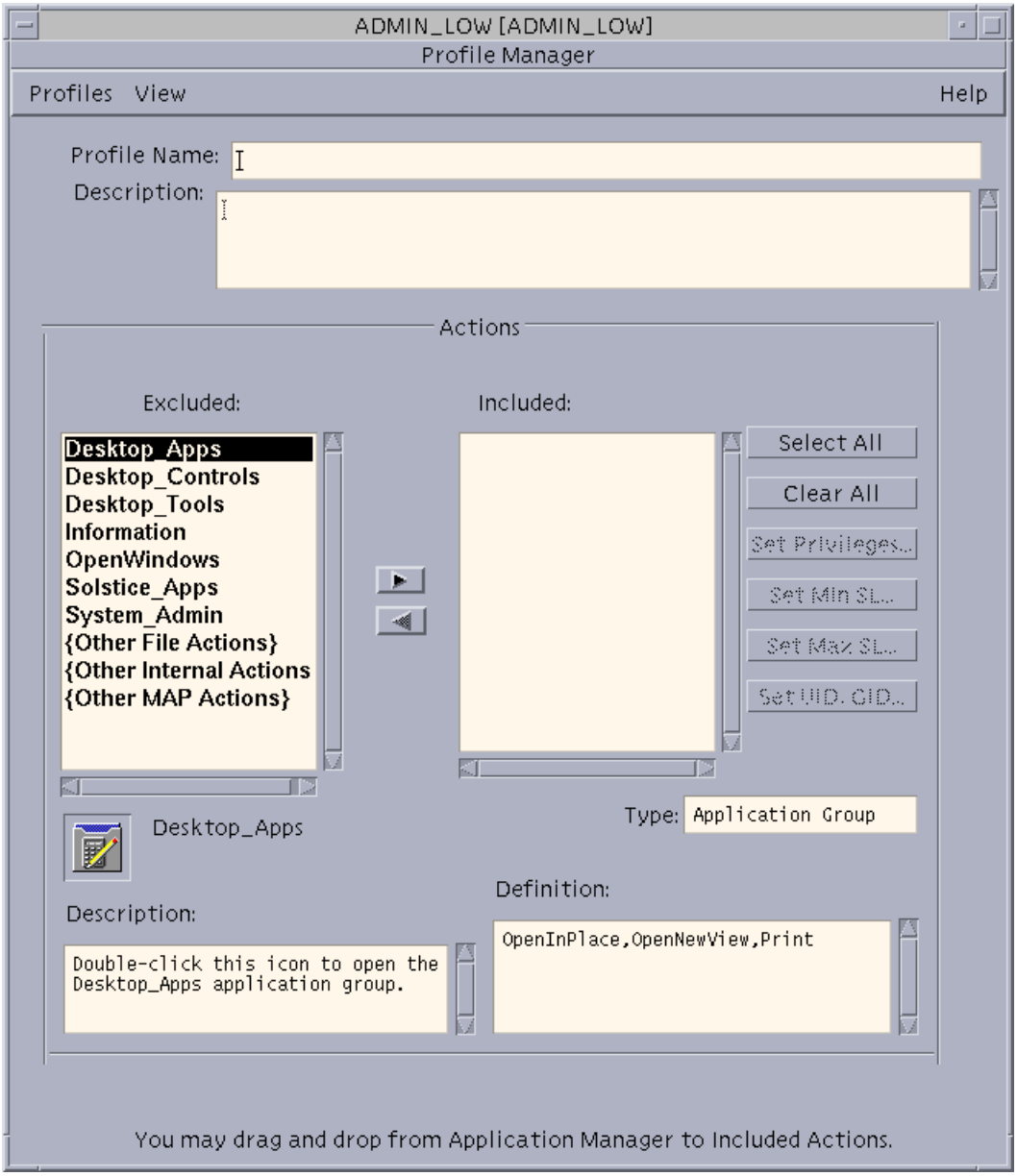


Figure 8–15 The Profile Name and Description Fields in the Profile Managerprofmgr.auth.mode.rs

Switching Among Actions, Commands, and Authorizations Modes

The Actions mode is the first one that comes up after you click OK on the Profile Manager: Load dialog (as shown previously in Figure 8–12). The following figure shows the View menu for switching among the actions, commands, and authorizations modes.

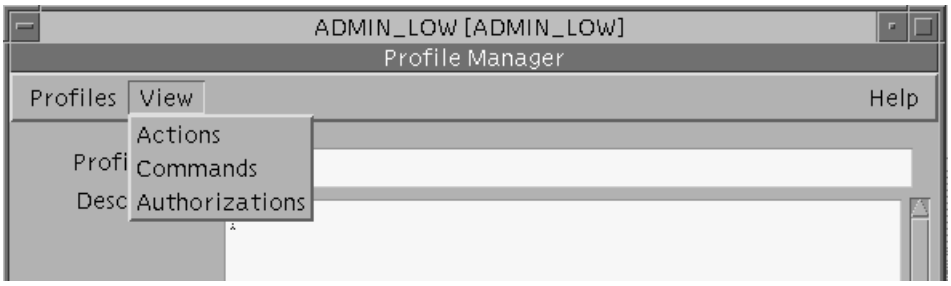


Figure 8–16 The Profile Manager View Menu for Switching Between Actions, Commands, and Authorizations profmgr.view.menu.rs

See “Working in Command Mode” on page 224, “Working in Authorizations Mode” on page 227, and “Working in Actions Mode” on page 229.

Working with the Excluded and Included Lists

The following figure shows an example of the Profile Manager Excluded and Included Lists, which are used for actions, command, authorizations, and privileges. The example shows actions, but the methods for working with the items in these lists apply to any of the Profile Manager dialogs that show excluded and included lists.

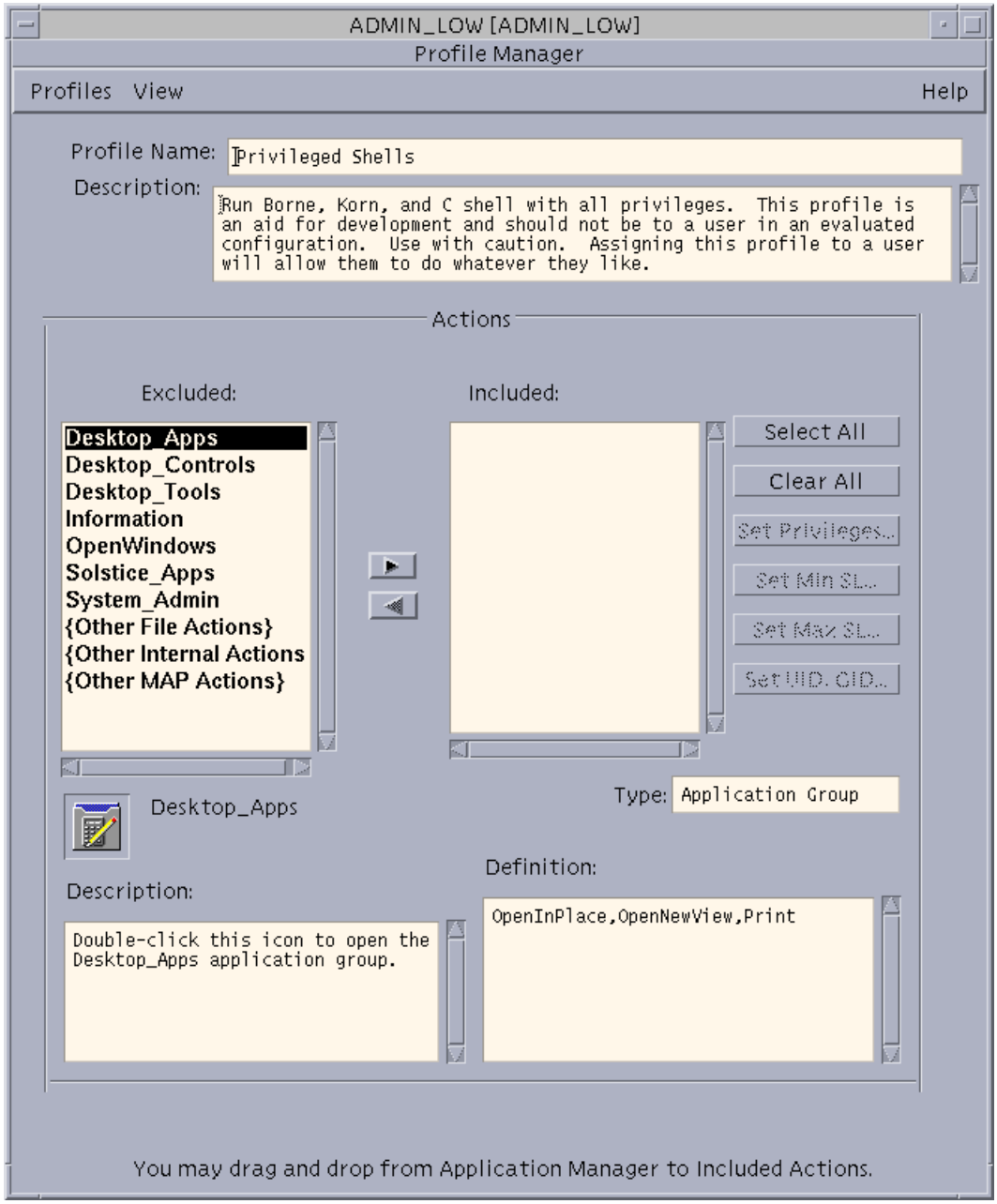


Figure 8–17 Profile Manager Loaded with the Privileged Shells Profileprofmgr.specified.profile.rs

Moving Items Between Lists

You can move an item (whether it is an authorization, a privilege, a command, a complete directory, an individual action, or a group of actions in an application group) by highlighting the item's name and using the arrow button to move it to the desired list.

Dragging and Dropping Into the Included List

As noted at the bottom of the Profile Manager, you may drag and drop items that match the current mode from the File Manager into the included list. For example, in commands mode you could drag the icon in for the `mount` command's executable file from the `/etc` folder to the included list.

Moving and Clearing Many List Items with the Select All and Clear All Buttons

All of the Profile Manager modes and the Set Privilege Dialog Box have the Select All and Clear All buttons.

Select All

Select All moves all of the items in the excluded list to the included list

Clear All

Clear All moves any items in the included list to the excluded list

Working with Common Features of the Commands and Actions Modes

This section describes using common features of the Profile Manager's commands and actions modes in the following subsections

Expanding and Contracting Application Group and Directory Listings in the Command and Actions Modes

Each heading in the default excluded list stands for a group of actions (in the Actions Mode) or commands (in the Commands Mode). By highlighting the heading and double-clicking on it or on its icon (if one exists), you can view all the individual items (actions, or commands) it contains. For example, when you double-click on the highlighted Desktop Apps action heading it expands to list all the actions in that grouping, as shown in the following figure.

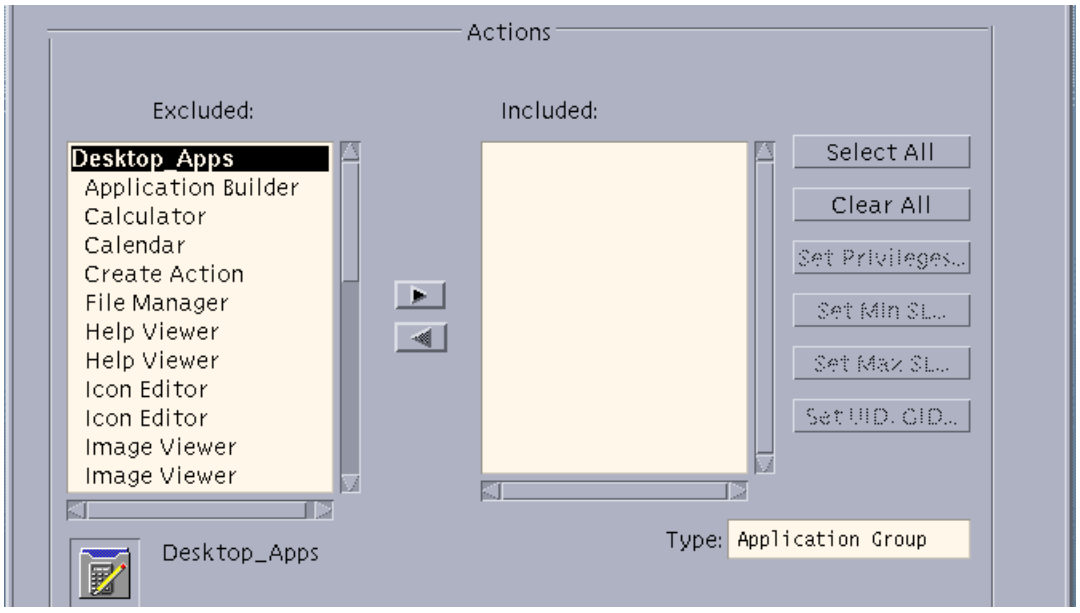


Figure 8–18 Expanding a Grouping Name to List All of its Contentsprofmgr.command.mode.rs

By double-clicking on an expanded command heading, you can contract the list back to its heading or directory name. Items in an expanded list can be moved between the excluded and included list one by one or the group can be moved by moving the heading or directory name.

Using the Buttons to Set Security Attributes on Commands and Actions

The buttons shown in the following figure only display when an item in the included list is highlighted and only when security attributes may be set for that item.

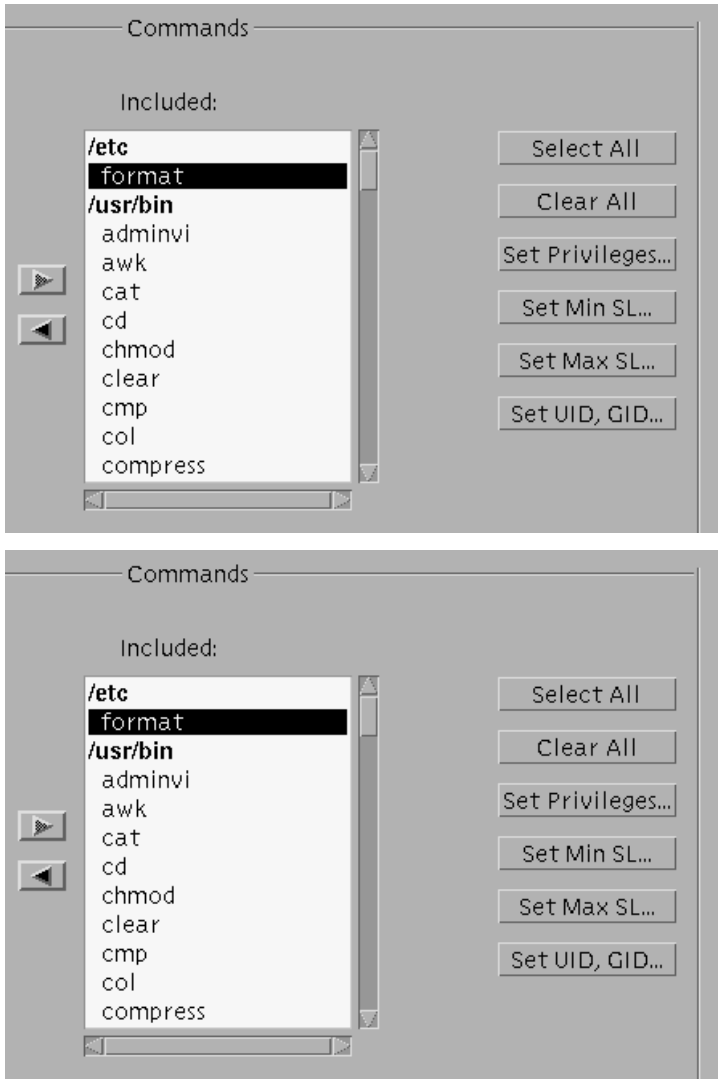


Figure 8–19 Buttons for Setting Privileges, Label Range, UID and GIDprofmgr.cmds.move.cmd.rs

Figure 8–20 shows the format command highlighted, and because privileges, an SL range, UID and GID may be set on commands, the buttons are visible and usable. In contrast, because Tool Talk messages cannot have security attributes, when an action that a Tool Talk message (TT_MSG appears in the description field) is highlighted, the buttons are grayed.

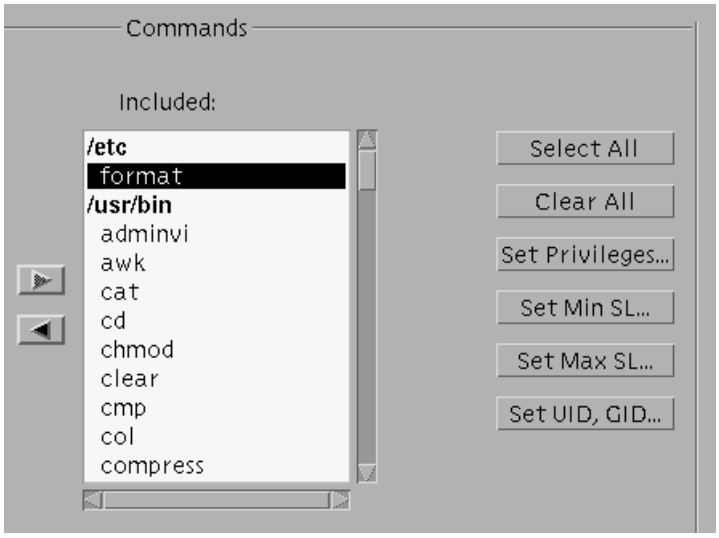


Figure 8–20 Buttons for Setting Privileges, Label Range, UID and GIDprofmgr.cmds.move.cmd.rs

Setting Privileges on Commands and Actions

After a command or action is highlighted in the Profile Manager, clicking the Set Privileges button brings up the Set Privileges dialog, as shown in Figure 8–21. A number of privileges are listed in the excluded list. Whether the included list is empty or not depends on whether the command has previously been assigned any privileges.

Description

The Description field describes what the privilege allows the process executing the command to do to bypass security policy.

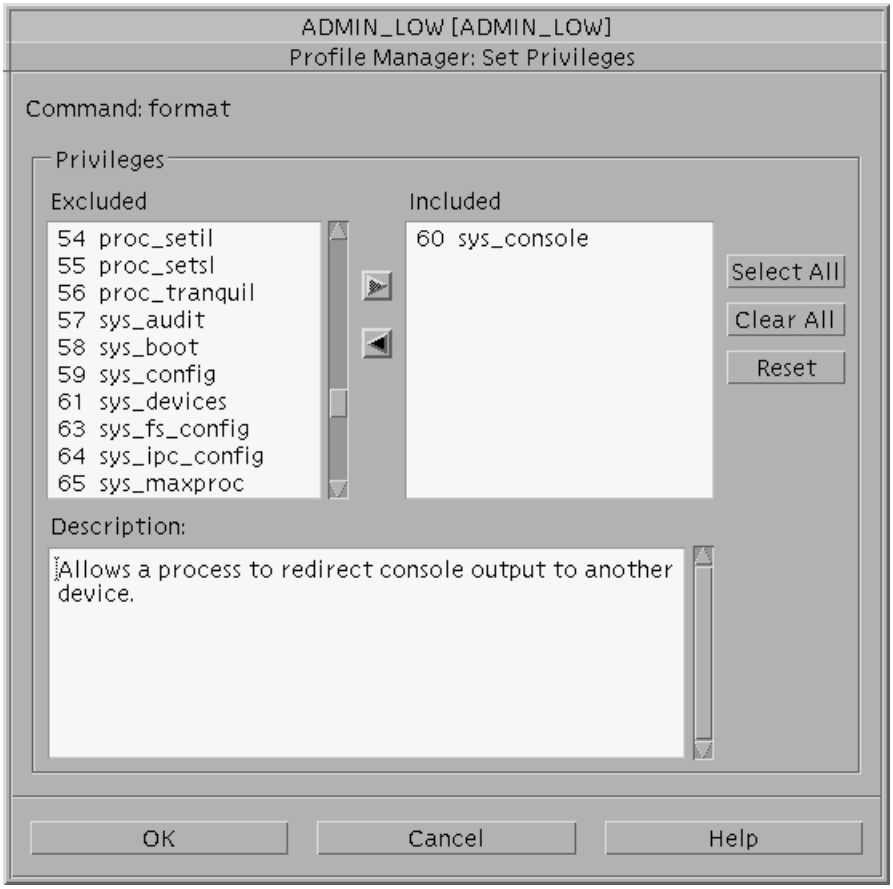


Figure 8–21 Profile Manager: Set Privileges Dialog profmgr.cmds.move.cmd.rsBox

Setting a Label Range for a Command or Action

After a command or action is highlighted in the Profile Manager, you set a label range by setting a minimum and maximum sensitivity label. Clicking the Set Minimum SL or Set Maximum SL button brings up a label builder. (The Set Minimum SL dialog is shown in Figure 8–22.) Using these dialogs to set the minimum and maximum SL of a command or action is the same as using any other Trusted Solaris label builder.

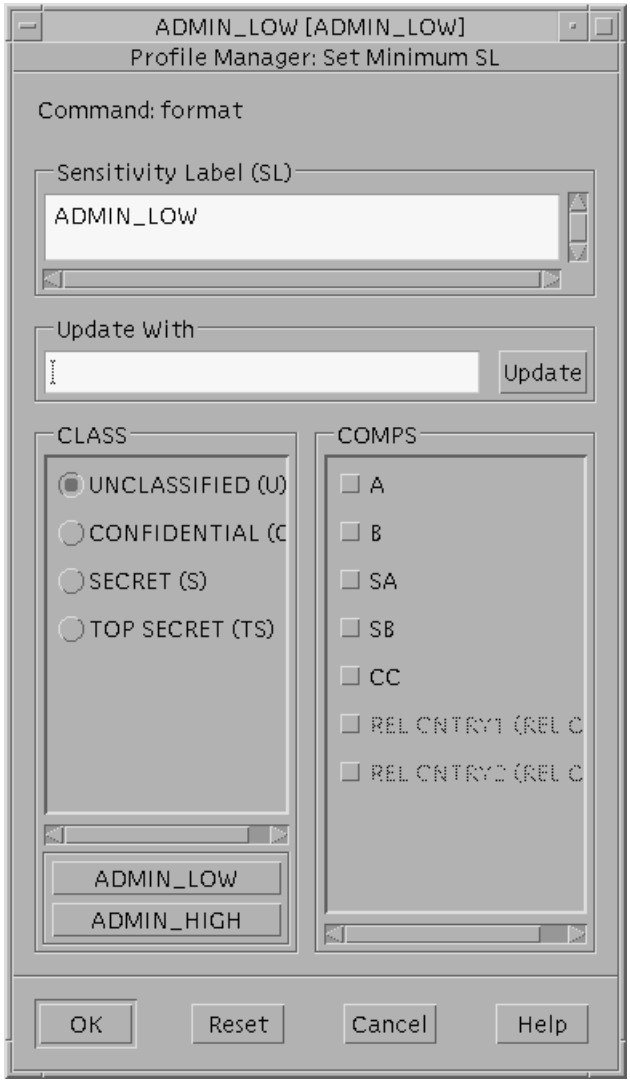


Figure 8–22 Profile Manager: Set Minimum SL Dialog profmgr.cmds.set.minSL.rs

Working in Command Mode

This section explains how to use features of the command mode that are not on any of the dialogs in the authorizations and actions modes. It does not duplicate information provided in the following sections on topics that apply to working in command mode:

- "Entering or Changing the Profile Name or Description" on page 215

- “Working with the Excluded and Included Lists” on page 217
- “Working with Common Features of the Commands and Actions Modes” on page 219

Selecting Commands from the Profile Manager View menu brings up the command mode, as shown in Figure 8–23. A number of directories are listed in the excluded list. Whether the included list is empty or not depends on whether the profile has been assigned any commands.

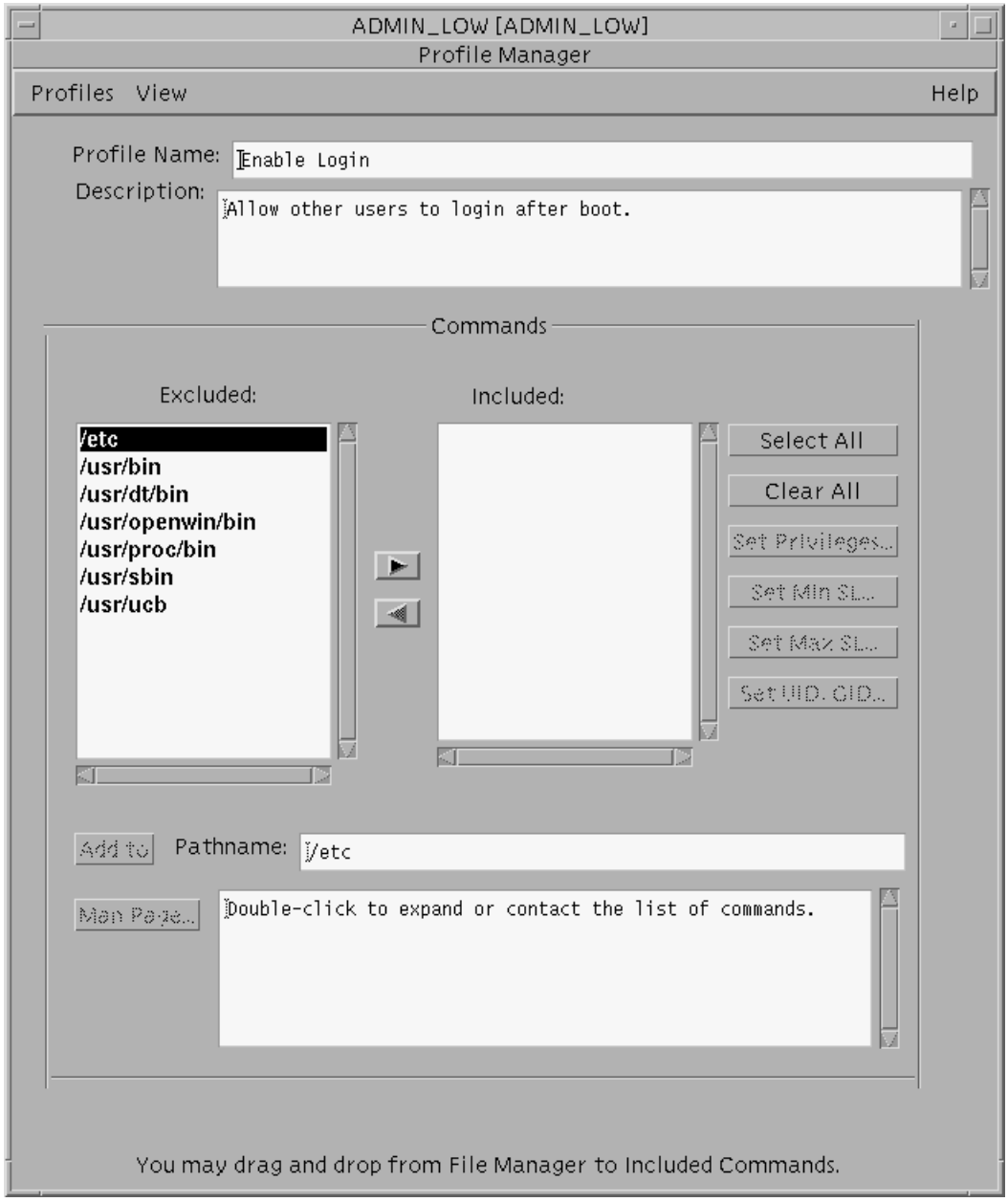


Figure 8–23 The Profile Manager Command Mode profmgr.command.mode.rs

Loading a New DirectoryYou may add another directory to the excluded list by typing the pathname of the directory into the text entry field next to the Load button, and then clicking the load button. In the following figure, the `/etc` directory has been typed into the Pathname: field next to the Load button and loaded into the excluded list.

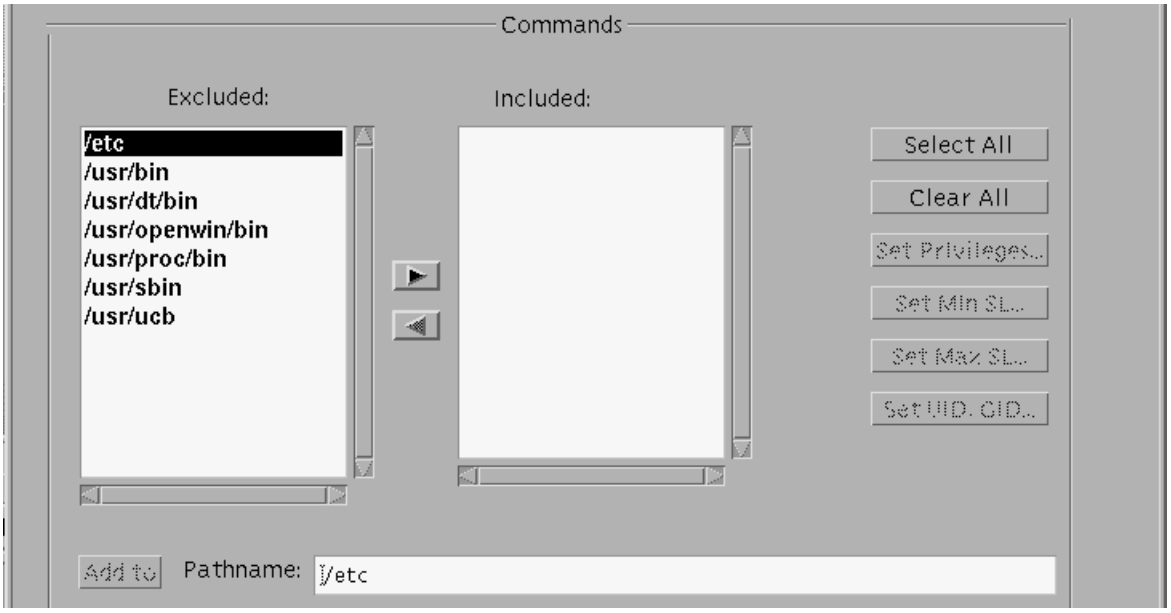


Figure 8–24 Entering the Pathname of the `/etc` Directory to Choose from its Commandsprofmgr.cmds.load.dir.rs

Viewing a Command's Man Page

When a command is highlighted in the included list, you can view its man page starting with the DESCRIPTION section by clicking the Man Page button.

Working in Authorizations Mode

The authorizations mode does not have any features that are not on the dialog boxes for the commands and actions modes. Working in authorizations mode is described in the previous sections:

- "Entering or Changing the Profile Name or Description" on page 215
- "Working with the Excluded and Included Lists" on page 217

- "Working with Common Features of the Commands and Actions Modes" on page 219

Selecting Authorizations from the Profile Manager View menu brings up the authorizations mode, as shown in Figure 8–25. A number of authorizations are listed in the excluded list. Whether the included list is empty or not depends on whether the profile has been assigned any authorizations.

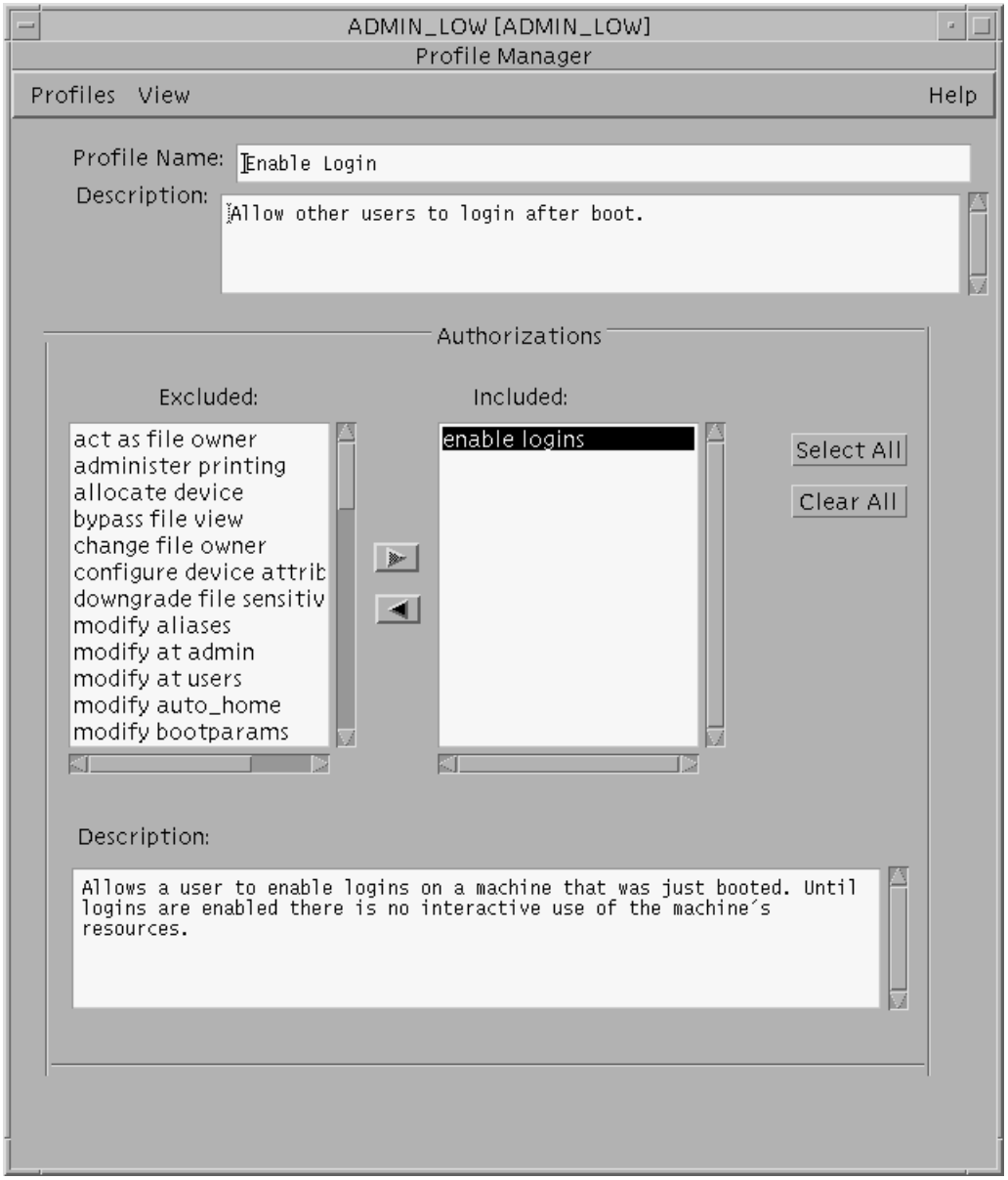


Figure 8–25 Profile Manager in Authorization Mode profmgr.auth.mode.rs

Working in Actions Mode

This section explains how to use features of the actions mode that are not on any of the dialogs in the commands and authorizations modes. It does not duplicate

information provided in the following sections on these topics that apply to working in actions mode:

- “Entering or Changing the Profile Name or Description” on page 215
- “Working with the Excluded and Included Lists” on page 217
- “Working with Common Features of the Commands and Actions Modes” on page 219

Selecting Actions from the Profile Manager View menu brings up the actions mode, as shown in Figure 8–27. A number of actions are listed in the excluded list. Whether the included list is empty or not depends on whether the profile has been assigned any actions.

Icon

The icon shown in Figure 8–26 is the icon assigned to the highlighted action or application group. As stated in the description section at the bottom of Figure 8–27, you can double-click the icon of an applications group to list all of the items contained in the group.

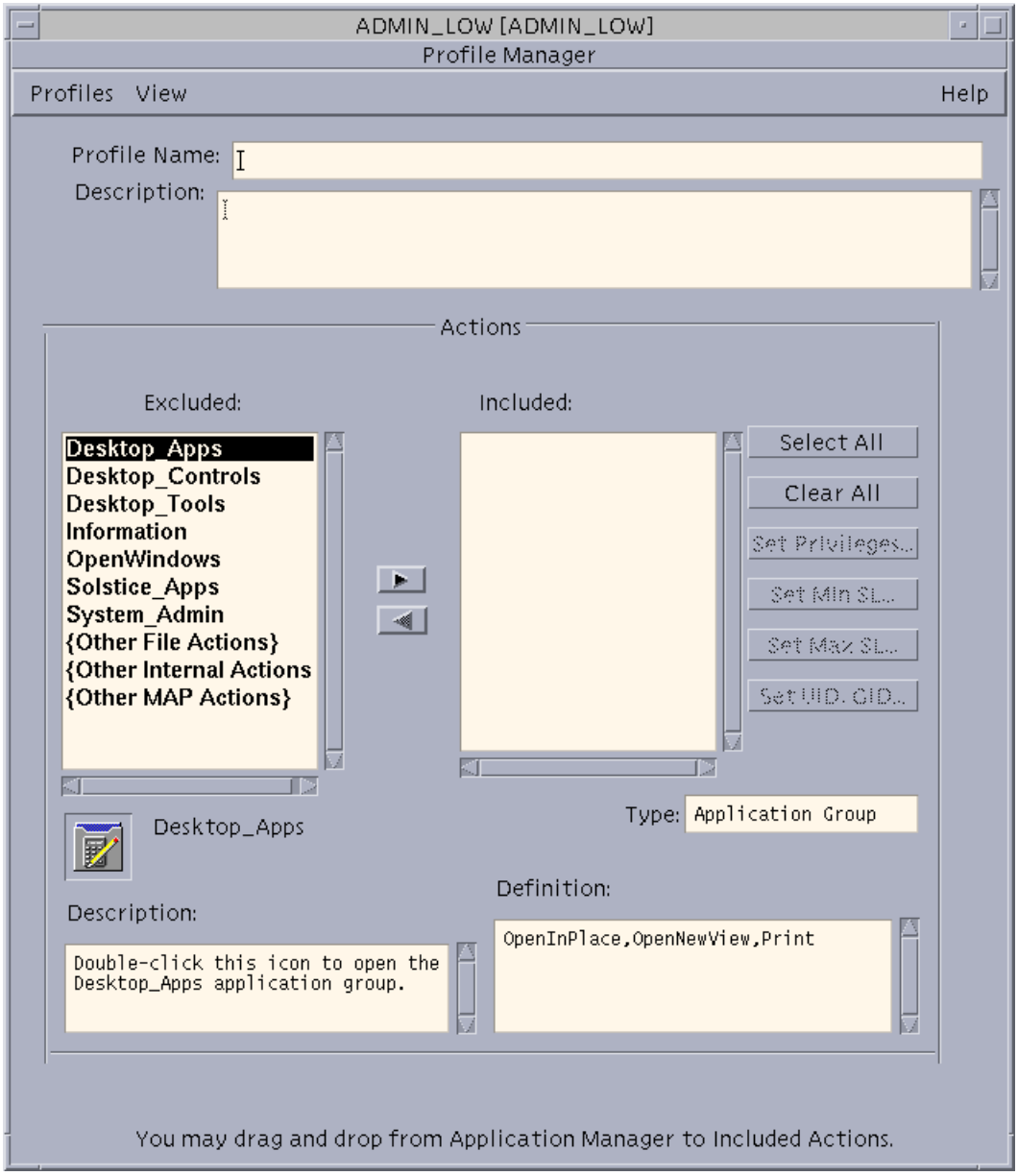


Figure 8–26 Icon and Type in Action Mode

Type

The Type field indicates the type of item highlighted. An application group is a heading that includes a number of actions, which may be expanded to list all its included actions. When an action is highlighted, the possible types are:

COMMAND TT_MSGprofmgr.open.none.rs

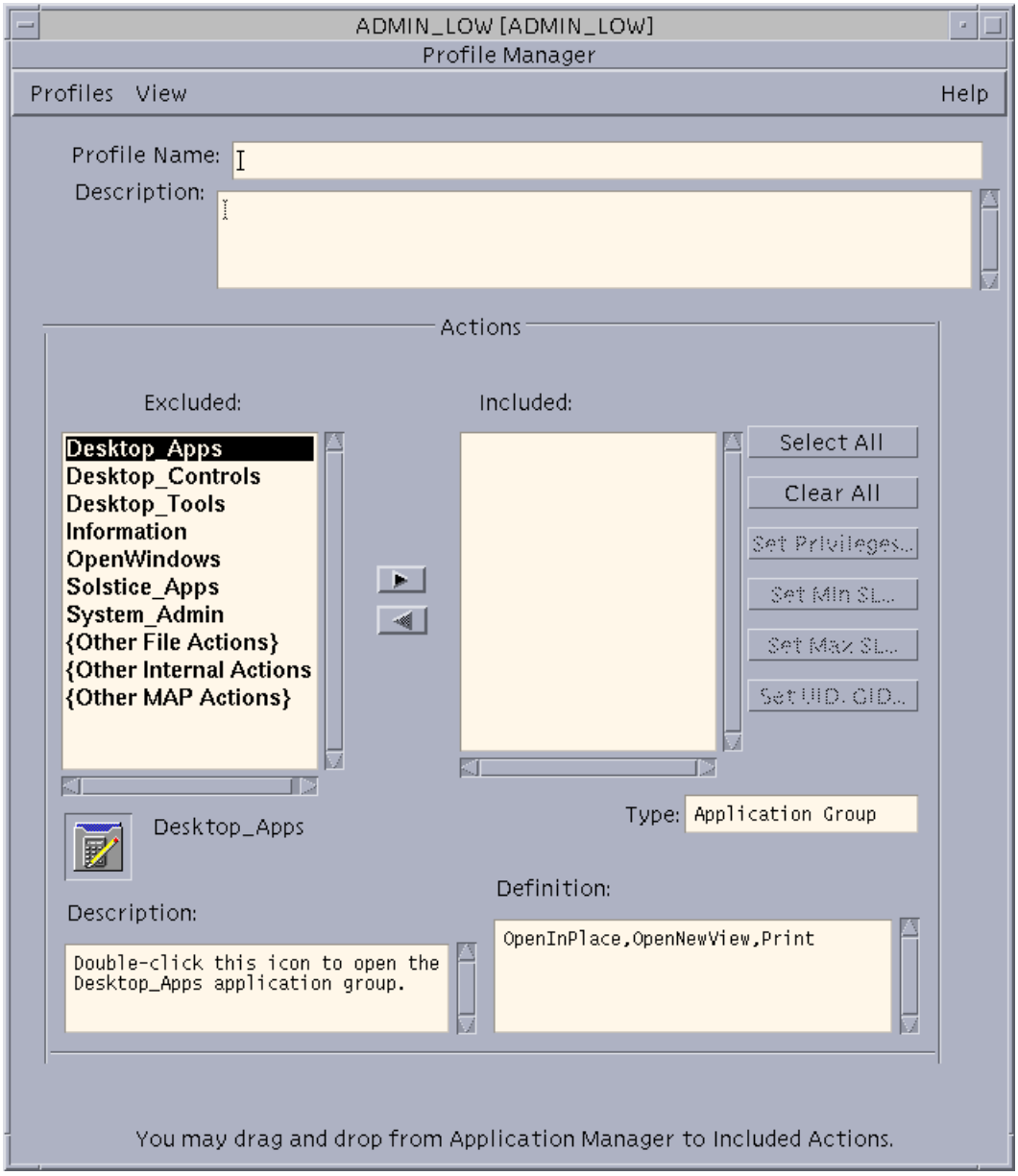


Figure 8–27 Profile Manager in Action Mode profmgr.open.none.rs

▼ To Access the Profile Manager

1. **Assume the security administrator role.**

See “To Login and Assume an Administrative Role” on page 15, if needed.

1. **In a workspace at ADMIN_LOW, bring up the Profile Manager by clicking on the Profile Manager icon in the Application Manager.**

   The following figure shows the Profile Manager highlighted in the Solstice_Apps folder.

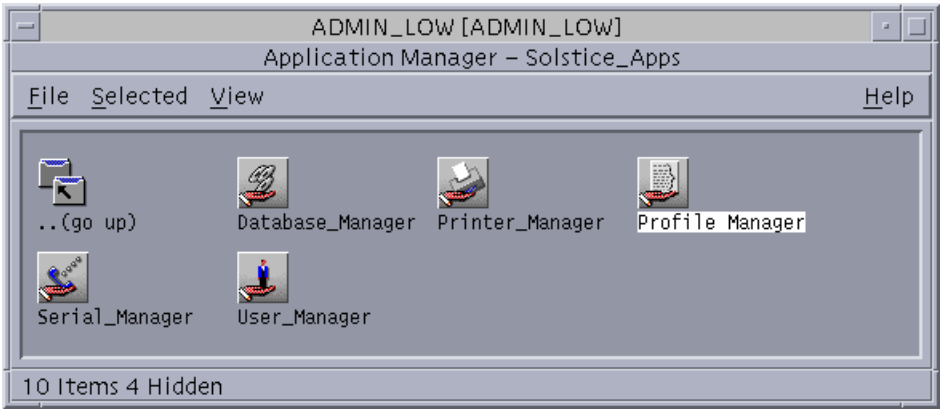


Figure 8–28 The Profile Manager Icon Highlighted in the Solstice_Apps Folderclick.prof_mgr.rs

The Profile Manager: Load window displays with the current domain name in the Domain field (see Figure 8–29).

▼ To Pick a Naming Service and Filter for Profiles

1. **Access the Profile Manager.**

   If needed, see “To Access the Profile Manager ” on page 233. The Profile Manager Load Dialog Box displays.

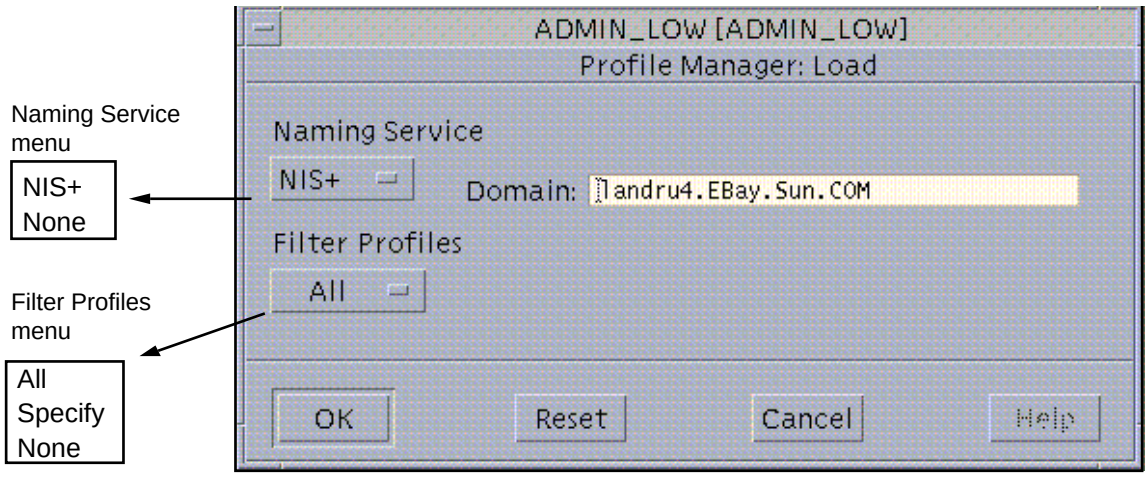


Figure 8–29 Profile Manager: Load Dialog Box profmgr.load.rs

2. Choose either NIS+ or None from the Naming Service menu.

a. If you choose NIS+, accept or change the name of the domain in the text field next to the NIS+ menu item (as shown in Figure 8–29).

b. If you choose None for the Naming Service, accept or change the name of the local host in the text field next to the None menu item (as shown in the following figure).

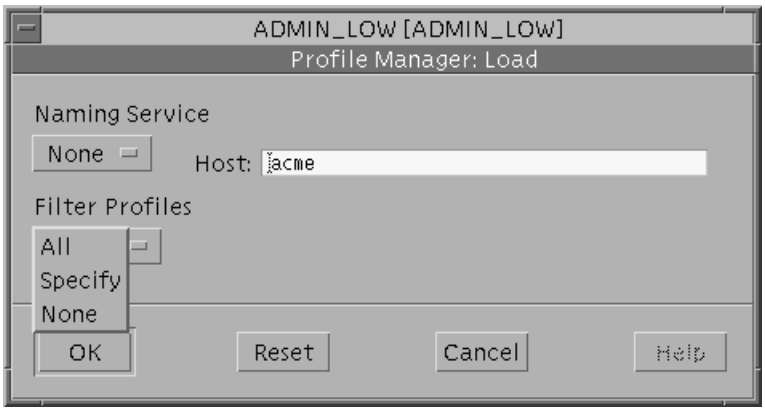


Figure 8–30 Choosing None from the Profile Manager: Load, Naming Service Menu profmgr.load.none.rs?

3. **Choose an item from the Filter Profiles menu to specify whether you want the Profile Manager to display with all profiles loaded, to display a specified profile, or to display with no profile loaded.**

   Choose All, Specify, or None, as shown in Figure 8–31.

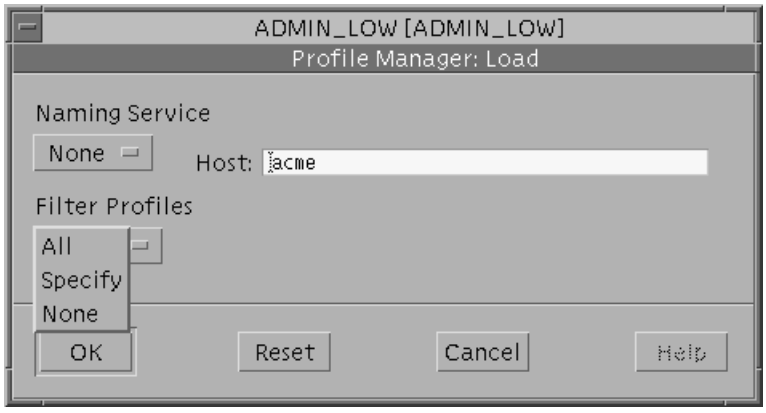


Figure 8–31 Profile Manager: Load, Profile Filter Choices
profmgr.load.filters.menu.rs

   a. **If you choose All, specify a profile from the list of all profiles.**

      i. **Click OK.**

         The Profiles: Load dialog box displays with all the existing execution profiles listed.

      ii. **Select a profile from the list.**

      iii. **Click Load.**

         The Profile Manager displays with the profile or profiles loaded.

   a. **If you choose Specify, specify the name of a profile to load.**

      When you choose Specify, a text entry field displays next to the menu item, as shown in the following figure.

Figure 8–32 Specifying Profile Names Using a Regular Expression on the Profile Manager: Load, Filter Profiles Menu profmgr.load.specify.rs

i. Enter the name of an existing execution profile or a regular expression in the text entry field.

The example in Figure 8–32 shows P* entered to locate any profile beginning with P.

ii. Click OK or press Return.

A list of one or more profiles displays in the Profile Manager: Open dialog (as shown in Figure 8–33.

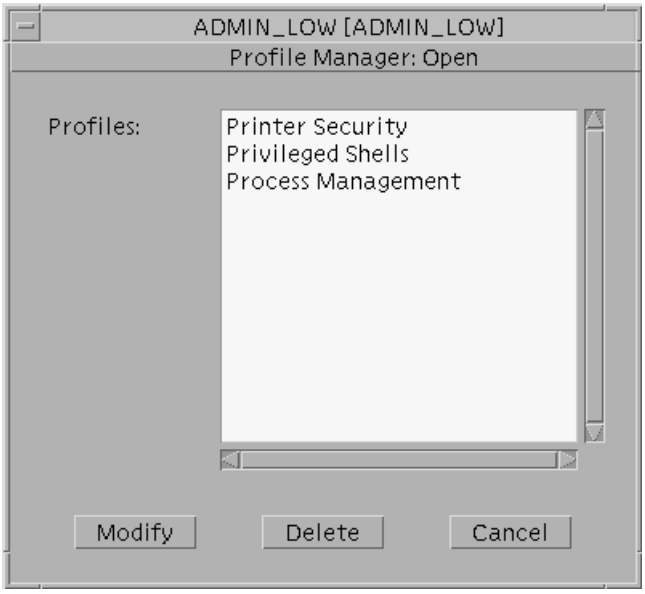


Figure 8–33 Profiles Displayed When P* is Specified profmgr.open.specify.rs

iii. Highlight the desired profile name.

iv. Click Load.

The Profile Manager displays with the profile or profiles loaded.

a. If you chose None, go to "To Enter the Name and Description for a New Profile" on page 241.

The Profile Manager displays the list of available actions with no profiles loaded, as shown in Figure 8–34.

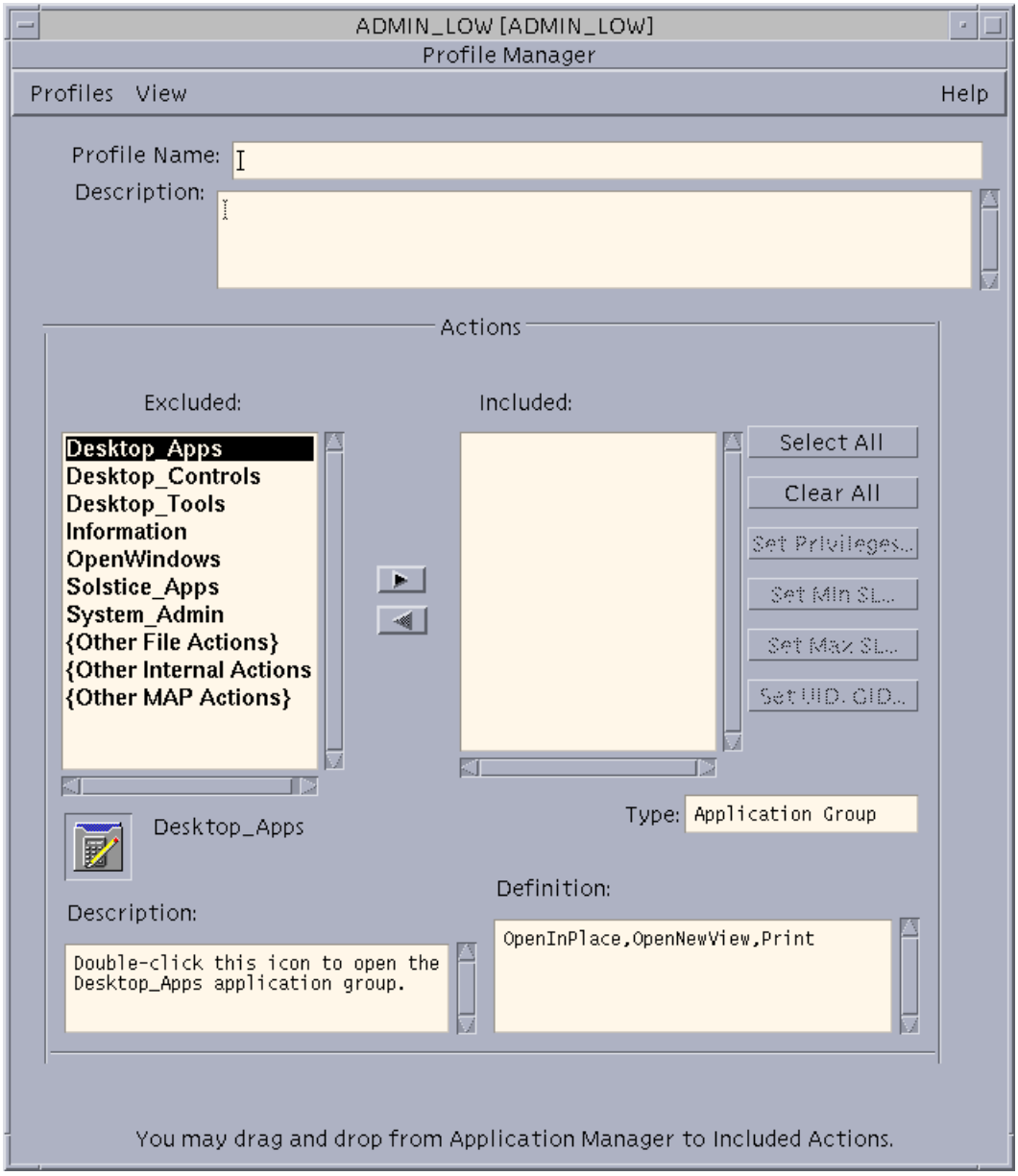


Figure 8–34 Empty Profile Manager in Action Mode profmgr.open.none.rs

Specifying a New Profile

Begin with:

- “To Access the Profile Manager ” on page 233

When creating a new execution profile, you can rename and modify an existing profile or start with all fields empty.

To specify an existing execution profile, on Step 3 on page 236 under “To Pick a Naming Service and Filter for Profiles” on page 234),” do Step 3 on page 241.Step 3 on page 236 or Step 3 on page 236.

To enter a new name and start with all fields empty, do Step 3 on page 236.Step 3 on page 238, and go to the following:

- “To Enter the Name and Description for a New Profile” on page 241

Define the new profiles actions, commands, and authorizations, as desired, as described in:

- “ To Specify Commands in the Profile Manager” on page 241
- “To Specify Actions in an Execution Profile” on page 242
- “To Specify Authorizations in an Execution Profile” on page 244

Modifying an Existing Profile

When modifying an existing profile, start with:

- “To Access the Profile Manager ” on page 233

On Step 3 on page 236 under “To Pick a Naming Service and Filter for Profiles” on page 234,” do Step 3 on page 241.Step 3 on page 236 or Step 3 on page 241.Step 3 on page 236 to specify the name of an existing profile.

Redefine any of the profile’s sets of actions, commands, or authorizations as desired, as described in:

- “ To Specify Commands in the Profile Manager” on page 241
- “To Specify Actions in an Execution Profile” on page 242
- “To Specify Authorizations in an Execution Profile” on page 244

Execution Profile Procedures

▼ To Enter the Name and Description for a New Profile

1. **If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager " on page 233 and "To Pick a Naming Service and Filter for Profiles" on page 234.**

2. **Enter the name in the Profile Name: field.**

3. **Enter a description of the profile in the Description: field.**

4. **Choose Save from the Profiles menu.**

▼ To Specify Commands in the Profile Manager

1. **If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager " on page 233 and "To Pick a Naming Service and Filter for Profiles" on page 234.**

2. **Choose Commands from the View menu.**

3. **Put the name of the command you want to specify into the included list.**

   If the directory where the command resides is not in the default list, do Step 3 on page 241 and Step 3 on page 241 or do Step 3 on page 241.

   a. **Enter the name of the directory where the command resides into the Pathname field, click the Load button, and do Step 3 on page 241.**

   b. **Expand the name of the directory where the command resides, highlight the name of the command, and move it to the included list.**

   c. **Drag the icon for the command from its folder into the included list.**

4. **To specify security attributes, highlight the name of the command in the included list, and click on the appropriate buttons.**

   Specify privileges in Step 5 on page 241.

5. **Specify privileges, if desired.**

a. **Click the Set Privileges button.**

b. **On the Set Privileges dialog box, move the desired privileges to the included list.**

c. **Click OK.**

6. **Specify a label range, if desired.**
   a. **Click the Set Minimum SL button.**

   b. **On the Set Minimum SL dialog box, specify a minimum SL.**

   c. **Click OK.**

   d. **Click the Set Maximum SL button.**

   e. **On the Set Maximum SL dialog box, specify a maximum SL.**

   f. **Click OK.**

7. **Specify effective UID/GID, if desired.**
   a. **Click the Set UID/GID button.**

   b. **On the Set UID/GID dialog box, move the desired user name to the included list.**

   c. **Move the desired group name to the included list.**

   d. **Click OK.**

8. **On the Profile Manager, choose Save from the Profiles menu.**

9. **If you are finished, choose Close from the Profiles menu.**

10. **If you are not finished, choose Actions or Authorizations from the View menu.**

▼ To Specify Actions in an Execution Profile

1. **If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager " on page 233 and "To Pick a Naming Service and Filter for Profiles" on page 234.**

2. **Choose Actions from the View menu.**

3. **Put the name of the action you want to specify into the included list.**

4. **If the action is not in an application group in the default list, drag the icon for the action from its folder into the included list.**
   a. **Highlight the name of the action in the included list.**

5. **Specify security attributes for an action of the appropriate type, if desired.**
   If the type is TT_MSG, the buttons for specifying security attributes are grayed and you cannot specify security attributes for the action. If the type is COMMAND, click on the appropriate buttons to specify security attributes, as described in Step 6 on page 242, Step 7 on page 243, and Step 8 on page 242.

6. **Specify privileges, if desired.**
   a. **Click the Set Privileges button.**
   b. **On the Set Privileges dialog box, move the desired privileges to the included list.**
   c. **Click OK.**

7. **Specify a label range, if desired.**
   a. **Click the Set Minimum SL button.**
   b. **On the Set Minimum SL dialog box, build or type in a minimum SL.**
   c. **Click OK.**
   d. **Click the Set Maximum SL button.**
   e. **On the Set Maximum SL dialog box, build or type in a maximum SL.**
   f. **Click OK.**

8. **Specify effective UID/GID, if desired.**
   a. **Click the Set UID/GID button.**
   b. **Move the desired user name to the included list.**
   c. **Move the desired group name to the included list.**

d. Click OK.

9. **Choose Save from the Profiles menu.**

10. **If you are finished, choose Close from the Profiles menu.**

11. **If you are not finished, choose Commands or Authorizations from the View menu.**

▼ To Specify Authorizations in an Execution Profile

1. **If the Profile Manager is not up, bring it up as described in "To Access the Profile Manager " on page 233 and "To Pick a Naming Service and Filter for Profiles" on page 234.**

2. **Choose Authorizations from the View menu.**

3. **Put the name of the authorization you want to specify into the included list by highlighting its name and moving it to the included list.**

4. **Choose Save from the Profiles menu.**

5. **If you are finished, choose Close from the Profiles menu.**

6. **If you are not finished, choose Commands or Actions from the View menu.**

▼ To Customize an Administrative Role

1. **To modify the custom role profile for admin, root, or oper, assume the security administrator role. To modify the custom secadmin role profile, assume the root role.**

   See "To Login and Assume an Administrative Role" on page 15, if needed.

Note - The security administrator role can modify its own custom root profile only if the secadmin role has been given the *permit self modification* authorization. Only the root role can assign the needed authorization or modify any other aspect of the secadmin role.

---

2. **Go to an ADMIN_LOW workspace and open the Profile Manager.**

   See “To Access the Profile Manager ” on page 233, if needed.

3. **On the Profile Manager: Load dialog box, choose a naming service and choose the custom role profile to change.**

   See “To Pick a Naming Service and Filter for Profiles” on page 234, if needed.

   a. **To make a change to the role’s profile on all NIS+ clients, choose NIS+ from the naming service menu. To make a change to a role profile on the local host, select None as the naming service.**

   b. **Specify the name of the custom role profile to modify, and click the Modify button.**

4. **Specify one or more commands, if desired, with any Trusted Solaris security attributes.**

   See “ To Specify Commands in the Profile Manager” on page 241, if needed.

5. **Specify one or more actions, if desired, with any Trusted Solaris security attributes.**

   See “To Specify Actions in an Execution Profile” on page 242, if needed.

6. **Specify one or more authorizations, if desired.**

   “To Specify Authorizations in an Execution Profile” on page 244.

7. **Choose Save from the Profiles menu to save the profile.**

8. **If you chose None for the naming service, make sure that the administrator role uses the Name Service Switch action to change to the `tsolprof` entry to specify `files` before `nisplus`, if needed.**

   See “To Launch Administrative Actions” on page 29, if needed for how to access the Name Service Switch action. The tsolprof entry should look like the following example.

```
tsolprof: files nisplus
```

PART III Managing Hosts and Networks

Most hosts running Trusted Solaris are connected to a network. Installing a non-networked standalone host is covered in the *Trusted Solaris Installation and Configuration* manual. This part of the *Trusted Solaris Administrator's Procedures* manual contains the needed background and procedures for configuring hosts and networks. sharing files, and managing communications among hosts on local and remote networks.

Chapter 9 provides definitions of needed concepts.

Chapter 10 provides the following main topics:

Chapter 11 includes the following main topics:

Chapter 12 includes the following main topics:

Chapter 13 includes the following main topics:

Chapter 14 includes the following main topics:

Chapter 15 includes the following main topics:

Chapter 16 includes the following main topics:

CHAPTER 9

Trusted Solaris Concepts for Managing Hosts and Networks

This chapter provides the necessary concepts and background for administering hosts on a network in the following sections.

How to update the trusted network database and other related procedures are in Chapter 10. For an overview, see also Chapter 5, "Administering Trusted Networking" in the *Trusted Solaris Administration Overview*.

Review of Trusted Network Communications

The Trusted Solaris software supports network communications between Trusted Solaris 2.5.1 hosts and the following three types of hosts:

- Other Trusted Solaris 2.x hosts (Trusted Solaris 2.5 and 2.5.1)
- Hosts running operating systems or operating environments that do not recognize security attributes (such as Solaris and other UNIX systems)
- Hosts running other trusted operating systems that recognize some of the Trusted Solaris 2.x security attributes (such as Trusted Solaris 1.x hosts).

Network communications and services are managed by a number of different Trusted Solaris subsystems.

- Trusted NFS is used to manage mounted filesystems.

  Mounts among Trusted Solaris 2.x hosts and mounts among Trusted Solaris 2.x hosts and other hosts that recognize NFS are supported. See Chapter 11" for how to set up mounts.

- NIS+ provides centralized management of configuration files defining hosts, networks and users.

  A Trusted Solaris 2.5.1 NIS+ master can be used to manage data for Trusted Solaris 2.x NIS+ clients or Solaris 2.x NIS+ clients. Hosts running the 1.x version of the Trusted Solaris operating environment cannot be clients of a Trusted Solaris 2.x NIS+ master because Trusted Solaris 1.x uses NIS, not NIS+. Trusted Solaris 2.x hosts cannot be clients of Solaris NIS+ masters. See Chapter 12" for how to manage NIS+.

Note - Trusted Solaris 2.x NIS+ masters can be identified with the tsix host type to allow security attributes to be transmitted using token mapping.

- Solstice AdminSuite tools are used to centrally manage users and hosts.

  Solstice tools maintain most administrative data in NIS+ tables on the NIS+ master server (with the option of updating the corresponding local files on individual hosts, without relying on NIS+).

  To add new workstations (including diskless workstations) and servers to an already-configured distributed system, see the *Trusted Solaris Installation and Configuration* manual. See the other chapters of this manual for how to use the Solstice tools to administer users and hosts.

- Trusted networking and extended routing software supports trusted network communications.

The security administrator role enforces the desired degree of either openness or control for communications among hosts across networks. See “Goals of Trusted Networking ” on page 251.

Goals of Trusted Networking

The trusted networking software ensures the following:

- Proper labeling of data in network communications
- Enforcement of mandatory access control (MAC) rules when data is sent or received across the network
- Enforcement of MAC rules when data is routed to distant networks

The tnrhdb(4TSOL), tnrhtp(4TSOL), and the tnidb(4TSOL) trusted network databases store labels and other security attributes that apply to hosts and networks and to network interfaces. The trusted network databases are explained in detail in “Trusted Network Databases” on page 286 and the procedures are described in Chapter 10.”

Certain fields in the trusted network databases can be used by the security administrator to specify that communications from a Trusted Solaris machine are routed only through gateways that have a security level that matches the sensitivity of the data being transmitted. This type of routing is called:

- Trusted routing

The routing mechanism and databases used in the Trusted Solaris environment are discussed under “Routing” on page 261.

Trusted Solaris Network Examples

A single standalone homogeneous distributed system is the simplest, the easiest-to-protect, and the only network configuration that fully meets government evaluation criteria for protection of information transmitted across a network. A distributed system consists of one or more hosts that are:

- Running the Trusted Solaris 2.5 or 2.5.1 operating environment
- Administered by the same NIS+ master server, with the same set of security attributes for all machines and with each user uniquely identified on the network
- On the same wire (or configured so as to be virtually on the same wire)

Hosts administered this way can be considered to be within the same *security domain*. Within a single security domain, no routing table is needed, because a packet sent out on the wire is picked up by all other hosts on the same network.

Physical protection by the customer against tapping of such a network is assumed, because Trusted Solaris software does not provide encryption.

As already implied, a single host running the Trusted Solaris operating environment by itself is considered to be a standalone security domain.

Example of a Homogeneous Security Domain

Figure 9–1 shows a single security domain with host F as the NIS+ master. Each of the hosts is connected to the network by means of a single network interface. The *network interface* is the physical connector on a host that connects it to a network cable. Most often, the interface is an Ethernet connector to an Ethernet cable that links all the hosts in the local network. The *network accreditation range* for a host or for a network interface (which applies to all communications through the interface) is defined by a minimum sensitivity label and a maximum sensitivity label. The accreditation range is decided upon by the security administrator and specified by the security administrator role in the trusted networking databases.

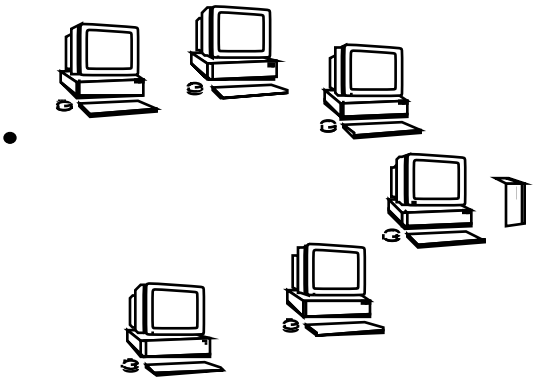


Figure 9–1 A Single Security Domain

Heterogeneous Networks

Trusted Solaris hosts can communicate with hosts running other operating systems whether the hosts are on the same wire or connected to another network.

When including other hosts on the local network that are running other operating systems at varying levels of trust, the security administrator should consider providing some measure of protection for the wire. In trusted network database

entries for other types of hosts, the security administrator needs to specify certain security attributes that are not transmitted from the other types of hosts and to the specify the labels at which communications are allowed with the other hosts. Figure 9–2 illustrates a heterogeneous network and some different types of hosts with which Trusted Solaris hosts can communicate.

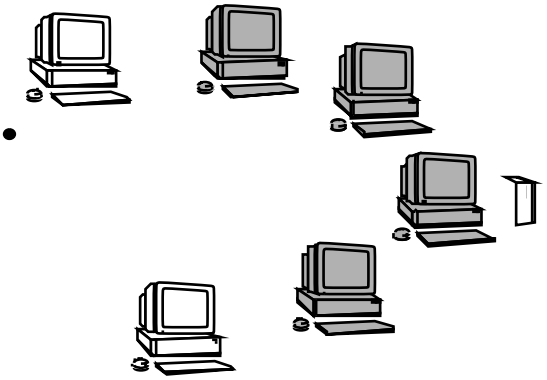


Figure 9–2 Heterogeneous Network

In the figure above, the NIS+ master host F could have the Solaris host as a NIS+ client if the client is running a version of Solaris that supports NIS+, while the CIPSO, RIPSO, and TSIX hosts would be configured as standalone systems on the network with their configuration information maintained in local files.

Host Types, Templates, and Protocols

In the entries made in a combination of the `tnrhdb` and `tnrhtp` databases, multiple hosts and networks may be associated with a specific *host type*. The host type identifies the *protocols* used by the kernel for interpreting a message, and the protocols tell the kernel which security attributes to look for in the packet header. The following table shows the host types for which entries can be made in the trusted network databases.

TABLE 9–1 Host Type Names

Type of Host	Name Used in Trusted Network Databases	Protocols and Notes
Trusted Solaris 2.5 or 2.5.1	sun_tsol	The TSOL protocol simplifies passing security attributes between Trusted Solaris 2.x machines. 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. Used only between Trusted Solaris 2.x machines.
TSIX/RE	tsix	Trusted Security Information Exchange for Restricted Environments protocol (uses token mapping to pass security attributes)
MAXSIX	msix	Supports network connectivity with Trusted Solaris 1.2 machines by providing the networking protocol used in Trusted Solaris 1.x. This protocol originated in MaxSix 1.0 and passes the security attributes in tokenized form in the IP option field. The label is passed in a CIPSO tag type 3.
TSIX(RE) 1.1 – CIPSO	cipso	Common IP Security Option protocol TSIX(RE) 1.1 is used to specify security labels that are passed in the IP options field. CIPSO labels are derived from the data's sensitivity label. Tag type 1 is used to pass the security label. This label is then used to make security checks at the IP level and may be used to label the data in the network packet.
RIPSO	ripso	Revised IP Security Option 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 MAC checks. An administratively-set fixed RIPSO label 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.
unlabeled	unlabeled	Assigned to hosts running Solaris or other unlabeled operating systems. Packets go to the unlabeled host at any valid sensitivity label between ADMIN_LOW and the sensitivity label specified in the default label entry in the host's template. To restrict packets to a single sensitivity label, the minimum and maximum sensitivity labels must be equal. Incoming packets are assigned the default CMW label (information label[sensitivity label] combination)

The host types *names* in the second column above are used in the trusted network databases.

The default `tnrhtp` database contains a set of example templates, one for each host type. The templates can be used as is or copied, renamed, or modified by the

security administrator. The names of all templates in the tnrhtp database appear in a list in the Database Manager, when the Database Manager is being used to assign a template to a host's IP address in the tnrhdb database. See Figure 9–3.

For example, to configure a Trusted Solaris 2.5.1 host named trusted, the security administrator would add an entry for the host in the `tnrhdb` database and assign any appropriate template from the `tnrhtp` database that has the sun_tsol host type.

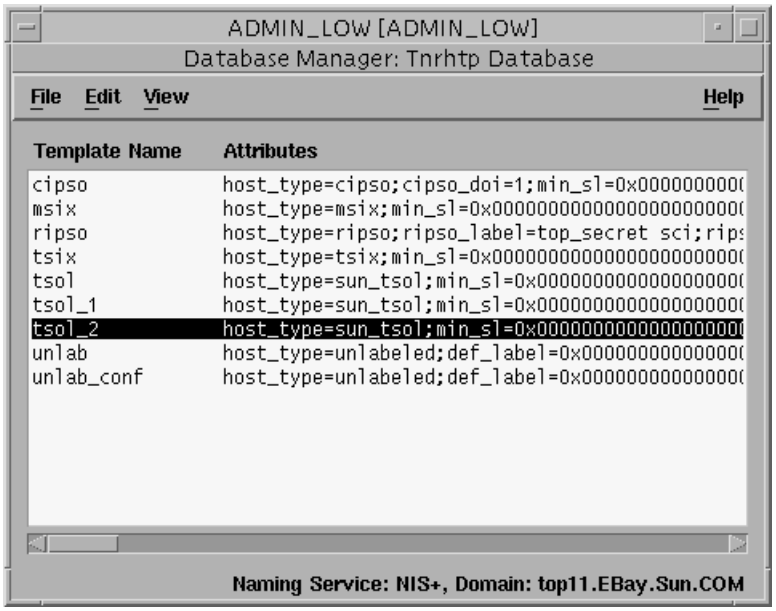


Figure 9–3 List of Template Names in the Tnrhtp: Add Menu

The security attributes that can be configured for each host type are shown in Table 10–1 in Chapter 10."

Example of Multiple Security Domains

Multiple security domains made up of multiple NIS+ domains may be connected by gateways. Figure 9–4 shows two security domains. Host F is the NIS+ master for security domain #1, and Host K is the NIS+ master for security domain #2. The two security domains are connected by gateway host G.

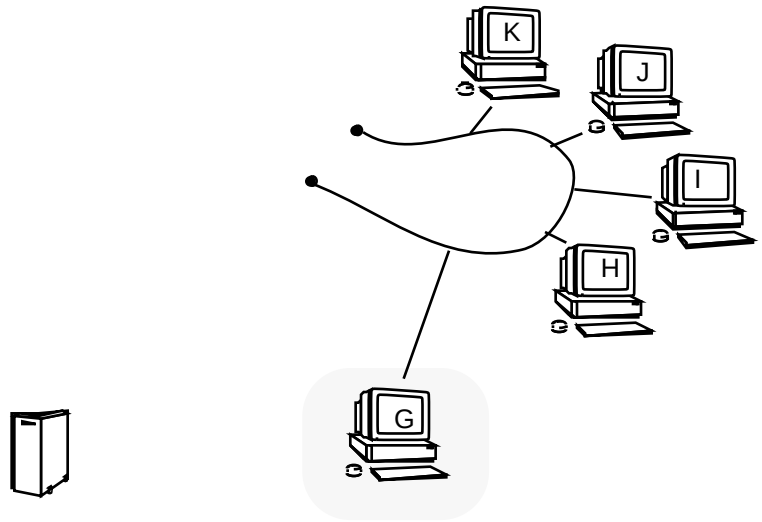


Figure 9–4 Two Security Domains

In Trusted Solaris administration terms, a *gateway* is usually a host that has more than one network interface. Multiple networks may be connected using Trusted Solaris gateways, which is the recommended configuration. Using a Trusted Solaris gateway allows the trusted networking software to use the attributes specified in the trusted network databases to enforce security policy on packets coming into and leaving the network and to support trusted routing of packets that have CIPSO and RIPSO labels.

Network Accreditation Range Requirements

Restricting a network's accreditation range in the trusted networking interface database, tnidb(4TSOL), on a gateway is one of the means by which the security administrator configures the appropriate degree of openness or control for communications with other networks.

The accreditation range is specified for most Trusted Solaris hosts as ADMIN_LOW to ADMIN_HIGH. With this accreditation range, every host in a homogeneous Trusted Solaris distributed system can receive packets from any other host in the same system at any valid sensitivity label. A gateway may be set up either to allow communications with hosts outside of the local distributed system at all sensitivity labels or to restrict communications to a subset of all labels.

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 sensitivity labels upon which those services depend. For example, communicating with the NIS+ master requires a network accreditation range of ADMIN_LOW to ADMIN_HIGH. For audit trail files at ADMIN_HIGH to be written to an audit server, the network interface needs to have ADMIN_HIGH its accreditation range.

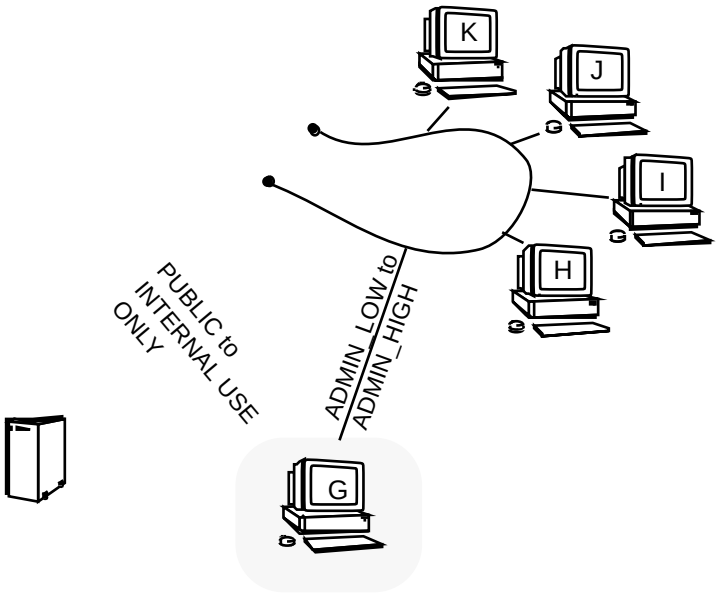


Figure 9–5 Two Security Domains With Differing Accreditation Ranges

To restrict communications, the `tnidb` entry on gateway G in Figure 9–5 could be specified as shown in Table 9–2

TABLE 9–2 Example tnidb Entry to Restrict a Network's Accreditation Range

Interface Name	Network Accreditation Range
le0 (interface to security domain #1)	PUBLIC to INTERNAL USE ONLY
le1 (interface to security domain #2)	ADMIN_LOW to ADMIN_HIGH

How Security Attributes Are Carried on the Network

Understanding how security attributes are added to packets is a prerequisite for understanding how various types of host are defined and for understanding how to specify security attributes in the trusted network databases. In addition, this topic is important for setting up trusted routing.

At the highest level, network packets have the format shown in Table 9–3:

TABLE 9–3 Packet Format

Ethernet Header	IP Header [IP Options]	TCP or UDP Header	Data	Ethernet Trailer

Both TCP/IP and UDP/IP can be used for network communications. When a host is specified with the Trusted Solaris 2.5 host type or the TSIX host type in its entries in the trusted networking databases, the trusted networking software inserts a *security attribute modulation protocol* (SAMP) header after the TCP or UDP Header and before the data. Therefore, when the host type is specified as Trusted Solaris 2.5 or 2.5.1 (sun_tsol) or TSIX, the packets have the format shown in Table 9–4:

TABLE 9–4 TSIX and Trusted Solaris 2.5 Packet Format

Ethernet Header	IP Header [IP Options]	TCP or UDP Header	SAMP Header	Data	Ethernet Trailer

The SAMP header carries the security attributes of the packet, including an attributes header that specifies whether the attributes are in binary or token form. The SAMP header is not used in any way in routing.

IP Options

The options field of the IP header (shown as [IP Options] in Table 9–4) can be used for including RIPSO or CIPSO labels and options in packets for the purpose of setting up trusted routing. There are two types of CIPSO options supported; tag type 1 for CIPSO hosts and tag type 3 for MSIX hosts.

On Trusted Solaris hosts, the tnrhdb(4TSOL)/tnrhtp(4TSOL) entries for a destination host can be configured so that the IP options field in outgoing packets carry either CIPSO or RIPSO labels information, which are used only for part of setting up trusted routing. See "CIPSO Labels in Packets" on page 259, below, "RIPSO Labels in Packets" on page 260, and "Creating Entries in the Trusted Network Databases" on page 304.

CIPSO Labels in Packets

The trusted networking software puts a CIPSO label and a CIPSO 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 MSIX host type
- Assigns the host the Trusted Solaris 2x host type, setting the IP label type to CIPSO
- Assigns the host the TSIX host type, setting the IP label type to CIPSO

If the IP Label field in a template is set to CIPSO, or if the remote host type is CIPSO, then tag type 1 is used. CIPSO tag type 3 is used when the remote host type is MSIX.

The CIPSO label that is carried in outgoing packets is derived by the trusted networking software from the actual sensitivity label associated with the data. Sometimes Trusted Solaris sensitivity labels might match directly to a CIPSO label. For example, even though a direct mapping of the sensitivity label of CONFIDENTIAL to the CIPSO label of CONFIDENTIAL is possible, most Trusted Solaris sensitivity labels probably 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 should plan ahead to configure the site's labels so they map well to CIPSO labels.

A CIPSO DOI (domain of interpretation) must also be specified, and the same CIPSO DOI must be shared by:

- The sending host
- All gateways through which messages travel and
- The destination host

By default, a message is dropped if it is sent at a sensitivity label that is too big to map to a CIPSO label. The Trusted Solaris 2.x ADMIN_HIGH sensitivity label is one example of a sensitivity label too big to map to a CIPSO label. As explained in *Trusted Solaris Label Administration*, the classification portion of the ADMIN_HIGH sensitivity label is 32767 bits and the compartments portion is 256 bits, all turned on.

The security administrator can add a switch to the `/etc/system` file to choose what action should be taken when a packet has the ADMIN_HIGH sensitivity label, whether:

- The label should be mapped to a CIPSO label with the highest classification and all compartments turned on, or
- The packet should be dropped.

See "To Substitute a Valid CIPSO Label for the ADMIN_HIGH Sensitivity Label " on page 333.

RIPSO Labels in Packets

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 tsol host type (shown as Trusted Solaris 2x in the Database Manager), and specifies the IP Label Type to RIPSO
- Assigns the host the TSIX Host Type, setting the IP Label Type to RIPSO

RIPSO labels on outgoing packets are administratively defined. The security administrator specifies them in the `tnrhtp` database, putting the classification in the RIPSO Send Class field and the compartment(s), or protection authority flags (PAF) in the RIPSO Send PAF field.

The following table shows the supported RIPSO Send classifications.

TABLE 9–5 Supported Classifications for RIPSO Labels

Supported Classifications for RIPSO Labels
Top_Secret
Secret
Confidential
Unclassified

The RIPSO Send PAF and Return PAF fields refer to Protection Authority Flags, which are shown in Table 9–6. PAFs specified in the Send PAF field are used like compartment names along with the classification within the RIPSO 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 RIPSO labeled packets. The classification sent back in the ICMP message is the same as the RIPSO classification in the packet.

TABLE 9–6 Protection Authority Flags that Can Be Specified in the RIPSO Send PAF or as RIPSO Return PAF Fields

Protection Authority Flags (may be used with RIPSO labels or specified as RIPSO errors)
GENSER
SIOP-ESI
SCI
NSA
DOE

Routing

The Trusted Solaris environment supports several methods for routing communications between networks, so that the security administrator 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

When packets are sent from one host to another on the same network, no routes or routers are needed. (Because gateways *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 performed at the destination.

When the source and destination hosts are on two different physical networks, the packet is sent from the source host to a gateway. The accreditation range of the destination and 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 2.5.1 gateways, accreditation range checks are performed. Otherwise, intermediate gateways do no accreditation checks, simply passing the packet along.

At some sites, it may be acceptable to route communications through any intermediate router, whether or not the router recognizes labels or enforces MAC. For this type of routing, the security level of the routers need not be considered.

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. Trusted Solaris 2.5.1 extensions enable *trusted* routing based on the shortest path to the destination *that also satisfies security requirements*. Trusted routing is achieved by including IP security options in a packet so that IP labels are available for accreditation range checks on intermediate routers. Trusted routing depends on all gateways recognizing RIP, the extended 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.

Modified TCP/IP Routing Features

Security policy at some sites may require that communications are routed only through routers whose level of trust exactly matches the sensitivity of the information being transmitted. (This type of routing, called trusted routing, is described in “Setting Up Trusted Routing ” on page 279.)

Some sites using trusted routing need to enable secure communications with Trusted Solaris 2.5.1 hosts that are on the other side of a cluster of unlabeled hosts and that go through one or more routers that do not understand labels. At these sites, the security administrator needs to set up tunneling. (The terms *clusters* and *tunneling* are defined under “Terms and Concepts” on page 263.”)

Before beginning to configure trusted networking and to set up routes, the security administrator needs to understand:

- TCP/IP routing, on which Trusted Solaris routing is based

The security administrator also needs to understand the following, as described in the rest of this section:

- New Trusted Solaris 2.x routing terminology
- Trusted Solaris 2.x extensions to routing
- The site’s routing requirements, and
- The Trusted Solaris features that are available to fulfil the site’s requirements

The following TCP/IP routing features have been modified to support Trusted Solaris extended security information:

- The routing table maintained in the kernel

- The routed(1MTSOL) routing daemon
- The Routing Information Protocol (RIP), and
- The route(1MTSOL) command
- The netstat(1MTSOL) command

Terms and Concepts

The terms in this section are in order of need to know, since each definition is built on the ones before it.

Routers

Strictly speaking, a router is any machine that has two or more network interfaces and that forwards packets from one network to another. The term *gateway* is often used interchangeably with *router*. Because security administrators usually need to choose routes carefully in the Trusted Solaris environment, they need to understand the security characteristics of all routers through which communications are passing.

For the highest degree of trust, routes should be set up with Trusted Solaris hosts 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.

A Solaris router does not drop packets when it finds labels it does not understand in the IP options section; it just passes the packets along. CIPSO, RIPSO, and MSIX routers do drop packets when they do not find the corresponding type of label in the IP options section of the packet. For example, a CIPSO router drops a packet if it does not find a CIPSO label in the packet's IP options section. Be aware of these considerations when setting up communications between hosts, and make sure that packets are routed through the appropriate types of routers.

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

When routing is needed, the routing software tries to find a route to the destination host in the route tables. When the host is not explicitly named, the 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.

Note - In some situations, multiple default gateways may lead to loops.

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 "SRI" on page 264" and "Emetric" on page 265, which are discussed later in this section.)

Routing table entries are created either of the following two ways:

- Dynamically

  routed(1MTSOL) routing daemon dynamically creates the route entries including the emetric

- Statically

  The administrator creates static routes manually in one of two routing files. The administrator chooses whether to supply an emetric with the route entry.

With a small network, it is possible for a system administrator 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, so they are invisible to almost all other systems, which almost always use dynamic routing.

With larger networks, to manually configure and maintain static routes on large networks is impossible. Dynamic routing is used more often than not.

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 SL
- Maximum SL

As described on the route(1MTSOL) man page, the SRI can also incorporate the security attributes listed below:

- CIPSO DOI
- RIPSO label

- RIPSO error
- CIPSO only
- RIPSO only
- MSIX only

Which attributes are included depends on one of two things:

- When the SRI is obtained by the dynamic routing software, the security information initially is derived from the `tnrhdb/tnrhtp` entries for the gateway on the router.
- When the SRI is entered manually in a static routing table, it is defined by the security administrator.

Extended RIP

Xerox Routing Information Protocol (RIP) version 1 is extended in the Trusted Solaris 2.5.1 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.

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(1MTSOL)` command. (See "Accreditation Checks" on page 271 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. The routing software selects the route whose SRI matches the security attributes on the packet.

sec_response Packets

In the base Solaris system, `in.routed` transmits a *request* packet on each interface and listens for *request* and *response* packets from other hosts. In the Trusted Solaris 2.5.1 system, every time `in.routed` sends a response packet, it also broadcasts a *sec_response* packet, which includes the SRI along with the route's metric. Similar to the response packet, sec_response propagates a route while adjusting its metric and SRI one hop at a time.

Trusted Solaris 2.5.1 gateways can propagate sec_response packets, but non-Trusted Solaris gateways cannot. To send security information about routes from a Trusted Solaris router to non-Trusted Solaris gateway(s) between two Trusted Solaris hosts, tunneling must be set up on the Trusted Solaris router(s) on the edge of the cluster. This implementation makes it possible to route within an Intranet containing a mixed environment of Trusted Solaris 2.x and non-Trusted Solaris 2.x systems. See "Clusters/Clouds" on page 266" described below and "Tunneling" on page 267.

Clusters/Clouds

For our discussion, a cluster is a collection (cloud) of zero or more hosts and one or more gateways. In a Trusted Solaris 2.5.1 cluster, all hosts are running either Trusted Solaris 2.5 or 2.5.1 and gateways are running the Trusted Solaris 2.5.1 operating environment, while all hosts or gateways in a non-Trusted Solaris 2.5.1 cloud are running another operating system or environment. When there is a non-Trusted 2.5.1 cluster between two Trusted Solaris 2.5.1 hosts that wish to communicate, tunneling must be used to supply emetrics for routes to the other side of the non-Trusted 2.5.1 cluster.

In the four illustrations that follow, the Intranet and the clusters it contains are shown as cloud shapes. Trusted Solaris 2.5.1 clusters are white clouds and gateways are in white circles. Non-Trusted Solaris 2.5.1 clusters are shaded clouds and the gateways are shaded squares.

The following figure shows an Intranet with three clusters. Trusted Solaris host H1 is connected by means of a Trusted Solaris 2.5.1 gateway to a non-Trusted Solaris cloud. Host H2 broadcasts response and sec_response packets to its network. Response packets are propagated to the network where host H1 resides, but not the sec_response packets. Without the sec_response packets, host H1 cannot obtain the emetric with the security information for the route to host H2 because the standard RIP running on the non-Trusted Solaris gateways in the intervening cluster cannot pass the extended RIP packets from H2's network to H1's network.

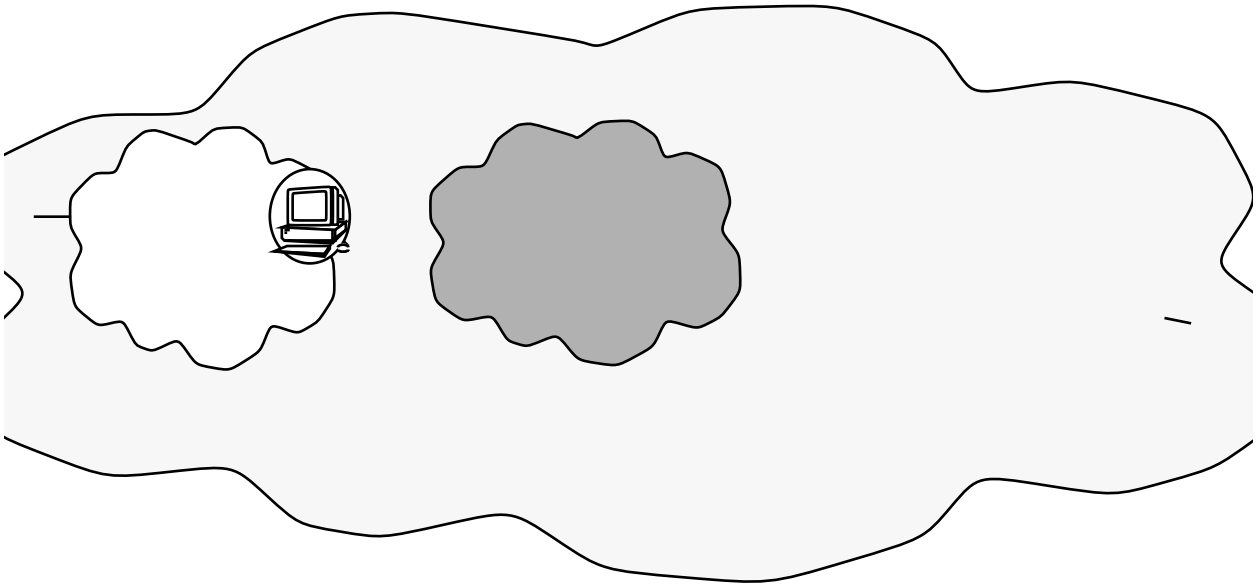


Figure 9–6 Example of Clusters Within an Intranet

Tunneling

Tunneling is only needed when trusted routing is being done using IP labels. When a non-Trusted Solaris 2.5.1 cluster using trusted routing exists between two Trusted Solaris 2.5.1 clusters, the security administrator must set up tunneling to make it possible for gateways in the two Trusted Solaris 2.5.1 clusters to exchange emetrics for routes. All clusters must be in the same Intranet. The Trusted Solaris gateways must be running Trusted 2.5.1 because dynamic routing and the extended RIP that handles emetrics are not in earlier releases. Figure 9–7 shows how tunneling is done.

A *forwarding host* is any Trusted Solaris router being set up to tunnel through one or more non-Trusted Solaris 2.5.1 gateway(s) to advertise the emetrics of its routes to Trusted Solaris 2.5.1 hosts on the other side. In Figure 9–7, a `tunnel` file on forwarding host gateway G1 contains the IP address of the network where host H2 resides. Trusted Solaris 2.5.1 routing software on G1 broadcasts the emetrics for the route to H1 directly to the network where H2 is connected. G2 obtains the emetric for the route to H1 and puts it in its routing table. (For two-way communications to occur, a tunnel file with the IP address of the network where host H1 resides would also need to be created on gateways G3 and G4.)

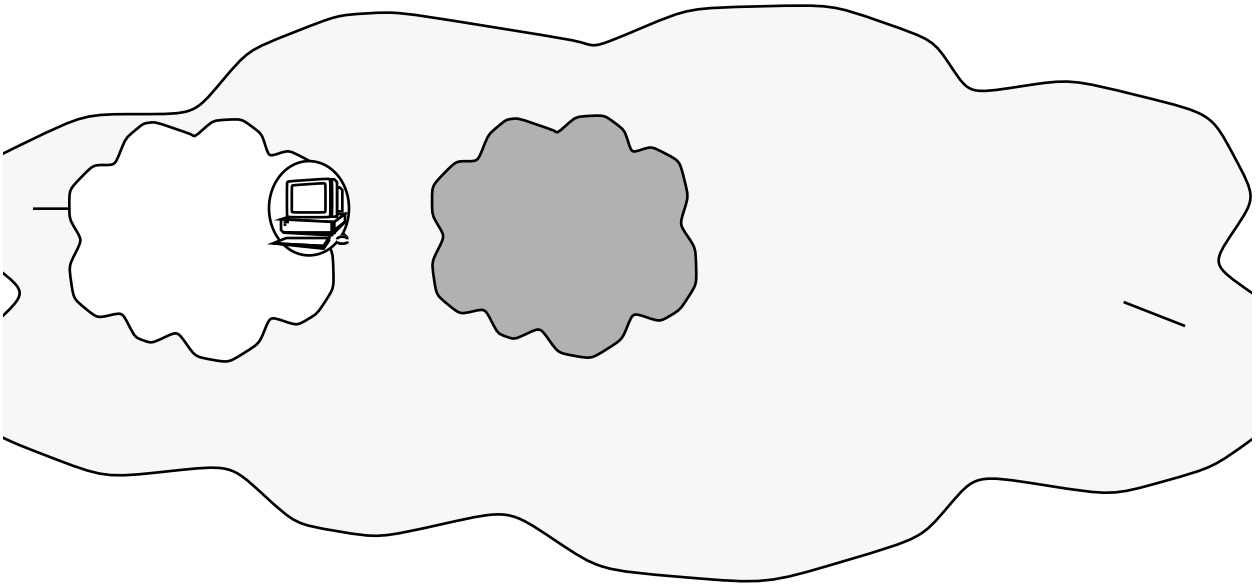


Figure 9–7 Tunneling Under a Non-Trusted Solaris 2.x Cluster in an Intranet

If the tunnel file exists with valid entries, the extended `in.routed` daemon broadcasts a special type of unlabeled response packet, the *sec_t_response* packet, with the emetric information of its routes directly to any networks listed in the tunnel file.

In Figure 9–7, a `tunnel` file on forwarding host gateway G1 contains the IP address of the network where host H2 resides. Trusted Solaris 2.5.1 routing software on G1 broadcasts a sec_t_response packet containing the emetrics for the route to H1 directly to the network where H2 is connected. By this means, G2 obtains the emetric for the route to H1 and puts it in its routing table. (For two-way communications to occur, a tunnel file with the IP address of the network where host H1 resides would also need to be created on gateways G3 and G4.)

Like the sec_response packet, the *sec_t_response* packet is broadcast whenever a response packet is sent out by the in.routed daemon from the Trusted Solaris router. The sec_t_response packet needs to be broadcast directly to the destination network because the sec_response packet cannot be passed through any non-Trusted Solaris 2.5.1 routers in between.

All non-TSOL routes in the non-Trusted 2.x cluster must have the same SRI, which means they are defined on the Trusted Solaris systems with the same default label and same IP label options (if any).

See the in.routed(1MTSOL) man page for more information, and see “Setting Up Tunneling” on page 311 and “To Set Up Tunneling” on page 341 in Chapter 10.

Static Routing

Static routing can be used only for communicating with other Trusted Solaris hosts and networks that also use static routing. Routers using static routing do not advertise their availability, so they are invisible to systems using dynamic routing. Either of the following files may be used by the `route(1MTSOL)`at boot time command uses to create routing entries in the kernel table.

- Establish default gateways using defaultrouter(4TSOL)
- Establish gateways for specific networks and default gateways using tsolgateways(4TSOL)

Dynamic Routing

Trusted Solaris 2.5.1 uses two dynamic routing protocols used in Solaris 2.5:

- Routing Information Protocol (RIP) version 1 implemented in in.routed(1MTSOL)
- ICMP Router Discovery Message implemented in in.rdisc(1MTSOL)

The RIP protocol has been extended.

`in.rdisc`

If the `/usr/sbin/in.rdisc` program exists on a host and if `/etc/defaultrouter` and `/etc/tsolgateways` files do not exist, ICMP Router Discovery (RDISC) is used for dynamic routing, and it installs a default gateway in the routing table.

By default, non-gateway hosts have `in.rdisc` enabled. To cause `in.routed` to run on a host, remove or rename `/usr/sbin/in.rdisc`. Default gateways set up by `in.rdisc` have no emetrics associated with them.

`in.routed`

Hosts run `in.routed` if `/usr/sbin/in.rdisc` does not exists. Hosts that are configured as routers (as described in the *TCP/IP and Data Communications Administration Guide*) automatically run both Routing Information Protocol (RIP) version 1 and RDISC protocols. `in.routed` installs a full routing table.

Types of Routing

Figure 9–8 summarizes how a Trusted Solaris host that is not a gateway determines whether which type of routing to do. See the `/etc/rc2.d/S69inet` script if you want to change the defaults.

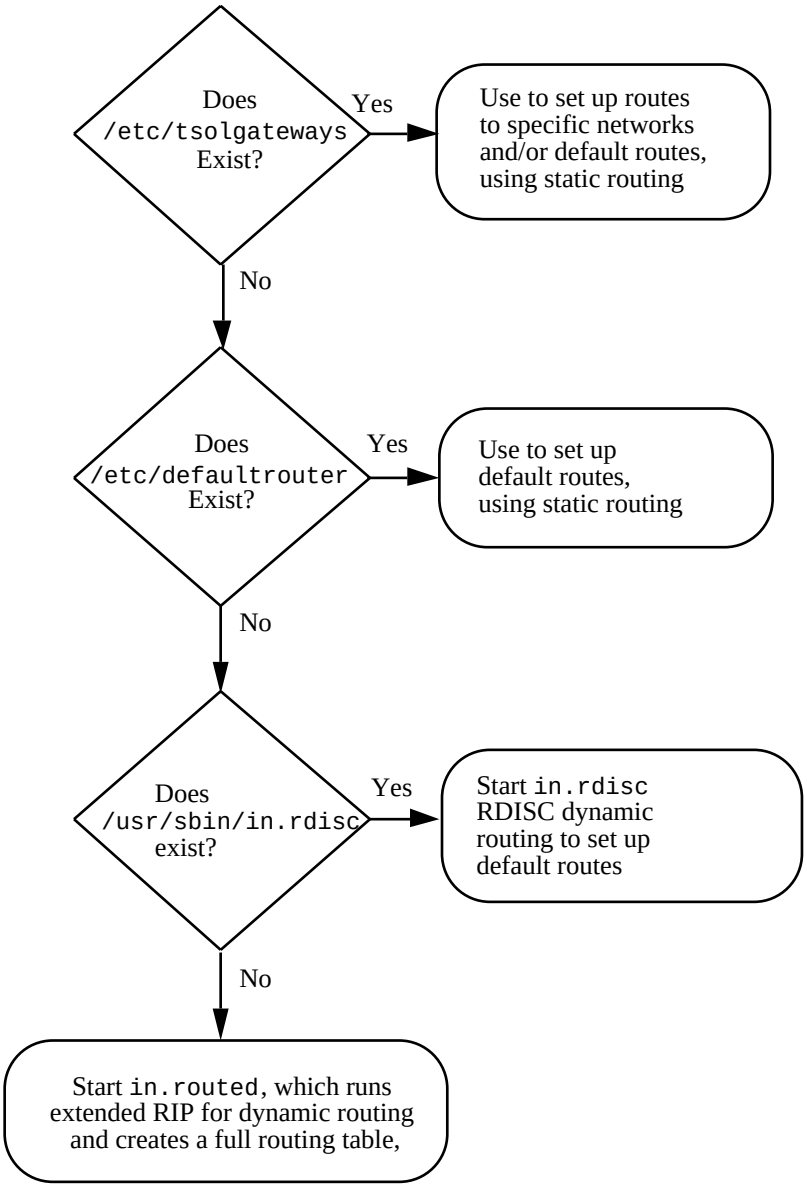


Figure 9–8 How a Host Determines Which Type of Routing to Do

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 (accreditation range and any CIPSO or RIPSO label information that may be specified) are obtained from a host's entries in the `tnrhdb`/`tnrhtp` files. 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 RIPSO 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 SL is compared to the SL range of the incoming and outgoing interface. Its SL is also compared to the SL range of the next hop gateway.

MAC Enforcement on Outgoing Messages

The following accreditation checks are performed on the sending host.

- The sensitivity 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 RIPSO label, then its RIPSO label and PAF flag must match the RIPSO label and PAF flag of the destination and of the route's emetric. If no emetric is specified for the route, the RIPSO label and PAF flag must match the RIPSO label and PAF flag of the first hop gateway.
- If the destination is specified as a MSIX host, then the sensitivity 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 for the next hop and of the network interfaces are performed.

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
  - 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 RIPSO label information, the following must be true for a packet to be forwarded:

- The route's emetric must include the RIPSO option. If no emetric is specified for the route, the next hop gateway's entry must be defined as either of the following:
  - RIPSO host type
  - tsol host type with a RIPSO IP label
  - tsix host type with a RIPSO IP label
- The RIPSO label of the packet and PAF must be the same as the RIPSO label and RIPSO PAF in the emetric of the route, or if no emetric is specified for the route, the packet's RIPSO label and RIPSO PAF must be the same as the RIPSO label and RIPSO PAF specified in next hop gateway's entry

If the sensitivity label of a message is not within the minimum and maximum sensitivity 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 sensitivity 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 CIPSO label, then its DOI must match the DOI specified in the receiving host's trusted network database entry
- If the packet has a RIPSO label, then its RIPSO label and PAF flag must match the RIPSO label and PAF flag specified in the trusted network database entry for the receiving host

For incoming communications, the Trusted Solaris networking software obtains sensitivity 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 that applies to the interface through which the message arrives.

Setting Up Static Routing

To set up static routing for communications outside of the local area network, the Trusted Solaris security administrator must specify gateways in one of two static routing tables in the `/etc` directory on each host. Two types of gateway entries can be specified in the tsolgateways(4TSOL) file.

- Network gateways

  A network entry routes all communications to the specified network exclusively through the named gateway. Emetrics can be specified for network gateways.

- Default gateways

  Routing table entries for default gateways allow communications through the named gateway to all networks that are not specifically listed.

A routing table with only network entries would restrict communications only to the specified network(s). A routing table with one or more network entries can also contain a default entry.

Default gateways for simple networks can be specified in the `/etc/defaultrouter` file [which is described in the route(1MTSOL) man page].

As mentioned in “Static Routing” on page 269, the trusted networking software looks first for `tsolgateways`, and if one exists, the `defaultrouter` file is not consulted. If neither file exists, dynamic routing is done.

The number entered into the metric field of the `tsolgateways` file is equal to the number of gateways between the originating host and the destination. A transmission to a host on the same network has 0 hops. Figure 9–9 shows four hosts (H1, H2, H3, and H4) on the same network with 0 hops.

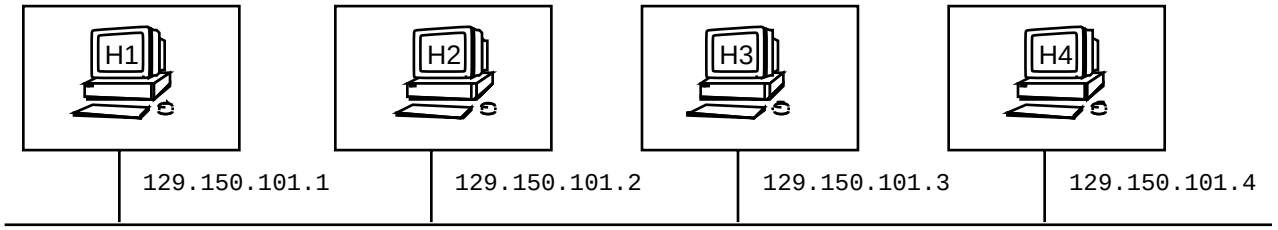


Figure 9–9 Example of 0 Hops for Communications Between Four Hosts in a Single Security Domain

A transmission to a network that is connected directly to the gateway of the originating network has 1 hop. Figure 9–10 shows network `129.150.101.0` connected directly to `129.150.102.0` by gateway G. Gateway G has a host name for each of its interfaces: gateway-101 is the name for the interface connected to the 101 network, and gateway-102 is the name for the interface connected to the 102 network. Since `/etc/defaultrouter` can be used for simple networks with only one gateway, hosts 1, 2, 3, and 4 on net `129.150.101.0` could have a `defaultrouter` file specifying only gateway-101. If the `tsolgateways` file is used, the network address needs to be specified as `129.150.102.0` and the metric would be specified as 1 hop.

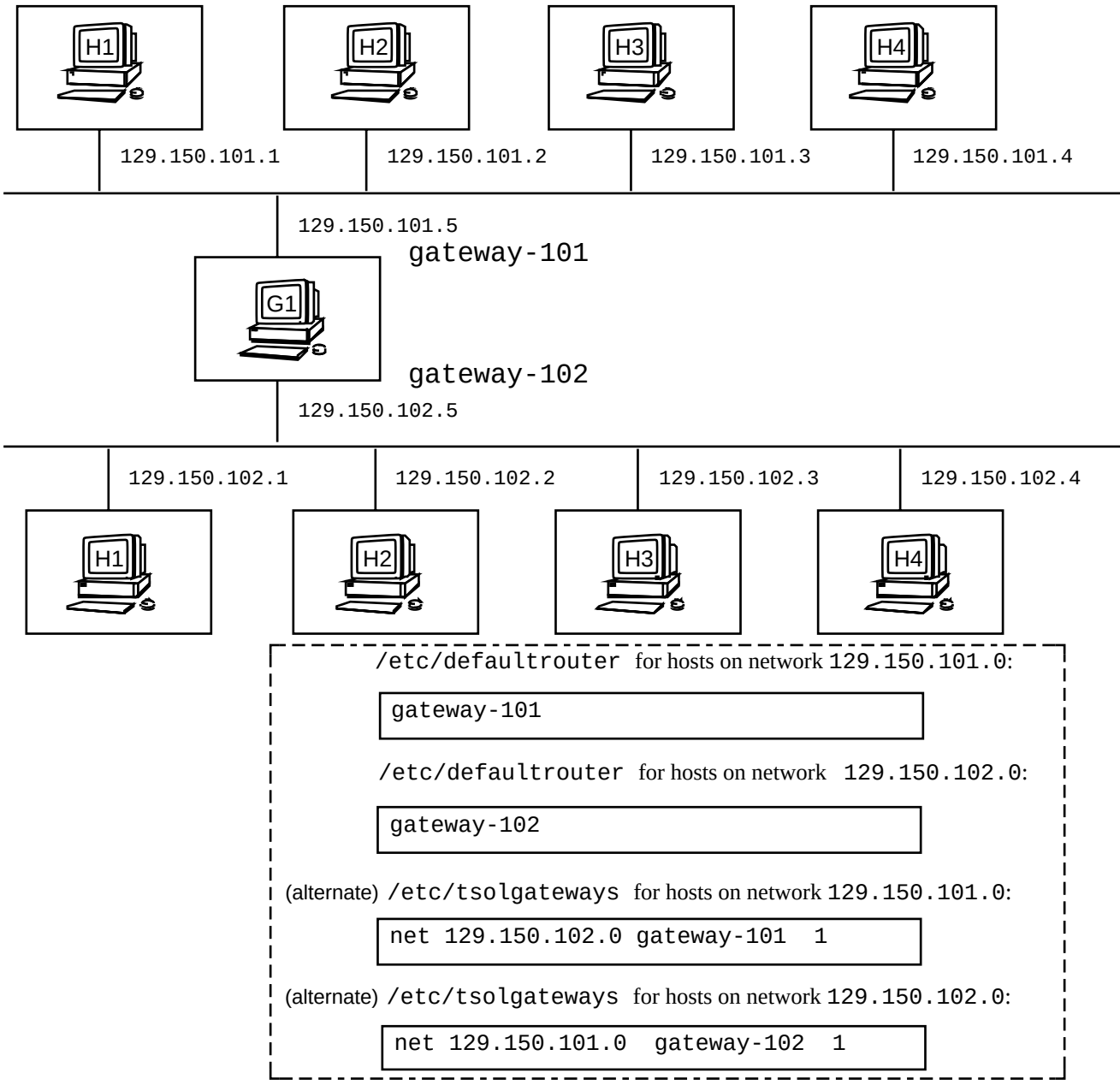


Figure 9–10 Example: Default and Network Routes for Two Security Domains with a Single Gateway

Transmission to a network two gateways away takes two hops as shown in Figure 9–11. Figure 9–11 shows an example `tsolgateways` file for hosts 1, 2, 3, and 4 on the network whose IP address is `129.150.110.0`. As illustrated, gateway G1

connects network `129.150.101.0` to network `129.150.102.0`, and gateway G2 connects network `129.150.102.0` to network `129.150.103.0`. The first entry in `tsolgateways` sets a network route to `129.150.102.0` through gateway1-101, and the metric 1 indicates one hop to the destination network. The second entry sets a network route to `129.150.103.0` through gateway1-101, and the metric 2 indicates two hops to the destination network.

Figure 9–12 shows a more complex network topology, with gateways G1, G2, and G3.

For the procedure, see “To Set Up a Simple Default Route for a Network with One Gateway ” on page 334.

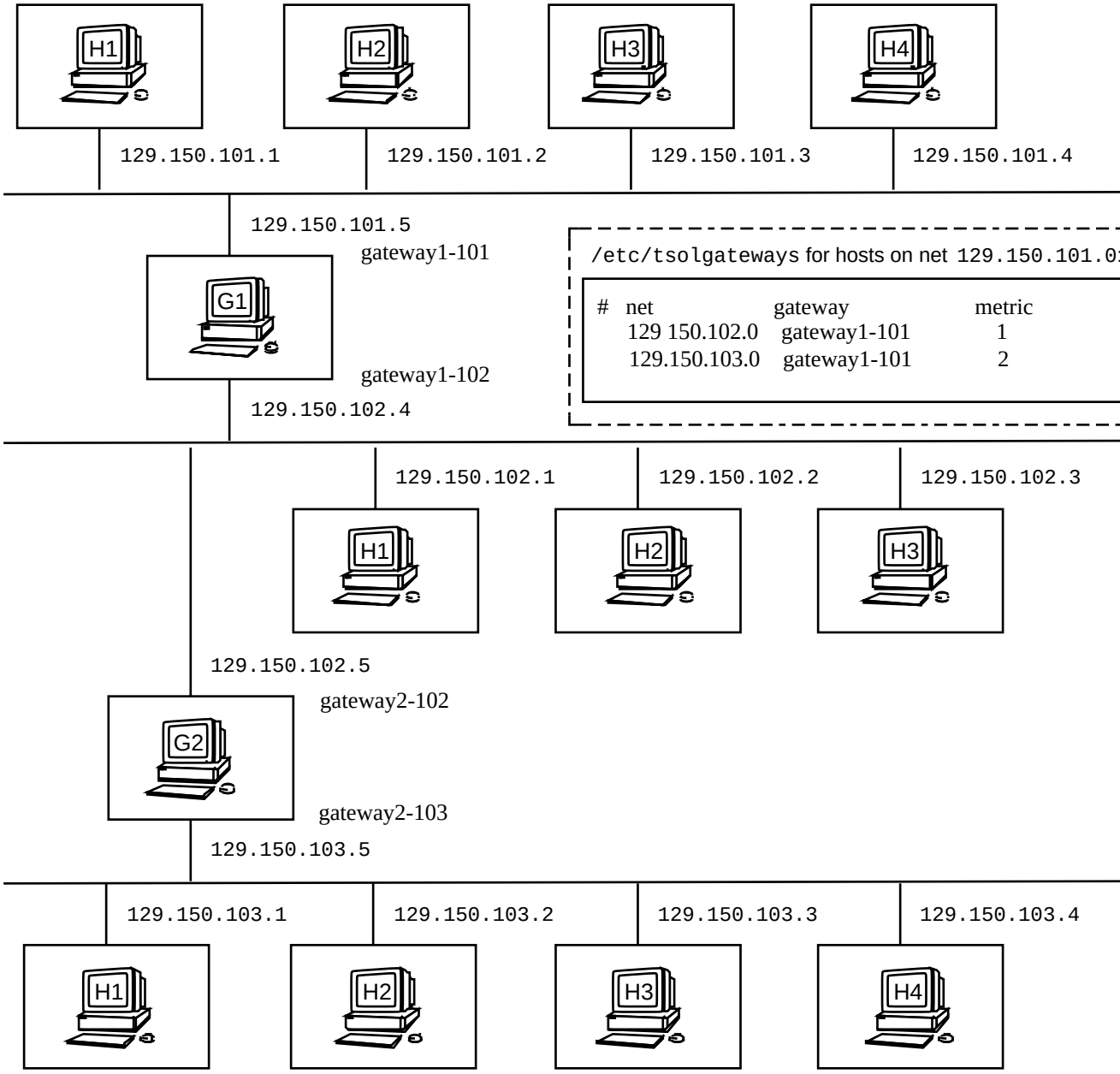
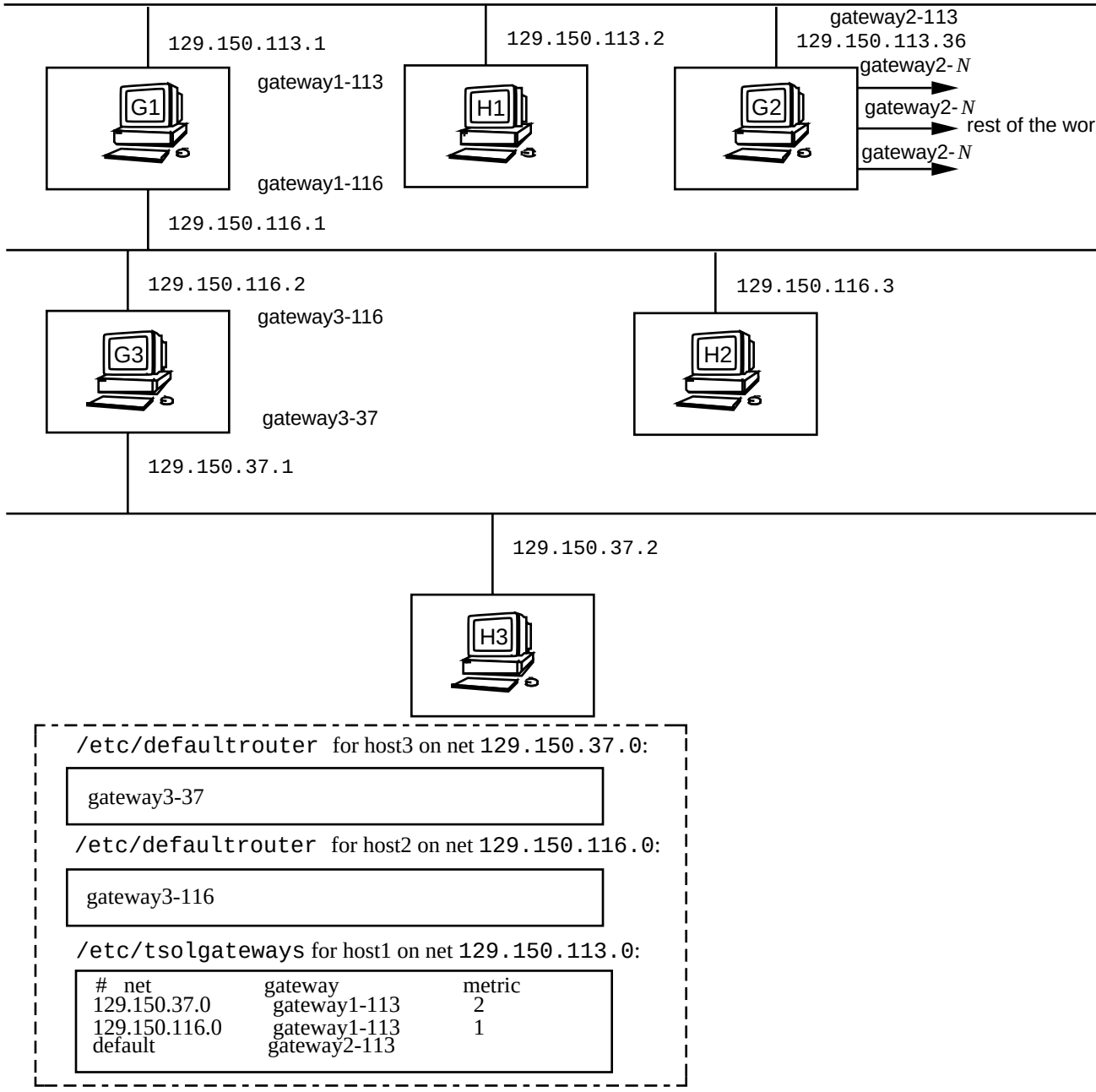


Figure 9–11 Example `tsolgateways` File for Communications Among Three Networks



`/etc/defaultrouter` for host3 on net `129.150.37.0`:

```
gateway3-37
```

`/etc/defaultrouter` for host2 on net `129.150.116.0`:

```
gateway3-116
```

`/etc/tsolgateways` for host1 on net `129.150.113.0`:

```
# net           gateway          metric
129.150.37.0    gateway1-113     2
129.150.116.0   gateway1-113     1
default         gateway2-113
```

Figure 9–12 Example Complex Gateway Configuration with Routing Tables

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 that matches the sensitivity of the data being sent out. This section assumes you have read and understood “Routing” on page 261. See “To Set Up Trusted Routing ” on page 336 of Chapter 10 for the procedure.

Routing based on CIPSO and RIPSO labels is called trusted routing because putting the CIPSO or RIPSO label in the IP portion of a packet makes it possible to ensure that the packet is routed only through gateways that have the same label as the packet. Trusted routing requires that the security administrator 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 RIPSO IP label (classification) and RIPSO PAF (compartments), the combined RIPSO classification and compartments assigned to the destination must match the RIPSO classification and compartments assigned to the sending host. The template for the first hop gateway must define a RIPSO label and PAF that match the destination. When an IP label is specified by the security administrator 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 sensitivity label of the packet. When the IP label type is RIPSO, the trusted network software uses the RIPSO 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.

Example of Trusted Routing Considerations

The following simple example and figures illustrate the rules described in the previous section. Hosts H1 and H2 are Trusted Solaris 2.5.1 hosts that want to communicate using trusted routing based on CIPSO IP labels. H1 is running `in.rdisc`, which has set up a default route to G1. There may or may not be other Trusted Solaris gateways between G1 and H2.

Figure 9–13 shows the definitions that must be made in the `tnrhdb/tnrhtp/tnidb` files on source host H1 for H1 to be able to route a packet through G1 to H2. The packet has a sensitivity label of CONFIDENTIAL, which in this case maps directly to a CIPSO label of CONFIDENTIAL

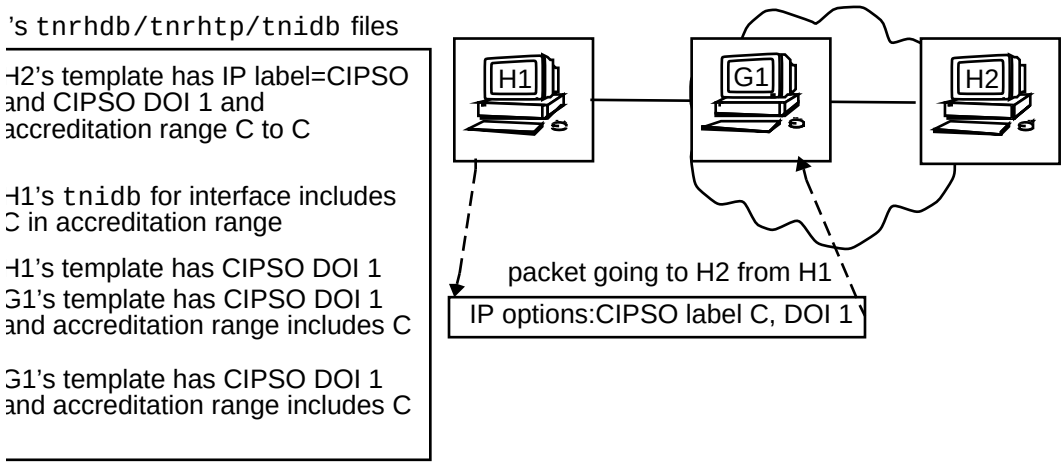
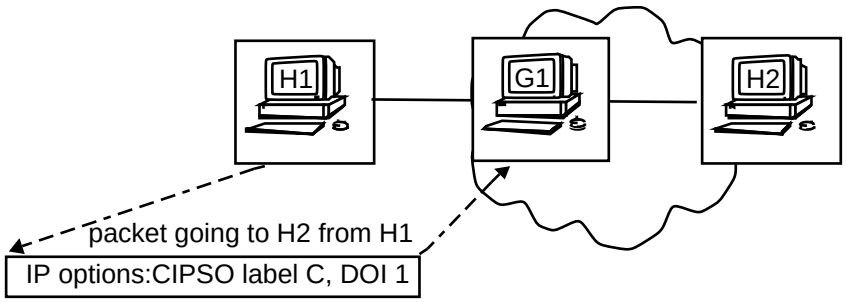


Figure 9–13 Trusted Network File Definitions on a Sending Host

On a sending host, the trusted network software makes the following checks before it sends out a packet:

- Checks H2's template and finds it specifies an IP label of type CIPSO and a CIPSO DOI of 1. The CIPSO label derived by the software for the packet is C. Since C is within the accreditation range for the H2, which is confidential to confidential, the next check is done.
- Checks H1's template to make sure that H1 has a CIPSO DOI that matches H1's CIPSO DOI. It also checks that the packet's label is within the accreditation range of the outgoing interface, which is configured in the template assigned to H1.Since both checks pass, the next check is done against G1's template.
- Checks G1's template to make sure it has the same CIPSO DOI and that its accreditation range includes the CIPSO label of the packet.

All the checks are passed, and the trusted network software inserts the CIPSO label C and CIPSO DOI 1 into the packet and sends it to G1.Figure 9–14 continues the illustration in Figure 9–13, illustrating the checks performed on a gateway before forwarding a packet.



G1's `tnrhdb/tnrhtp/tnidb` files (two `tnrhdb` entries, one for each network IP address associated with each network interface)

G1's incoming interface's `tnrhtp` template has IP label CIPSO and CIPSO DOI 1

Incoming interface's `tnidb` accreditation range includes C

If forwarding to another gateway, G2's template has CIPSO label (either CIPSO router or Trusted Solaris 2.5.1 router) with CIPSODOI 1 and accreditation range includes C

Figure 9–14 Checks Performed on a Gateway Before Forwarding Packets

On a gateway, the trusted network software makes the following checks before it forwards a packet:

- Checks the IP options portion of the packet and finds it specifies an IP label of type CIPSO and a CIPSO DOI of 1
- Makes sure the template assigned to the incoming network interface has the CIPSO IP label and CIPSO DOI 1
- Makes sure the accreditation range of the incoming interface includes C, by checking its `tnidb` entry.
- If it were forwarding the packet to another gateway, the trusted network software would make sure the template for G2 has IP label type CIPSO and CIPSO DOI 1 and its accreditation range includes C.

Figure 9–15 continues the illustration in Figure 9–14, illustrating the checks performed on the destination host when it receives a packet.

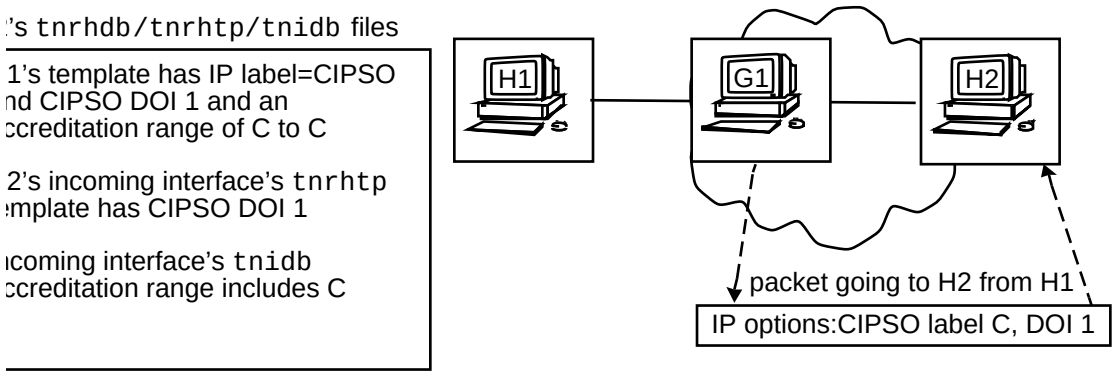


Figure 9–15 Checks Performed on a Receiving Host

On the destination host, H2, the trusted network software makes the following checks before it receives a packet:

- Checks the IP options portion of the packet and finds it specifies an IP label of type CIPSO and a CIPSO DOI of 1
- Checks H1's template and finds it specifies an IP label of type CIPSO and a CIPSO DOI of 1. The CIPSO label derived by the software for the packet is C. Since C is within the accreditation range for the H1, which is confidential to confidential, the next check is done.
- Makes sure that the tnrhtp entry for H2's incoming interface has a CIPSO DOI of 1 and that the accreditation range in the `tnidb` entry for the incoming interface includes C.

The security administrator role begins by specifying entries with the same host type and with the same IP label type for:

- the first hop gateway, and
- the destination host

To specify the IP label, the host type must be either:

- Trusted Solaris 2.x
- TSIX
- CIPSO or
- RIPSO

and the IP Label Type must be specified the same, either:

- RIPSO or
- CIPSO

For example:

To specify an IP label for a gateway running a non-Trusted Solaris system supporting RIPSO or a commercial RIPSO router, enter the host type of:

- ripso

To specify the IP label for a gateway router running a non-Trusted Solaris system supporting CIPSO or a commercial CIPSO router, enter the host type of:

- cipso

To route through a Trusted Solaris 1.x host using the cipso type 3 IP Labels on specify:

- msix

A Trusted Solaris 2.5.1 gateway may be identified as a sun_tsol, tsix, cipso, or ripso host type in the template assigned to the gateway and then assigned an IP label of type RIPSO or CIPSO in the IP labels field. Non-Trusted Solaris hosts or commercial routers that support RIPSO or CIPSO labels can also be specified using the ripso or cipso host type. Trusted Solaris 1.x hosts carry type 3 CIPSO labels in their packets, so they cannot be used for trusted routing.

The CIPSO label is derived by the trusted networking software from the sensitivity label of the packet, while the RIPSO label must be specified by the security administrator in the RIPSO Class and RIPSO Send PAF fields in the template.

When a packet is being forwarded on a gateway, the CIPSO or RIPSO label of the packet is consulted by the trusted networking software and compared with emetric information of the route, if available from routing tables, or with the security information in the trusted network databases entries for the first hop gateway, to ensure the message is routed only through gateways at the appropriate level of trust.

The trusted networking software only looks at the portion of the packet shown in Table 9–7. The Trusted Solaris and TSIX labels and other security attributes are stored in inaccessible portions of the packet between the TCP/UDP Header and the Ethernet Trailer, while the CIPSO and RIPSO labels are stored in the accessible IP Header portion of the packet:

TABLE 9–7 Portions of a Packet Accessible to the Trusted Networking Software

Ethernet Header	IP Header [RIPSO Label in IP Options]	TCP or UDP Header	. . .	Ethernet Trailer

Using a combination of Trusted Solaris and TSIX host type specified with CIPSO or RIPSO IP labels enables packets to be sent with Trusted Solaris or TSIX security attributes to destination hosts that understand the security attributes. Additionally, the trusted networking software on *Trusted Solaris routers on the way to the destination can look at the CIPSO or RIPSO label in the accessible IP portion of the packet to control the routes the packet takes along the way. Without the IP label, the trusted networking software on a gateway would not be able to ascertain what a packet's label is and be able to do the necessary accreditation checks on the next hop,

because the trusted networking software does not look in the part of the packet where the label and other security attributes are stored.

Note - Because the CIPSO label is obtained from the actual sensitivity label of the data being sent out, the security administrator does not specify a CIPSO label. When the IP label type being used is CIPSO, the security administrators in each of the security domains through which communications using CIPSO labels are being routed must agree among themselves about which Trusted Solaris sensitivity labels are going to be used for communications between them, since the sensitivity label is going to be converted into a equivalent CIPSO label by the trusted networking software. The security administrators also need to agree on the same CIPSO DOI to specify for all the host types in the previous list.

Note - Because the RIPSO label and RIPSO error are administratively defined, each of the hosts in the list should have the same RIPSO label and RIPSO error. When the IP label type being used is RIPSO, the security administrators in each of the security domains through which communications using IP labels are being routed must agree among themselves about which RIPSO label, protection authority flags, and RIPSO error to specify for all the host types in the previous list.

Allowing a Single-label Gateway to Forward Packets at Multiple SLs

Every single-label host (specified with a host type of unlabeled or ripso) must be assigned a default sensitivity label in its template. The default sensitivity label is used for single-label network communications with the unlabeled host. An accreditation range assigned to each unlabeled host is used in routing to allow a single-label gateway to forward packets that it would not otherwise be allowed to receive based on its default sensitivity label alone.

A minimum and a maximum sensitivity label in the unlabeled host's template establishes the accreditation range that is used for routing. The accreditation range of the 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 accreditation range.

The default label range of ADMIN_LOW to ADMIN_HIGH is set in the default `unlab` template. The security administrator can adjust the default label range as desired.

CHAPTER 10

Specifying Security Attributes in Trusted Network Databases and Setting Up Routing

This chapter provides procedures for the security administrator to follow when specifying which hosts and networks are allowed to communicate with the Trusted Solaris system and which security attributes apply to communications with these hosts and networks. This chapter provides the following information:

- What values should be set for various security attributes in the trusted network database
- How the values set in the trusted network database are put to use
- How to set up routing and trusted routing

Chapter 9 and the sections listed below provide needed background information:

- "Trusted Network Databases" on page 286
- "Templates Assigned to Host Types in the Template Manager" on page 289
- "Creating Entries in the Trusted Network Databases" on page 304
- "Precedence Rules for Attributes in Trusted Network Databases" on page 305
- "Special Boot-time Trusted Network Databases" on page 309
- "Administering the Boot-time Trusted Network Databases" on page 310
- "Setting Up Tunneling" on page 311

This chapter includes the following procedures.

- "To Change the Default Entry in the Boot-time `tnrhdb/tnrhtp` Files" on page 312
- "To Access the Trusted Network Databases from the Database Manager" on page 313

Trusted Network Databases

The security administrator uses the Trusted Solaris version of the Solstice Database Manager, shown in the following figure, to make entries in the tnrhdb(4TSOL), tnrhtp(4TSOL), and tnidb(4TSOL) trusted network databases.

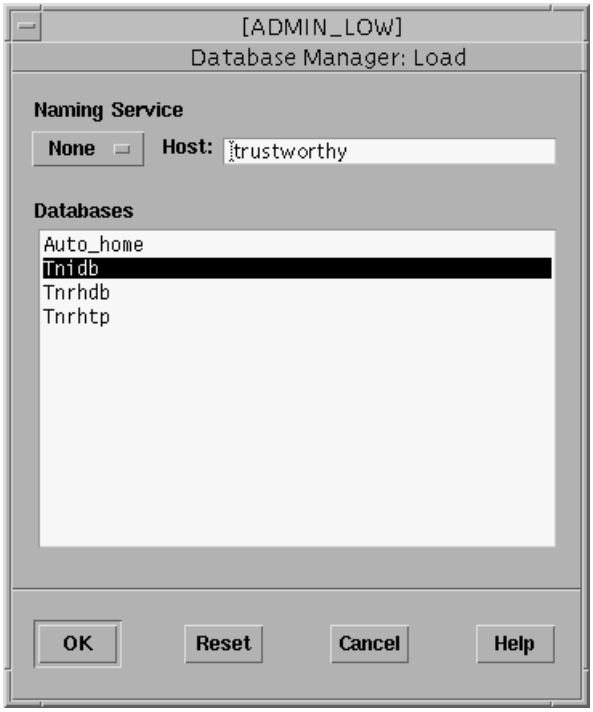


Figure 10–1 Tnidb Selected in the Database Manager: Load List

- The `tnrhtp` file contains templates that define sets of security attributes that apply to various host types.
- The `tnrhdb` file is used to assign templates from the `tnrhtp` file to hosts and networks.
- The `tnidb` file is used to assign security attributes to network interfaces.

The values specified in the `tnidb` file are consulted, along with any attributes specified in the `tnrhtp/tnrhdb` files for individual hosts and networks, to determine which security attributes apply to communications that come through the interface. (These topics are discussed in more detail in "MAC Enforcement on Incoming Messages" on page 272 and "Precedence Rules for Attributes in Trusted Network Databases" on page 305.)

Before the Database Manager can be used, a naming service must be selected. The two choices are:

- NIS+
- None

See "Using Solstice Administrative Tools in the Solstice_Apps Folder" on page 10, if needed, for how to bring up the Database Manager and select a naming service.

Because the `tnidb` file's values apply to the network interfaces on the local host, the `tnidb` file is *always* a local file in `/etc/security/tsol`.

The `tnrhdb` and `tnrhtp` files are usually administered as NIS+ tables. However, a standalone Trusted Solaris host may be configured with no naming service, and on such a host with no naming service, these files are kept in `/etc/security/tsol`.

The kernel caches all the information from `tn*db` databases. When no naming service is selected, the kernel cache is automatically updated on the local host after any changes are made using the Database Manager. When the NIS+ naming service is selected, it can take up to 1/2 hour for any `tn*db` changes to be pulled by NIS+ clients from the NIS+ master.

The Trusted Solaris version of `nsswitch.conf`(4) includes entries for `tnrhtp` and `tnrhdb` as shown below.

```
# TSOL
tnrhtp: files nisplus
tnrhdb: files nisplus
```

To modify these entries, the administrator role uses the Name Service Switch action. (See "To Launch Administrative Actions" on page 29, if needed, for how to access the Name Service Switch action.) To preserve the correct file attributes (owner, group, mode and label), this file should never be edited directly.

Security Attributes Configurable for Each Host Type

The security attributes that can be configured for each host type are shown in the following table. See the `tnrhtp` man page for more information about what is required compared to what is allowed.

TABLE 10–1 Security Attributes by Host Type

Host Type	Security Attributes	Default Template Name
unlabeled	sensitivity label, information label, clearance, UID, GID, forced privileges, audit UID, audit mask, audit terminal ID, audit session ID, (minimum SL and maximum SL for gateway hosts)	unlab
sun_tsol	allowed privileges, minimum SL and maximum SL, IP label, RIPSO label, RIPSO error, CIPSO DOI, audit UID, audit mask, audit terminal ID, audit session ID	tsol tsol_1: tsol with IP label RIPSO tsol_2: tsol with IP label CIPSO

TABLE 10–1 Security Attributes by Host Type *(continued)*

Host Type	Security Attributes	Default Template Name
ripso	sensitivity label, information label, clearance, UID, GID, forced privileges, RIPSO label, RIPSO error, audit UID, audit mask, audit terminal ID, audit session ID, (minimum SL and maximum SL for gateway hosts)	ripso
cipso	clearance, information label, UID, GID, forced privileges, minimum SL and maximum SL, CIPSO DOI, audit UID, audit mask, audit terminal ID, audit session ID	cipso
tsix	sensitivity label, information label, clearance, UID, GID, allowed privileges, forced privileges, minimum SL and maximum SL, IP label, RIPSO label, RIPSO error, CIPSO DOI, audit UID, audit mask, audit terminal ID, audit session ID	tsix
msix	sensitivity label, information label, clearance, UID, GID, minimum SL and maximum SL, audit UID, audit mask, audit terminal ID, audit session ID	msix

The third column in the table above lists the names of all the default templates shipped with the system. The sun_tsol row lists three default templates. The tsol template defines an accreditation range of ADMIN_LOW to ADMIN_HIGH, all allowed privileges, no IP Label, a RIPSO Send Class of Top Secret, a RIPSO Send PAF: of SCI, a RIPSO Return PAF: of GENSER, and a CIPSO Domain of 1. The tsol_1 template defines a sun_tsol host with the IP Label set to RIPSO but with no RIPSO label defined. The tsol_2 template defines a sun_tsol host with the IP Label set to CIPSO and a CIPSO Domain of 1.

Templates Assigned to Host Types in the Template Manager

Note - Once a host type has been selected in the Template Manager, fields that are not settable for the host type are grayed out and not active.

The sections listed belowdescribe the supported host types with figures that show the templates and fields that are supported for each type.

■ “Trusted Solaris 2.x (`sun_tsol`) Host Type” on page 290

- “TSIX (`tsix`) Host Type” on page 291
- “MSIX (`msix`) Host Type” on page 296
- “CIPSO (`cipso`) Host Type” on page 297
- “RIPSO (`ripso`) Host Type” on page 299
- “Unlabeled (`unlabeled`) Host Type” on page 301

Trusted Solaris 2.x (`sun_tsol`) Host Type

Ethernet Header	IP Header [IP Options]	TCP or UDP Header	SAMP Header [Binary Attributes]	Data	Ethernet Trailer

When the sun_tsol host type is specified, the trusted networking software uses the SAMP protocol for sending security attributes. The security attributes are represented in binary form, without token mapping.

Note - Diskless clients of Trusted Solaris servers must always be specified with the sun_tsol host type.

If the IP Label Type field is set in a Trusted Solaris 2.x type host’s entry to CIPSO or RIPSO, IP options in the IP Header are used to carry CIPSO or RIPSO labels that can be used for trusted routing of communications from Trusted Solaris 2.x hosts. See “Creating Entries in the Trusted Network Databases” on page 304. Also see “CIPSO Labels in Packets” on page 259.

The security attributes that can be specified for Trusted Solaris 2.5 hosts are shown in Figure 10–2. Fields that are not settable for the host type are grayed out and not active.

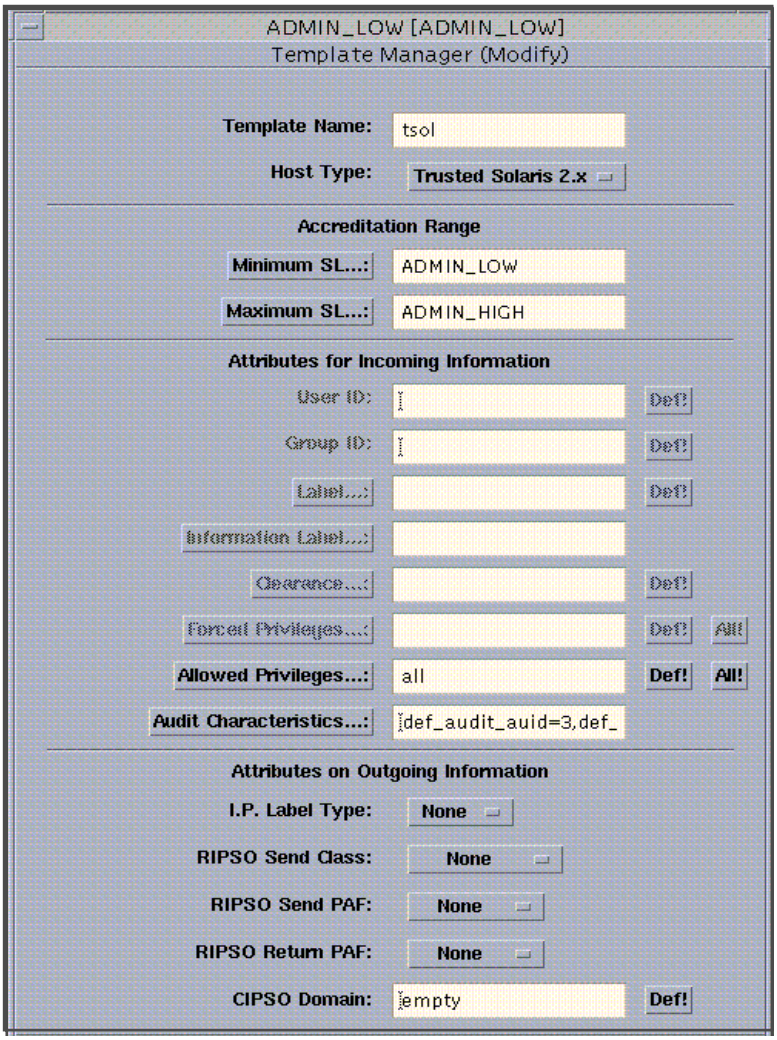


Figure 10–2 Configurable Fields in the Tnrhtp for the Trusted Solaris 2.x Host Type

TSIX (`tsix`

) Host Type

Ethernet Header	IP Header [IP Options]	TCP or UDP Header	SAMP Header	Data	Ethernet Trailer

When a TSIX host type is specified, the trusted networking software communications with that host observe the TSIX standard established by the Trusted Systems Interoperability Group (TSIG), of which Sun Microsystems Federal, Inc. is a member. For TSIX host types, the trusted networking software uses the SATMP token mapping protocol. An Attribute header in the SAMP header specifies that the attributes are in token form. Except that attributes are transmitted in token form rather than in binary form, the packets have the same format as the sun_tsol packets.

TABLE 10–2 Security Attributes for the `sun_tsol` Host Type

Configurable Attributes	**Notes**
minimum SL (min_sl=)	Specify ADMIN_LOW or use another valid sensitivity label to restrict communications with the sun_tsol host.
maximum SL (max_sl=)	Specify ADMIN_HIGH or use another valid sensitivity label to restrict communications with the sun_tsol host.
allowed privileges (allowed_privs=)	Specify to limit the effective privilege set propagated across the network by this host (both inbound and outbound). Only specified privileges are allowed. “all” means that all privileges are interpreted; “none” means that no privileges are interpreted. In the tnrhtp template that applies to the host, allowed_privs=empty; means that whatever is specified in the `tnidb` for this field applies. In the `tnidb` file, empty means that no privileges are applied.
IP label (ip_label=)	Valid types are: none, ripso, cipso
RIPSO label (ripso_label=)	If IP label=empty, ripso_label=empty. If IP Label Type is RIPSO, one of the supported Classification Level Encodings must be specified: Top_Secret, Secret, Confidential, Unclassified.
RIPSO error (ripso_error=)	If IP label=empty, ripso_error=empty. Must be specified if IP Label Type is RIPSO. Supported Protection Authority Flags: GENSER, SIOP-ESI, SCI, NSA, DOE. The Classification Level is taken from the ripso_label field.

TABLE 10–2 Security Attributes for the `sun_tsol` Host Type *(continued)*

Configurable Attributes	Notes
CIPSO DOI (cipso_doi=)	If IP label=empty, cipso_doi=empty. Must be specified if IP Label Type is CIPSO. A number corresponding to the host's Domain of Interpretation for CIPSO labeled packets.
audit mask, audit terminal ID, audit session ID	

In a TSIX host's entry, the IP Label Type field can be set to none or CIPSO or RIPSO. IP options in the IP Header are then used to carry CIPSO or RIPSO labels to be used for trusted routing. See "Creating Entries in the Trusted Network Databases" on page 304.

To communicate using the TSIX host type, both the sending and receiving hosts must be within the same tsix DOT (domain of translation).

Note - Sun's definition of the TSIX token maps is the first definition of a TSIX DOT, and other companies who wish to communicate with Trusted Solaris systems need to use the Sun-defined DOT—at least until more than one TSIX DOI is defined. There may be a TSIX DOI registry in the future to define what the tokens mean.

TSIX-type hosts that have the IP Label Field specified as CIPSO *must* have the `tsol_admin_high_to_cipso` switch in the system(4) file set to 1 or the TSIX host cannot communicate with Trusted Solaris hosts. See Chapter 13" for how to add the switch to the `/etc/system` file.

The trusted networking software assigns any missing security attributes on incoming packets from TSIX hosts by getting defaults from the trusted network database.The security attributes that can be specified for TSIX hosts are shown in the following figure. Fields that are not settable for the host type are grayed out and not active.

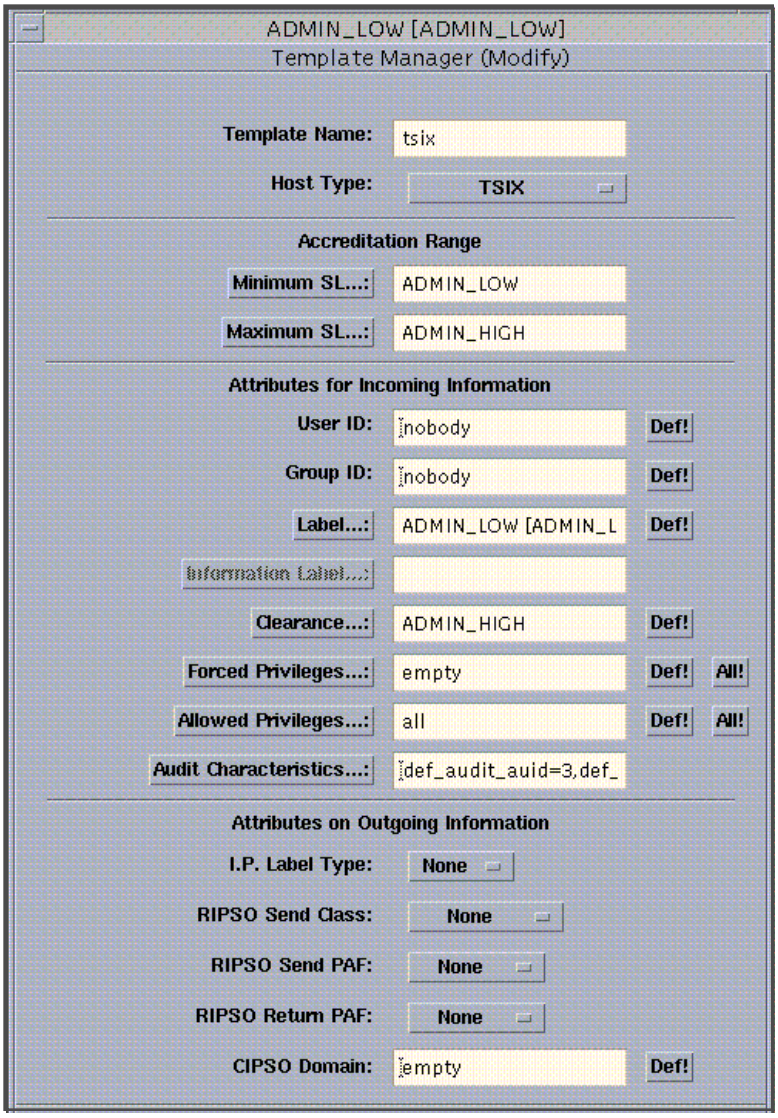


Figure 10–3 Configurable Fields in the Tnrhtp for the TSIX Host Type

TABLE 10–3 Required Security Attributes for tsix Host Types

Configurable Attributes	Notes
minimum SL (min_sl=)	Specify in hexadecimal format ADMIN_LOW or use another valid sensitivity label to restrict communications with the tsix host.
maximum SL (max_sl=)	Specify in hexadecimal format ADMIN_HIGH or use another valid sensitivity label to restrict communications with the tsix host.
allowed privileges (allowed_privs=)	Specifies limits for the effective privilege set propagated across the network (both inbound and outbound) by this host. Only specified privileges are allowed. “all” means that all privileges are interpreted; “none” means that no privileges are interpreted. In the `tnrhtp` template that applies to the host, allowed_privs=empty; means that whatever is specified in the `tnidb` for this field applies. In the `tnidb` file, an empty field means that no privileges are applied.
forced privileges (forced_privs=)	The defined privileges are applied to incoming packets received from the host being defined, because it does not supply privileges. “all” means that all privileges are applied; “none” means that no privileges are applied. In the tnrhtp template that applies to the host, `forced_privs=empty,` means that whatever is specified in the `tnidb` for this field applies. In the `tnidb` file, empty means that no privileges are applied.
default label (def_label=)	The specified CMW label is applied to incoming packets received from the host being defined, because it does not supply a label. If no information label is specified, an information label of ADMIN_LOW is applied.
default clearance	The specified clearance label is applied to incoming packets received from the host being defined, because it does not supply a clearance. A process can write-up to the sensitivity label corresponding to its clearance.
default UID	The specified UID is applied to incoming packets received from the host being defined, because it does not supply a UID.
IP label (ip_label=)	Valid types are: NONE, RIPSO, CIPSO
RIPSO label (ripso_label=)	If IP label=empty, ripso_label=empty. If IP Label Type is RIPSO, one of the supported Classification Level Encodings must be specified: Top_Secret, Secret, Confidential, Unclassified.

TABLE 10–3 Required Security Attributes for tsix Host Types *(continued)*

Configurable Attributes	Notes
RIPSO error (ripso_error=)	If IP label=empty, ripso_error=empty. Must be specified if IP Label Type is RIPSO. Supported Protection Authority Flags: GENSER, SIOP-ESI, SCI, NSA, DOE. The Classification Level is taken from the ripso_label field.
CIPSO DOI (cipso_doi=)	If IP label=empty, cipso_doi=empty. Must be specified if IP Label Type is CIPSO. A number corresponding to the host's Domain of Interpretation for CIPSO labeled packets.

MSIX (`msix`) Host Type

Ethernet Header	IP Header [IP Options]	TCP or UDP Header	SATMP Header	Data	Ethernet Trailer

Labels are the only security attributes supported in communications from MSIX hosts, and they are carried in the IP header.

The MSIX host type should be specified in templates for hosts running the Trusted Solaris 1.x operating environment, which includes MSIX 1.0 plus some features of MSIX 2.0. Because the Trusted Solaris version of MSIX does not support all of the features of MSIX 2.0, it should not be specified in templates for hosts that use the MSIX 2.0 protocol.

The security attributes that can be specified for MSIX hosts are shown in Figure 10–4. Fields that are not settable for the host type are grayed out and not active.

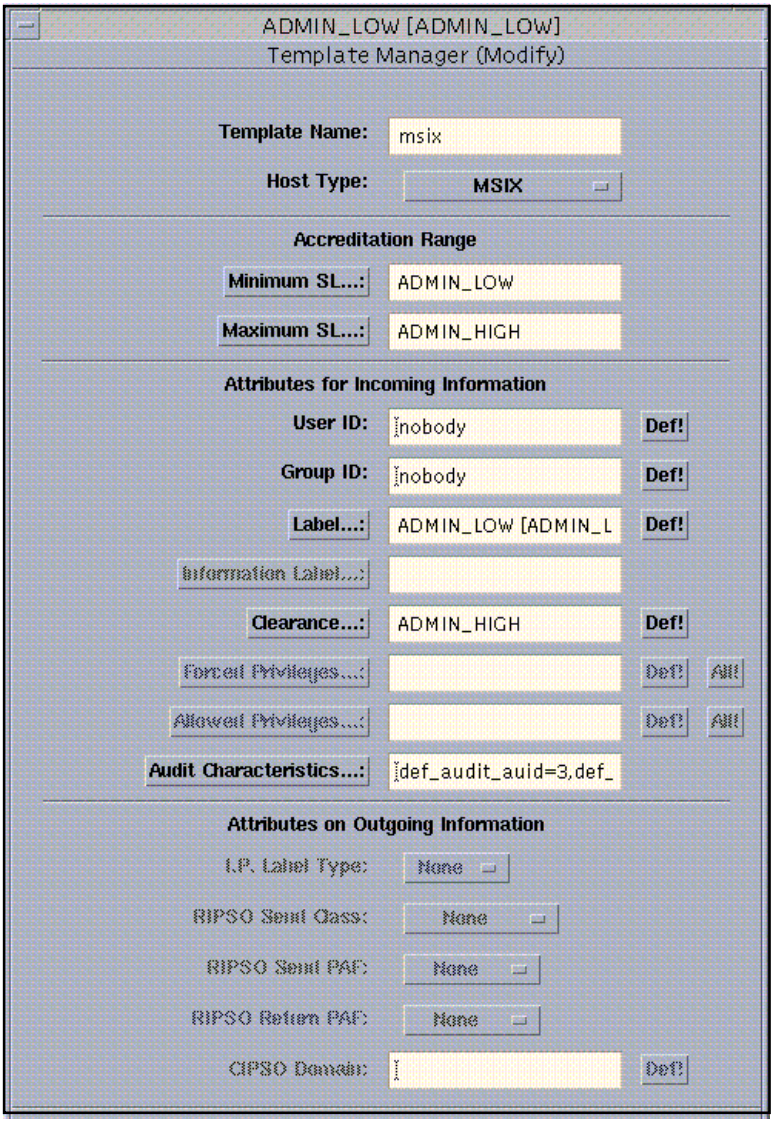


Figure 10–4 Configurable Fields in the Tnrhtp for the MSIX Host Type

CIPSO (`cipso`

) Host Type

Ethernet Header	IP Header [CIPSO Label in IP Options]	TCP or UDP Header	Data	Ethernet Trailer

The CIPSO host type supports communications with hosts that conform to the CIPSO standard. The only security attribute supported for CIPSO hosts is a CIPSO label, which is derived from the sensitivity label of the data and inserted into the IP header of packets going to hosts of host type CIPSO. The host entries must specify the same domain of interpretation (CIPSO DOI) as that of the destination host and of all gateways through which the packets from that host are routed. If the sending host is specified as a CIPSO host type, the trusted networking software looks for a CIPSO label in the IP Header of incoming packets.

The security attributes that can be specified for CIPSO hosts are shown in Figure 10–5. Fields that are not settable for the host type are grayed out and not active.

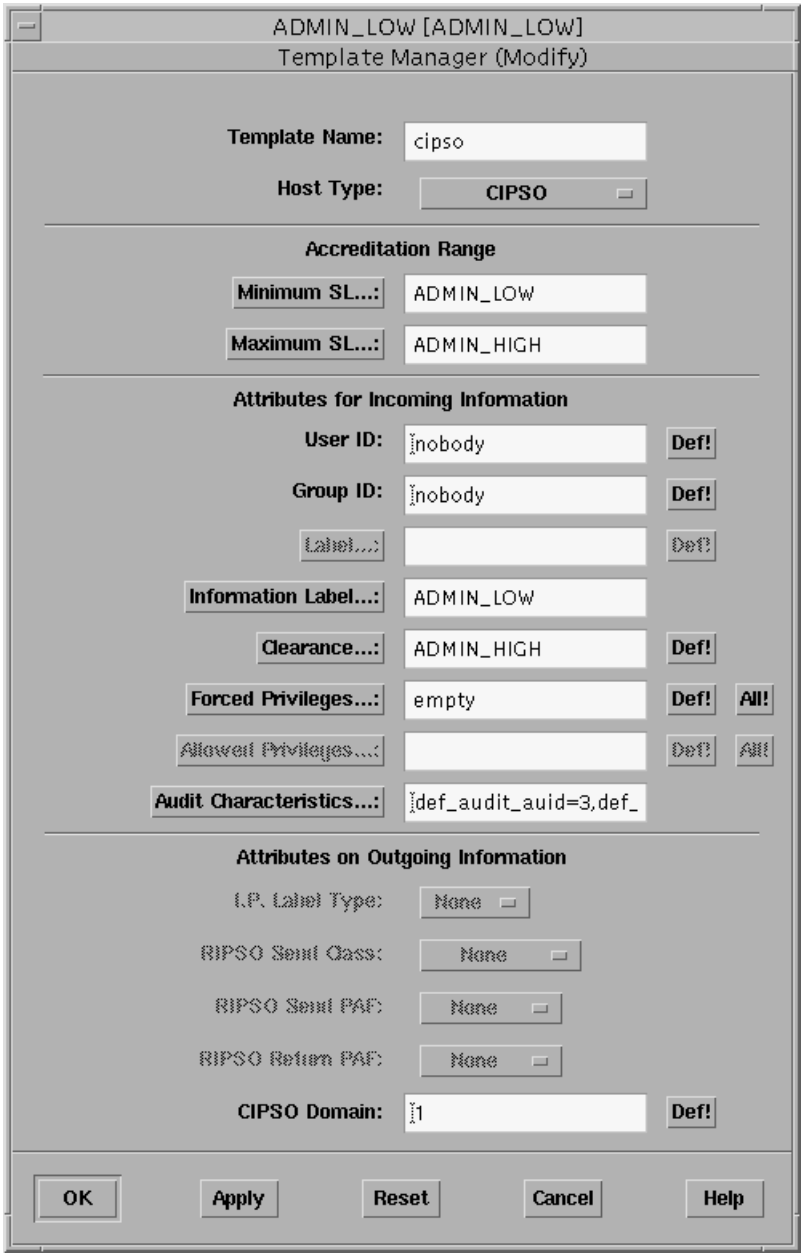


Figure 10–5 Configurable Fields in the Tnrhtp for the CIPSO Host Type

RIPSO (`ripso`

) Host Type

Ethernet Header	IP Header [RIPSO Label in IP Options]	TCP or UDP Header	Data	Ethernet Trailer

Supports hosts that conform to the RIPSO standard. The RIPSO labels that are used for communications with a host of type RIPSO are administratively defined in the template for the host, as specified in the tnrhtp(4TSOL) man page. Templates for hosts of type RIPSO must have the RIPSO Send Class set to one of Top Secret, Secret, Confidential, Unclassified, or a hexadecimal value. Templates must also have both the RIPSO Send PAF and Return PAF set to one of GENSER, SIOP-ESI, CI, NSA, DOE or a hexadecimal value.

The security attributes that can be specified for RIPSO hosts are shown in Figure 10–6. Fields that are not settable for the host type are grayed out and not active.

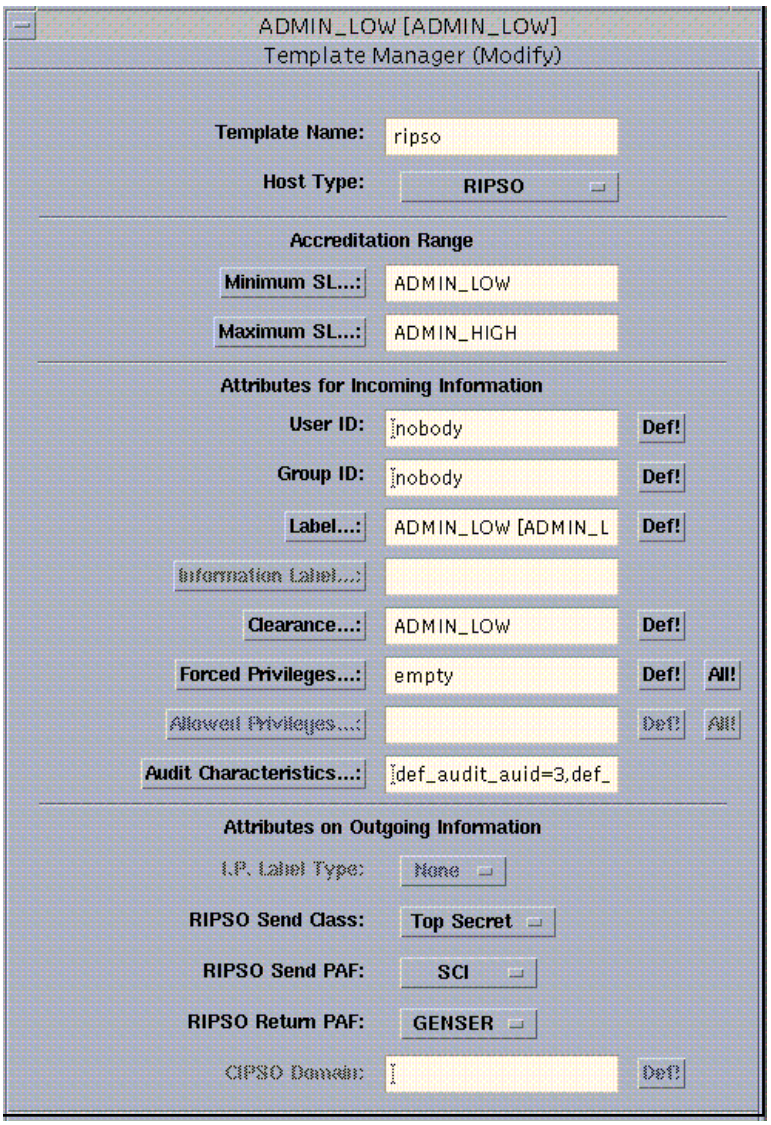


Figure 10–6 Configurable Fields in the Tnrhtp for the RIPSO Host Type

Unlabeled (`unlabeled`

) Host Type

Ethernet Header	IP Header [RIPSO Label in IP Options]	TCP or UDP Header	Data	Ethernet Trailer

The security attributes that can be specified for unlabeled hosts are shown in Figure 10–7. Fields that are not settable for the host type are grayed out and not active.

The minimum sensitivity label and maximum sensitivity label fields are used in trusted routing, to allow packets to be routed through an unlabeled gateway. The security administrator can use the minimum sensitivity label and the maximum sensitivity label to define the range of packets to forward through an unlabeled gateway.

For Unlabeled or RIPSO hosts, outgoing communications take place at all sensitivity labels between the minimum sensitivity label (which is specified in the Minimum SL field) and the default sensitivity label (which is specified in the Label field), while incoming communications can only occur at the default sensitivity label.

Note - Only if your site needs to restrict outgoing communications for an Unlabeled or RIPSO host to a single sensitivity label, would you need set the Minimum SL equal to the default sensitivity label that is specified in the Label field.

The clearance sets the upper limit for write operations performed by anyone on the unlabeled host. For example, a site could set a default sensitivity label of CONFIDENTIAL (C), and a clearance of SECRET (S) for an unlabeled host running the Solaris operating environment. As a result, someone on the Solaris host that was working in a file system mounted from a Trusted Solaris host could open an upgraded file with a sensitivity label of S and write into it (as long as the file name was visible to the user on the Solaris host).

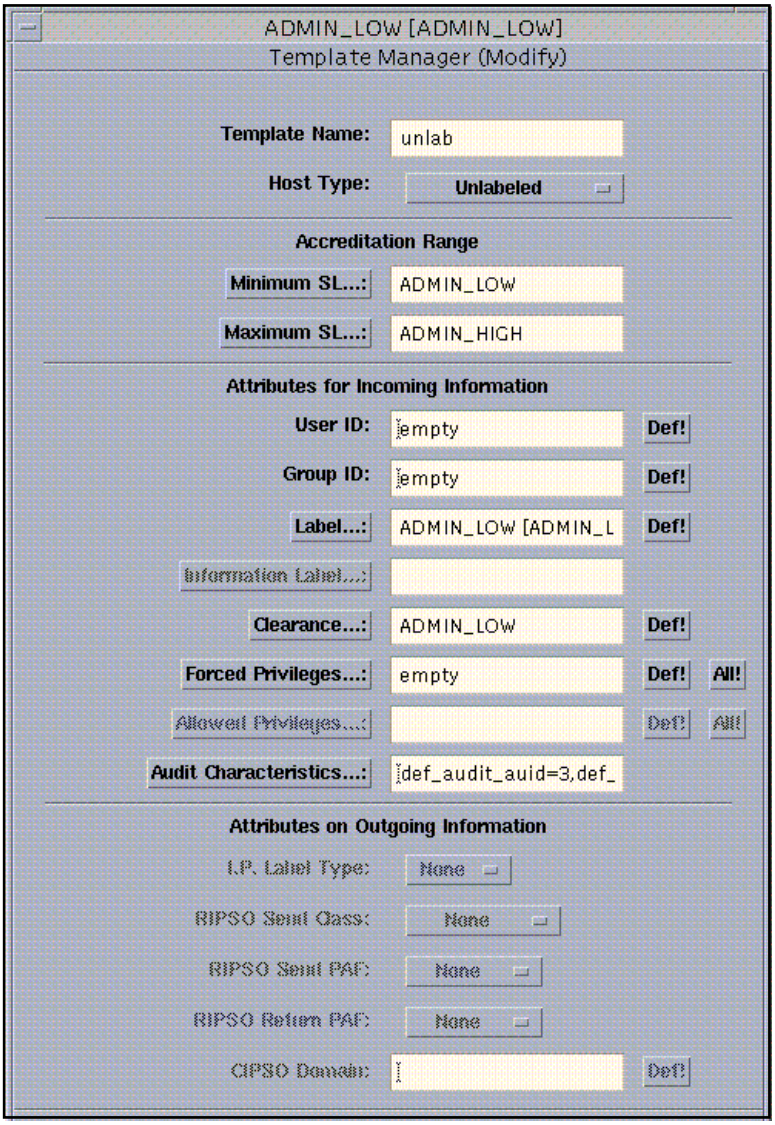


Figure 10–7 Configurable Fields in the Tnrhtp for the unlabeled Host Type

Creating Entries in the Trusted Network Databases

Before the security administrator can set up the entries in the `tnrhdb`, in which templates are assigned to hosts and networks to templates, he or she should do the following:

- Review the existing templates from the tnrhtp(4TSOL) file

  Use the Database Manager to bring up the Tnrhtp database, select each template in turn and view its contents in Template Manager modify dialog box.
- Modify existing templates or create any new templates needed for the site
- Decide which templates should be used for each host and network

Using `tnrhdb` Options to Achieve a Closed or Open Type of Network Configuration

Trusted networking supports both the following types of network configurations:

- An *open* configuration permitting *communication with any host*
- A *closely-controlled* configuration permitting communications only with specified hosts and networks

Each site chooses whether to be inclusive or exclusive in allowing communications with other hosts and networks.

Hierarchical Fallback Mechanism

A hierarchical fallback algorithm is used by the trusted networking software in looking for a host's entry in the tnrhdb(4TSOL) database, as described in the man page. The trusted networking software first looks for an entry specific to the host, and if it does not find one, it falls back to searching for a matching class C network entry, then a class B entry, a class A entry, and finally a wildcard entry (IP address 0.0.0.0), if one exists. If no entry in the `tnrhdb` database matches the IP address of the host, communications are not allowed with the host. Both open and controlled configurations can make selective use of the fallback mechanism.

Open Configuration Using a Wildcard

The security administrator can make a *wildcard entry* to specify a default set of attributes that will be assigned to any host for which an entry does not exist in the

trusted network databases, to allow communications with every host or network. The template assigned to the wildcard entry is configured by the security administrator in the tnrhtp(4TSOL) database, along with the other templates, as described under “Setting Up Templates” on page 305.

Closely-controlled Configuration

Communication with the host is not permitted if the trusted network software cannot resolve a host’s IP address to an entry in the `tnrhdb` database. To strictly restrict communications, the security administrator can simply not use a wildcard and be very careful about using network entries.

Setting Up Templates

See the following procedures, which show some of the different ways the security administrator may associate a host with a template.

- “To Assign a Template to a Single Host in the `tnrhdb` ” on page 318
- “To Assign a Template to a Group of Hosts in the `tnrhdb`” on page 320
- “To Create a Wildcard Entry for All Hosts Not Otherwise Specified” on page 322

Note - The examples assume that the *tsol_1*, *tsol*, and *wildcard* templates are specified in the `tnrhtp` file.

Precedence Rules for Attributes in Trusted Network Databases

If the security administrator supplies any value in the `tnrhtp` template for a host, the value in the `tnrhtp` file takes precedence over the values specified for the network interface in the `tnidb`, as shown in Figure 10–8. As shown in Figure 10–8, the values in the `tnidb` apply only when values have not been set in the `tnrhdb` entry for the host sending the packets.

Required fields must be specified in either one or both files. If there is a default field with its value specified as `empty` in the tnrhtp, the value *must* be supplied in the corresponding default field in the `tnidb` file.

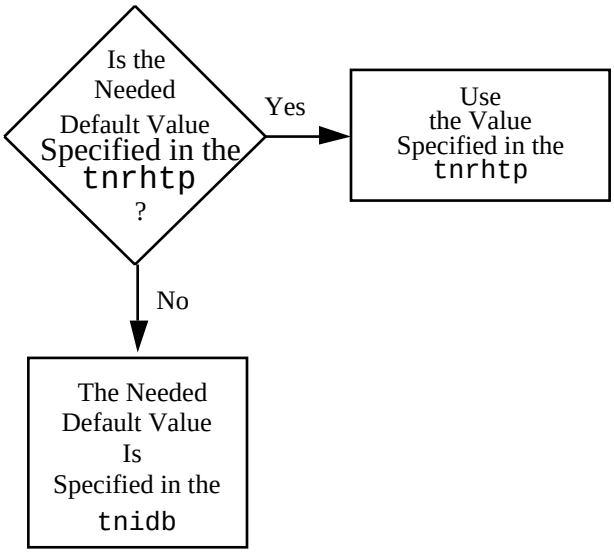


Figure 10–8 Attribute Precedence Rules

When adding a new interface, the security administrator must specify all of the fields in the Database Manager Interface Manager (Add) dialog box. If the security administrator does not make any changes to default definitions in the `tnidb` for the default set of interfaces, the default values shown in the Database Manager Interface Manager apply. Figure 10–9 shows a `tnidb` entry for le0, where the Minimum SL is ADMIN_LOW, the Maximum SL is ADMIN_HIGH, the Label is ADMIN_LOW[ADMIN_LOW], the Clearance is ADMIN_HIGH, the User ID and Group ID are nobody.

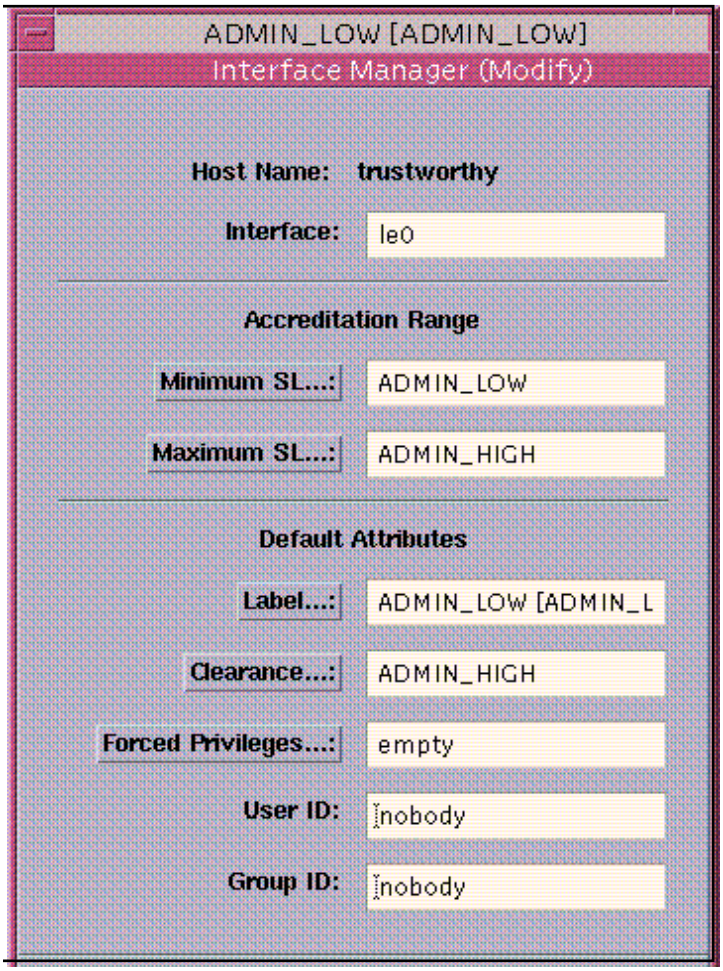


Figure 10–9 Default Entry for the le0 Interface in the Tnidb Database

Precedence Example

In the following figure, a template named wildcard is being defined for unknown hosts.The Label is specified as INTERNAL_USE_ONLY[INTERNAL], the Clearance is INTERNAL, the User ID is 12022 and Group ID is 10.

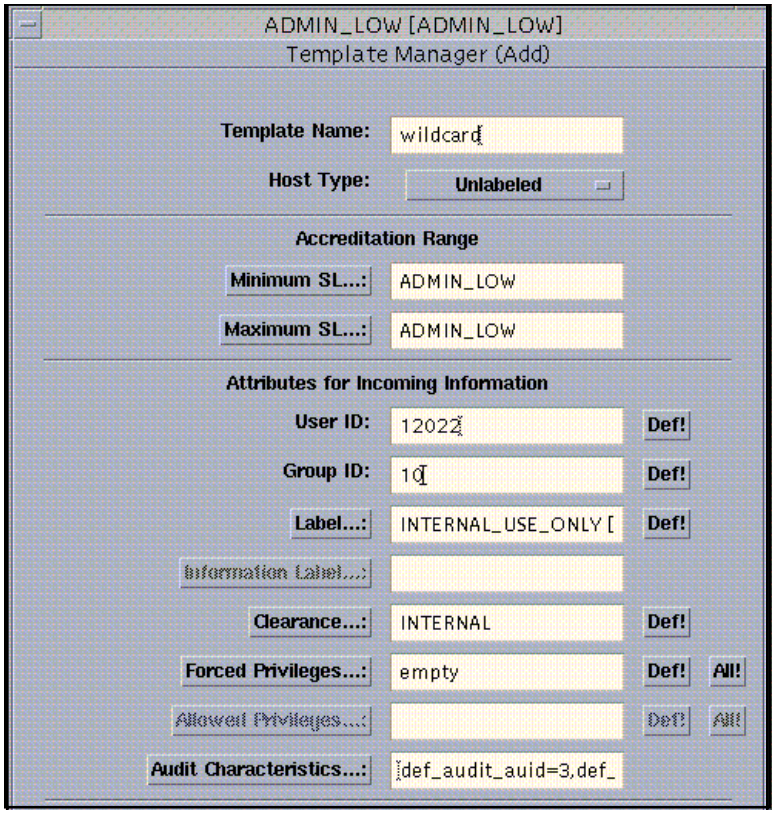


Figure 10–10 Assigning Default Attributes to Communications from Unspecified Hosts

For this example, the definitions for the interface are the defaults specified in Figure 10–9, with a Label of ADMIN_LOW[ADMIN_LOW], the Clearance as ADMIN_HIGH, the User ID and Group ID as nobody.

Because the Label, clearance, UID and GID are taken from the wildcard template, which takes precedence over the values in the `tnidb` for the interface, packets coming from an unknown host through the le0 interface are assigned a Label of INTERNAL_USE_ONLY[INTERNAL], a Clearance of INTERNAL, a User ID of 120022, and a Group ID of 10.

Network Accreditation Range

Each host and network has an accreditation range. The accreditation range is set by a min_sl and a max_sl entry in the `tnrhtp` file for a host and in the `tnidb` file for a network interface.

The example in Figure 10–11 shows host G with two network interfaces, le0 and le1. le0 is defined with an accreditation range of ADMIN_LOW to ADMIN_HIGH, while le1 is defined with an accreditation range of INTERNAL_USE_ONLY to NEED_TO_KNOW ENG.

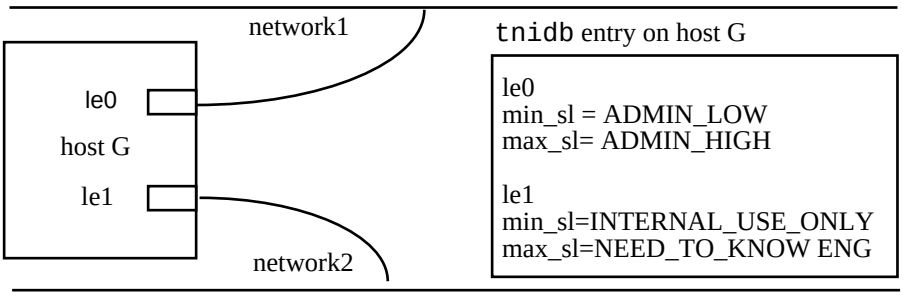


Figure 10–11 Two Network Interfaces and their Network Accreditation Ranges

Note - Communications within the Trusted Solaris distributed system require an accreditation range of ADMIN_LOW to ADMIN_HIGH.

See “To Configure a Network Interface ” on page 326.

Special Boot-time Trusted Network Databases

Special boot-time-only local versions of the `tnrhdb`/`tnrhtp` files reside in the `/etc/security/tsol/boot` directory. In the default Trusted Solaris environment,

these boot-time trusted network databases are relied upon to allow a newly installed but not yet configured NIS+ client host to contact the NIS+ master during boot. The two boot-time trusted network files are needed because neither the NIS+ tables or local files that contain the trusted network database entries are available during boot, and the trusted network daemon, tnd(1MTSOL) is not yet running.

After first boot of a NIS+ client, the install team replaces the wildcard entry in the `boot/tnrhdb` with the address of the NIS+ master. During subsequent boots, only the specific hosts listed in the boot-time databases are contacted.

After boot, when the NIS+ master is contacted, either the NIS+ tables or the local versions of the `tnrhdb`/`tnrhtp` files in the `/etc/security/tsol` directory or both are consulted by `tnd` when it comes up [depending on the setting in `nsswitch.conf`(4)]. Any duplicate entries from the boot time trusted network files are overwritten in the trusted network cache.

The default `tnrhdb` and `tnrhtp` entries in the `boot` directory define a wildcard entry that in effect allows communications with *any* host, treating it as a Trusted Solaris 2.x host (by assigning it the tsol template that defines it as a sun_tsol host type). Unless modified, the default entries in the boot versions of the trusted network databases allow communications with any Trusted Solaris 2.x hosts, and this state of affairs may not be consistent with your site's security policy.

Therefore, the default entries may conflict with your site's security policy for the following reasons.

- If you do not define a wildcard entry in your system's standard `tnrhdb` location (either in the `/etc/security/tsol/tnrhdb` file or in the `trnhdb` NIS+ table), and if you do not change the wildcard entry in the `boot/tnrhdb` to refer to a specific host address, you could be allowing more openness than you intended in your network communications.
- The wildcard entry from the boot time `tnrhdb` is not overridden unless your site defines a wildcard entry in the standard `tnrhdb`.

Administering the Boot-time Trusted Network Databases

The boot-time `tnrhdb` file in `/etc/security/tsol/boot` contains the wildcard entry `0.0.0.0:tsol`. The host type is defined as `sun_tsol` in the `tsol` template in the `/etc/security/tsol/boot/tnrhtp` file.

Caution - Sites that specifically define each host and network with which communications are allowed will want to change the default wildcard entry.

When no wildcard entry is desired, the entry `0.0.0.0:tsol` in `/etc/security/tsol/boot/tnrhdb` should be changed to the IP address of the NIS+ master. Also, if the host type of the NIS+ master is not sun_tsol, another template should be added to the `/etc/security/tsol/boot/tnrhtp` file for the desired host type, and the template's name should be specified in the entry in the `tnrhdb` file. These changes are usually made when a host is being configured.

For example, if the NIS+ master has a tsix host type, then the default tsix template should be copied from the `/etc/security/tsol/tnrhtp` to the `/etc/security/tsol/boot/tnrhtp`. The entry in the `/etc/security/tsol/boot/tnrhdb` would then need to be modified to replace tsol with tsix, like this: `129.150.119.20:tsix`. See "Templates Assigned to Host Types in the Template Manager" on page 289, if needed.

The security administrator can add other IP addresses and assign templates to these addresses in the `/etc/security/tsol/boot/tnrhdb` file, if a NIS+ client host needs to communicate with remote hosts or networks before the `tnd` is running.

These files are not needed on NIS+ masters and standalone systems that rely on local configuration files; on such systems, the files probably should be removed.

These files should be edited by root at ADMIN_LOW using the Admin Editor action. See "To Change the Default Entry in the Boot-time `tnrhdb/tnrhtp` Files" on page 312 for the procedure.

Setting Up Tunneling

As described under "Tunneling" on page 267, tunneling allows the sharing of emetrics for routes even when there is a non-Trusted 2.5.1 cloud of hosts and gateways between two Trusted Solaris 2.5.1 gateways. All hosts must be in the same intranet with gateways using RIP. Without tunneling, the special security response packets generated by Trusted Solaris extended RIP on one gateway cannot get to the remote Trusted Solaris 2.5.1 gateway to pass along the emetrics of its known routes.

To set up tunneling, the security administrator creates a `tunnel` file on a Trusted Solaris 2.5.1 gateway. The tunnel file contains the IP addresses of remote networks connected to Trusted Solaris 2.5.1 gateways. Unlabeled broadcast packets containing security information are sent directly to the networks listed in the `tunnel` file, where they are received by Trusted Solaris 2.5.1 gateways. See "To Set Up Tunneling" on page 341.

Procedures

▼ To Change the Default Entry in the Boot-time `tnrhdb/tnrhtp` Files

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

   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Use the Admin Editor action to open the `/etc/security/tsol/boot/tnrhdb` file for editing.**

   See “To Use the Admin Editor Action to Edit a File” on page 29, if needed.

3. **Substitute the IP address of the NIS+ master for the wildcard entry, if desired.**

   The following example shows the default entry.

```
0.0.0.0:tsol
```

   Replace 0.0.0.0 with the IP address of the NIS+ master. The example below shows an entry for a NIS+ master whose IP address is 129.96.22.40.

```
129.96.22.40:tsol
```

4. **Change the name of the template, if needed.**

   For example, if the NIS+ master has a host type of TSIX, replace tsol with the name of the tsix template, as shown below.

```
129.96.22.40:tsix
```

5. **Add entries for any other server that the system needs to communicate with during boot, supplying the IP address and the appropriate template that applies to the host type.**

```
129.96.22.40:tsix
   129.96.22.488:tsol
```

6. **Write and quit the file.**

```
:wq
```

7. **Use the Admin Editor action to open the `/etc/security/tsol/boot/tnrhtp` file for editing, and add additional templates that are needed for the entries in the `tnrhdb` file, if desired.**

8. **Write and quit the file.**

▼ To Access the Trusted Network Databases from the Database Manager

1. **Assume the root role and go to an ADMIN_LOW workspace.**
   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Open the Solstice_Apps folder in the Application Manager.**
   See “To Launch Solstice Administration Tools” on page 27, if needed.

3. **Double click on the Database Manager icon to bring up the Database Manager Load dialog box.**
   Figure 10–12 shows the Database Manager selected in the Solstice_Apps Folder.

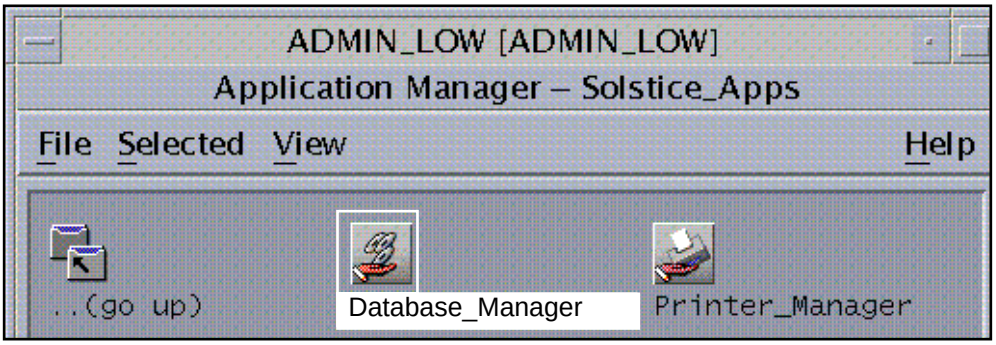


Figure 10–12 Database Manager Selected in the Solstice_Apps Folder

4. **Select a naming service from the Naming Service menu on the Database Manager Load dialog box**

Note - By default, as shown in Figure 10–13, the NIS+ naming service is selected, and the name of the NIS+ domain displays in the Domain text entry field.

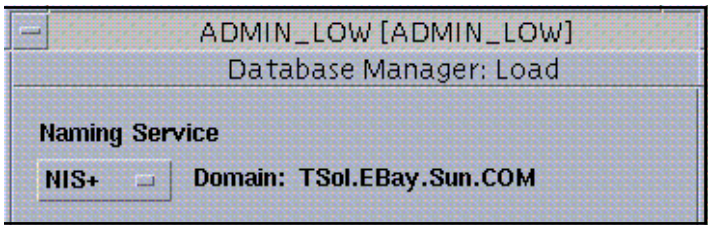


Figure 10–13 Loading a Naming Service in the Database Manager

---

Note - Remember that when using NIS+, `tnidb` is always a local file, while `tnrhdb` and `tnrhtp` are configured using the NIS+ naming service. When configuring a standalone Trusted Solaris system with no naming service, all three files are local files that must be modified with Naming Service set to None.

---

a. **For the `tnidb`, choose None.**

   Choosing None displays the name of the current host in the Host text entry field, as shown in Figure 10–14.

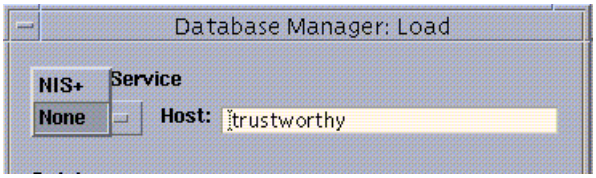


Figure 10–14 No Naming Service Selected in the Database Manager

b. **For the `tnrhdb` and `tnrhtp` files, choose whichever naming service applies, either NIS+ or None.**

5. **If the naming service is None, to configure a database on a remote host, enter the name of a host in the Host text entry field.**

6. **Double click the name of the trusted networking database to be edited or highlight the name and click OK.**

   See Figure 10–15.

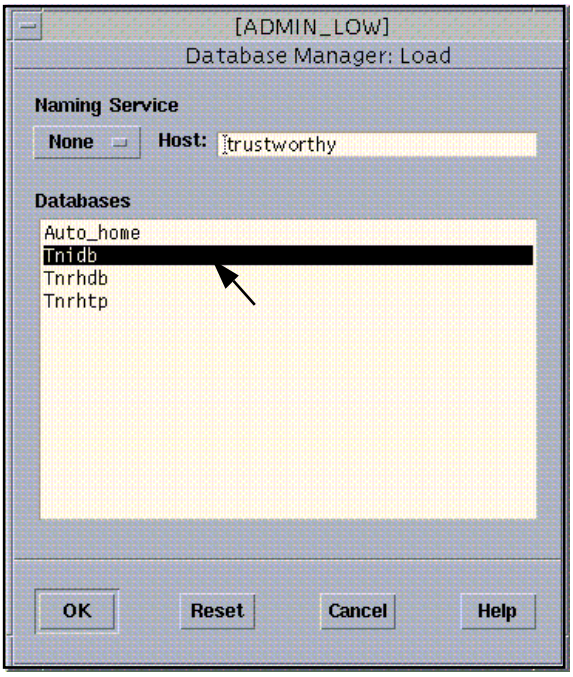


Figure 10–15 Tnidb Selected in the Database Manager: Load List

▼ To Create a New Template in the `tnrhtp`

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, and access the Database Manager.**

   See “To Access the Trusted Network Databases from the Database Manager” on page 313, if needed. See also the tnrhtp(4TSOL) man page for the definitions of the fields. As described in “Security Attributes Configurable for Each Host Type” on page 288, not all fields are on each template.

2. **Click on the Naming Service options menu and click on None.**

3. **Highlight Tnrhtp from the Database Manager Load list and choose Add from the Edit menu.**

   The Template Manager (Add) dialog box displays as shown in Figure 10–16.

Note - Several fields in the Template Manager dialog box have a default button (marked Def!) to their right. If the security administrator clicks the default button, a default value is supplied in the field.

---

4. **Type in the Template Name.**

5. **Click on the Host Type options menu button and click to select the desired host type from the menu.**

6. **Click the Minimum SL button to use the label builder to specify the desired minimum sensitivity label.**

   For hosts that are labels-cognizant, this field is used to set the lower bound of the host's accreditation range. For unlabeled and RIPSO host types, this field can be used for trusted routing when the host is a gateway.

7. **Click the Maximum SL button to use the label builder to specify the desired maximum sensitivity label.**

   For hosts that are labels-cognizant, this field is used to set the upper bound of the host's accreditation range. For unlabeled and RIPSO host types, this field can be used for trusted routing when the host is a gateway.

8. **Type in the desired UID, or click Def! to apply the default of empty.**

   For unlabeled hosts, if no UID is specified here, a UID must be specified in each local host's Tnidb.

9. **Type in the desired GID, or click Def! to apply the default of empty.**

   For unlabeled hosts, if no GID is specified here, a GID must be specified in each local host's Tnidb.

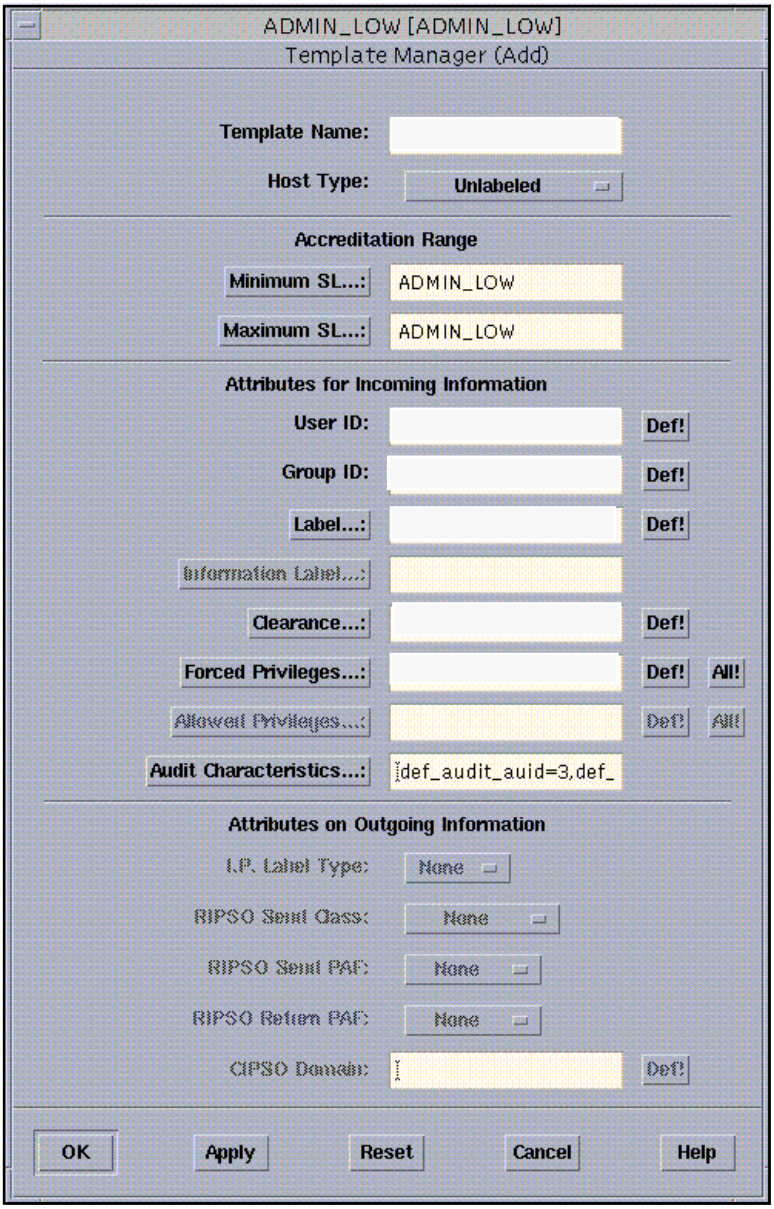


Figure 10–16 An Empty Template in the Template Manager (Add) Dialog Box

10. Click the Label button to use the label builder to specify the desired default CMW label (information label and sensitivity label).

11. **Click the Clearance button to use the clearance builder to specify the desired default clearance label.**

12. **Click the Forced Privileges button to bring up the Forced Privileges dialog box to move the desired forced privileges to the Included: list, click the Def! button to specify the default forced privilege set (empty) or click the All! button to specify all privileges.**

13. **Click the Allowed Privileges button to bring up the Allowed Privileges dialog box to move the desired allowed privileges to the Included: list, click the Def! button to specify the default allowed privilege set (empty), or click the All! button to specify all privileges.**

14. **Click the Audit Characteristics button to bring up the Audit dialog box and specify the Audit User ID, Audit Mask, Audit Terminal ID, and Audit Session ID.**

15. **Specify the IP Label fields for trusted routing.**

    See "Setting Up Trusted Routing " on page 279 in Chapter 9."

16. **When you are done making changes, click OK.**

▼ To Assign a Template to a Single Host in the tnrhdb

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, and access the Database Manager.**

   See "To Access the Trusted Network Databases from the Database Manager" on page 313, if needed.

2. **Highlight Tnrhdb from the Database Manager Load list and choose Add from the Edit menu.**

3. **Enter the IP address of the host into the IP Address: field.**

4. **Click on the Template Name: options menu button, and click to select the desired template name, as shown in the following figure.**

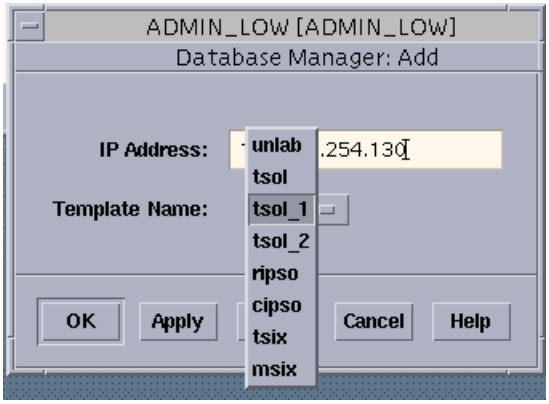


Figure 10–17 Selecting a Template from the Template Name Menu

5. When you are done making changes, click the OK button.

The following figure shows the host identified by its IP address (192.110.120.6) and assigned to the template tsol_1.

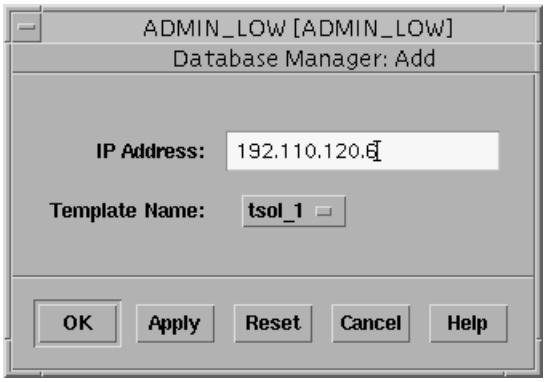


Figure 10–18 Adding a Host Entry to Tnrhdb and Specifying a Template

The following figure shows the new entry in the `tnrhdb` database.

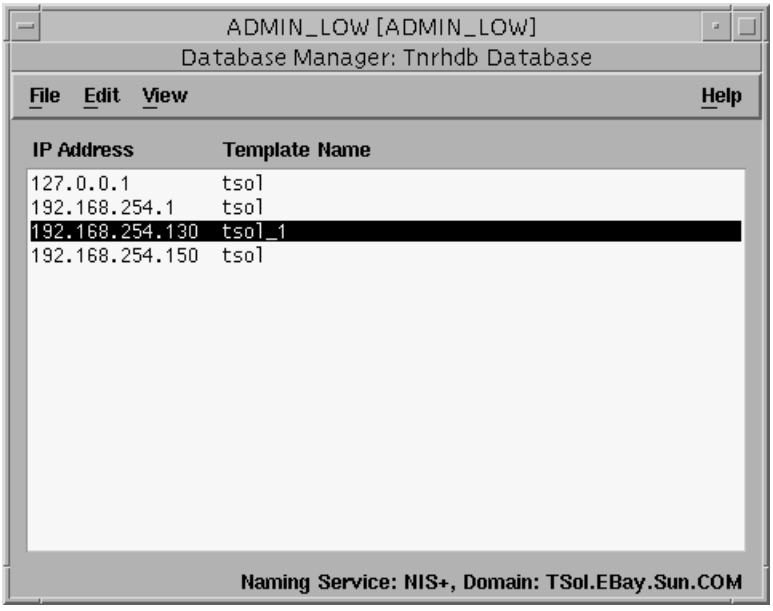


Figure 10–19 Tnrhdb Host Entry Assigned to the tsol_1Template

▼ To Assign a Template to a Group of Hosts in the `tnrhdb`

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, and access the Database Manager.**

   See "To Access the Trusted Network Databases from the Database Manager" on page 313, if needed.

2. **Decide which of the templates you want to assign out of the `Tnrhtp` database.**

3. **Highlight `Tnrhdb` from the Database Manager Load list and choose Add from the Edit menu.**

4. **Enter the IP address of the network.**

5. **Click on the Template Name: options menu button, and click to select the desired template name.**

6. **When you are done making changes, click the OK button.**

The following figure shows the network entry.

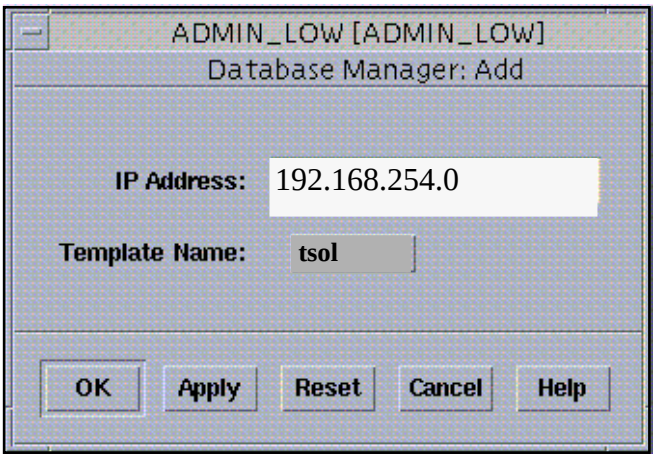


Figure 10–20 Adding a Network Entry to Tnrhdb and Specifying a Template

Figure 10–21 shows a new entry for the network 192.110.120.0, which has the tsol template and an entry for the host whose IP address in 192.110.120.6, which has the tsol_1 template. The host whose IP address in 192.110.120.6 has the tsol_1 template while all others in that network have the tsol template.

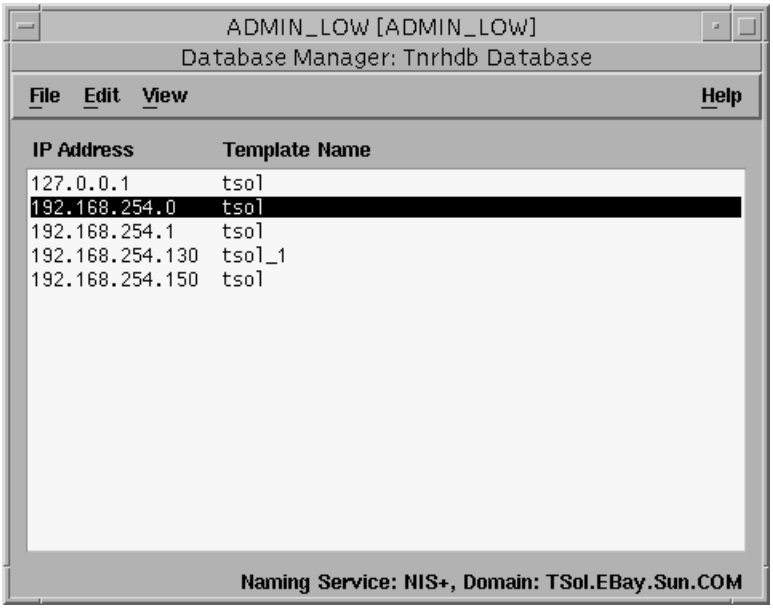


Figure 10–21 Tnrhdb Network Entry Assigned to the Template Named tsol

---

Note - Listing every host—either explicitly or implicitly (by making an entry for the network to which the host is connected)—creates a controlled configuration where only the listed hosts are allowed to communicate with the system.

---

7. **When you are done making changes, click on File to display the menu, and click the Exit option to close the Database Manager: Tnrhdb Database dialog box.**

▼ To Create a Wildcard Entry for All Hosts Not Otherwise Specified

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, and access the Database Manager.**

   See “To Access the Trusted Network Databases from the Database Manager” on page 313, if needed.

2. **Create a new template, if needed.**

See “To Create a New Template in the `tnrhtp` ” on page 315. The following figure shows a wildcard template that specifies the Unlabeled host type.

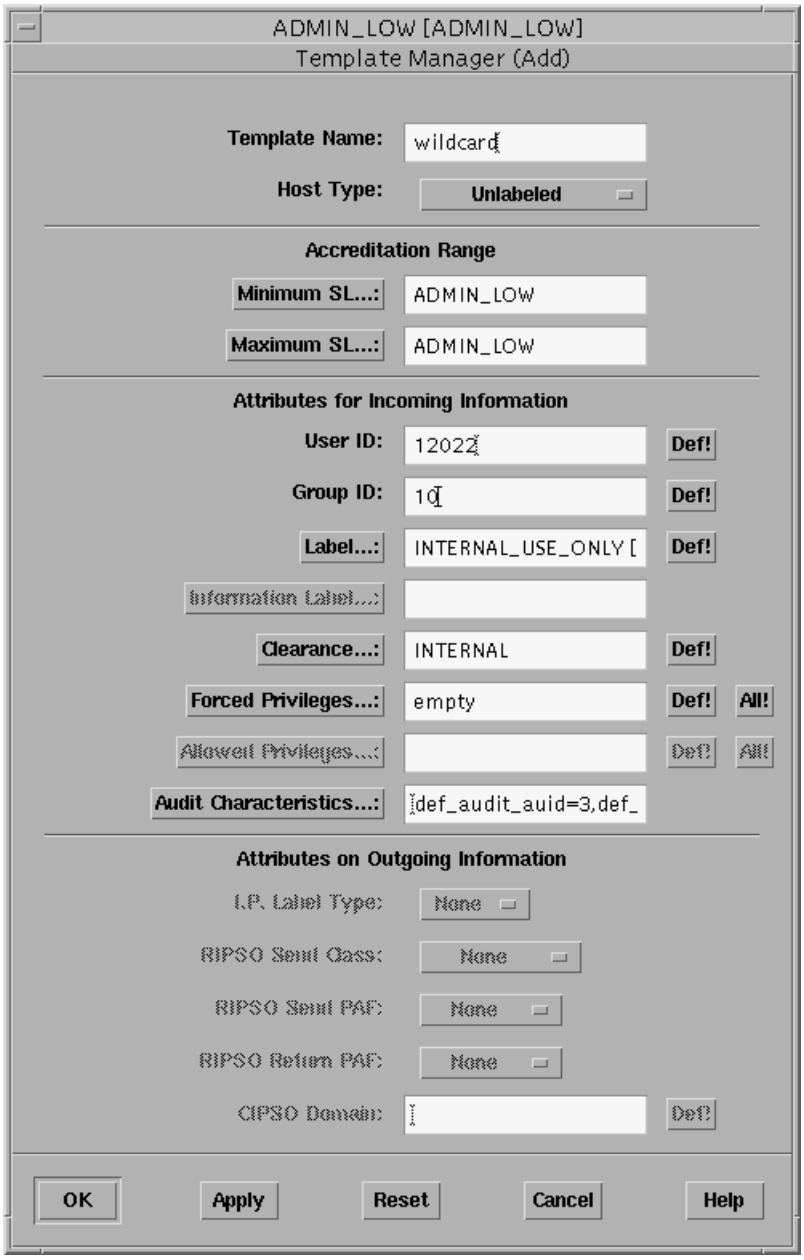


Figure 10–22 A New wildcard Template in the Tnrhtp Database Manager

3. **Highlight `Tnrhdb` from the Database Manager Load list, and choose Add from the Edit menu.**

   See "To Access the Trusted Network Databases from the Database Manager" on page 313, if needed.

4. **Enter the wildcard IP address of 0.0.0.0 and the Template name in the dialog box that displays.**

   The following figure assigns the wildcard template to the `0.0.0.0` wildcard entry.

Figure 10–23 IP Address and Template Name for a Tnrhdb Fallback Entry

5. **When you are done making changes, click the OK button.**

   Figure 10–24 shows a new entry for the wildcard IP address, associating it with a template called wildcard. The new wildcard entry assigns all unspecified hosts to the default entry.

Caution - Using a wildcard entry creates a wide open configuration where *any host* is allowed to communicate with the system.

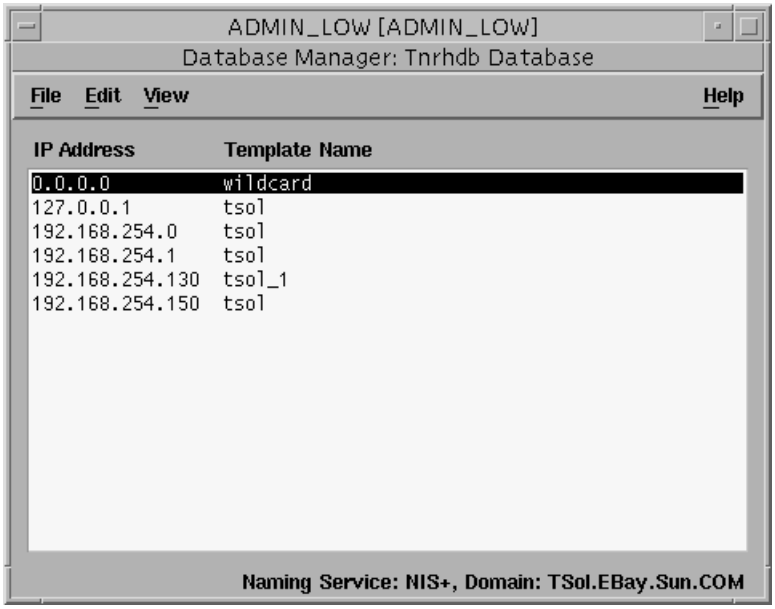


Figure 10–24 Tnrhdb Fallback Template Entry

6. **When you are done making changes, click on File to display the menu, and click the Exit option to close the Database Manager: Tnrhdb Database dialog box.**

▼ To Set an Accreditation Range in a Host Template or Network Interface Entry

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, and access the Database Manager.**

   See "To Access the Trusted Network Databases from the Database Manager" on page 313, if needed.

Note - A separate `tnidb` file resides on each host to specify attributes for that host's own network interfaces, so make sure to choose None in the Naming Service menu in the Database Manager load list when modifying the `tnidb`.

---

2. **To set a host accreditation range in a template that is assigned to one or more hosts, modify Tnrhtp and specify a Minimum SL and a Maximum SL.**

a. **Highlight `Tnrhtp` in the Database Manager Load list.**

b. **To change an existing template, highlight the name of the template and choose Modify.**

c. **Specify the desired sensitivity labels in the Minimum SL and Maximum SL fields, using the label builder dialog boxes that display when you click on the buttons.**

3. **To set a network accreditation range for one or more interfaces on the current host.**

   a. **Highlight `Tnidb` in the Database Manager Load list.**

   b. **To change an existing template, highlight the name of the template whose value you want to set and choose Modify.**

   c. **Enter the desired sensitivity label in the min_SL and max_SL fields.**

   Note - The default network accreditation range is from `ADMIN_LOW` to `ADMIN_HIGH`.

▼ To Configure a Network Interface

1. **If needed, before starting, add the physical network interface to the host, following the hardware and software installation steps in the manuals shipped with the interface.**

2. **If there is more than one network interface on the host, 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 the default settings for any interfaces on the host be changed, change the entries as described in "To Add a New Entry or Modify an Existing Entry in tnidb(4TSOL)" on page 328."**

   The default settings are shown inTable 10–4. If the defaults are acceptable, no changes to the tnidb(4TSOL) file are needed.

TABLE 10–4 Default tnidb Settings

Minimum SL	Maximum SL	Label (default)	Clearance	User ID	Group ID	Forced Privileges
ADMIN_LOW	ADMIN_HIGH	ADMIN_LOW[ADMIN_HIGH]	ADMIN_HIGH	nobody	nobody	empty

4. **If the name of the new network interface is not already in the `Tnidb` database, add it as described under "To Add a New Entry or Modify an Existing Entry in tnidb(4TSOL)" on page 328."**

   Figure 10–25 shows the names of the default network interfaces.

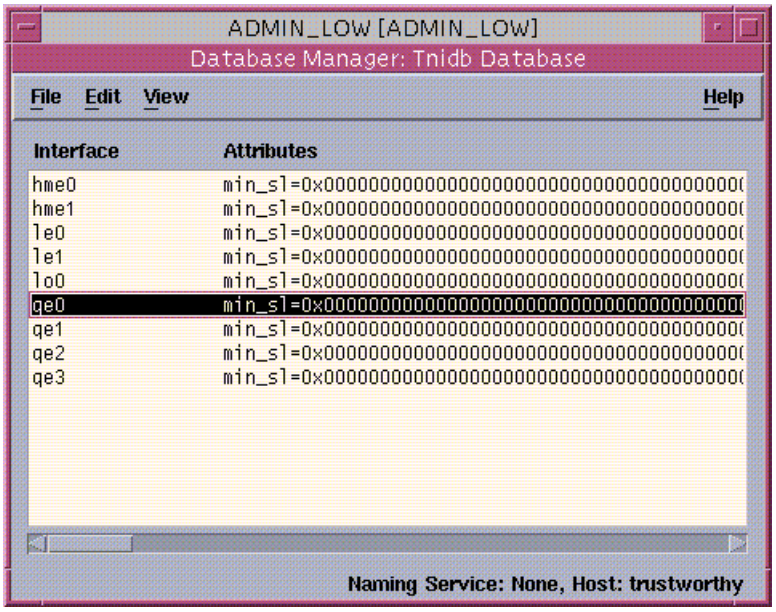


Figure 10–25 Default Interfaces Listed in the Tnidb Database

▼ To Add a New Entry or Modify an Existing Entry in tnidb(4TSOL)

1. **Assume the security administrator role, go to an ADMIN_LOW workspace, and access the Database Manager.**

   See “To Access the Trusted Network Databases from the Database Manager” on page 313 if needed.

2. **If needed, configure the interface.**

   See “To Configure a Network Interface ” on page 326, if needed.

3. **If the name of the interface is not in the default list, add it, and specify the attributes desired.**

   a. **Choose Add from the Edit menu.**

   Figure 10–30 shows the le0 interface name and the Add option highlighted on the Edit menu.

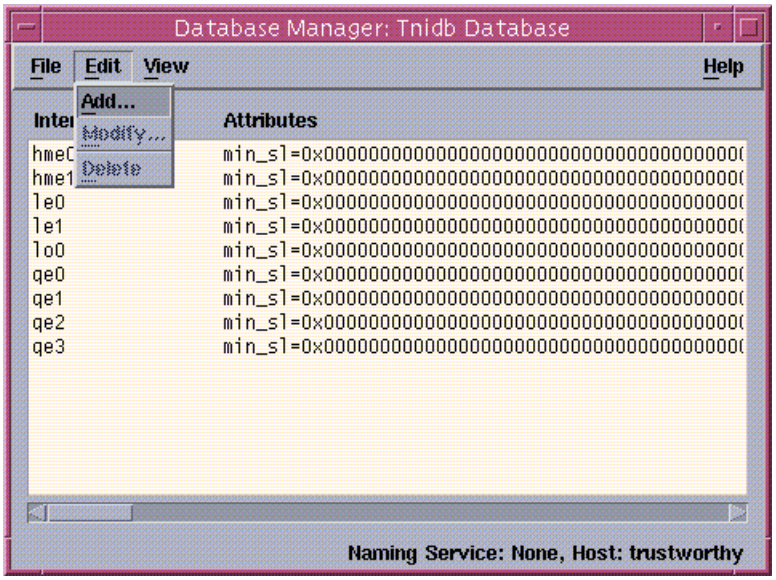


Figure 10–26 Add Option Selected from the Tnidb Edit Menu

   b. **Specify the name of the added interface and its desired attributes.**

   Figure 10–31 shows the Interface Manager (Add) Dialog Box.

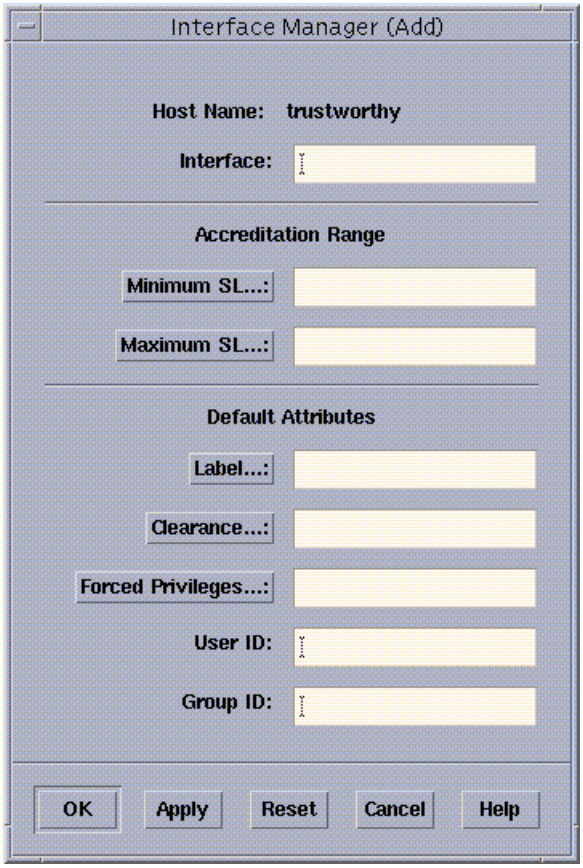


Figure 10–27 Interface Manager (Add) Dialog Box

4. If the name of the interface is in the default list, modify the default settings.

a. Highlight the name of the interface and select Modify from the Edit menu.

Figure 10–28 shows the le0 interface name and the Modify option highlighted on the Edit menu.

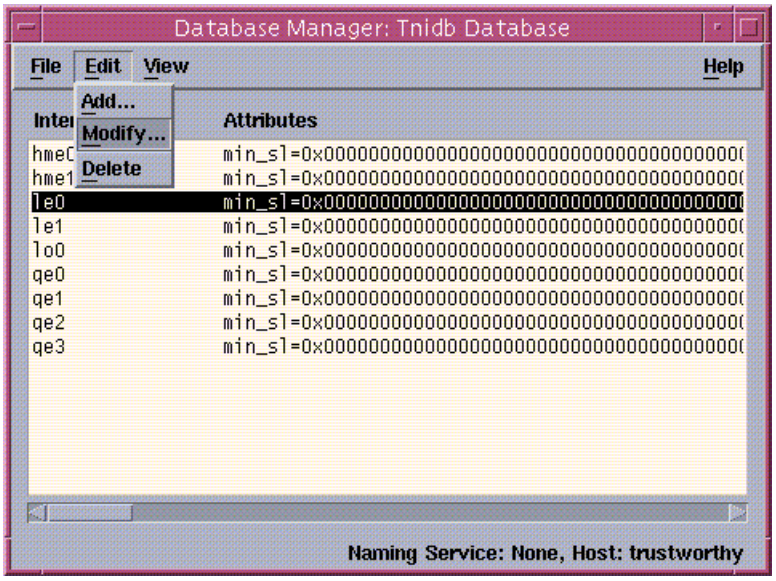


Figure 10–28 Tnidb Interface le0 Highlighted and the Edit > Modify Option Selected

b. Modify the settings as desired.

Figure 10–29 shows the default settings for hme0.

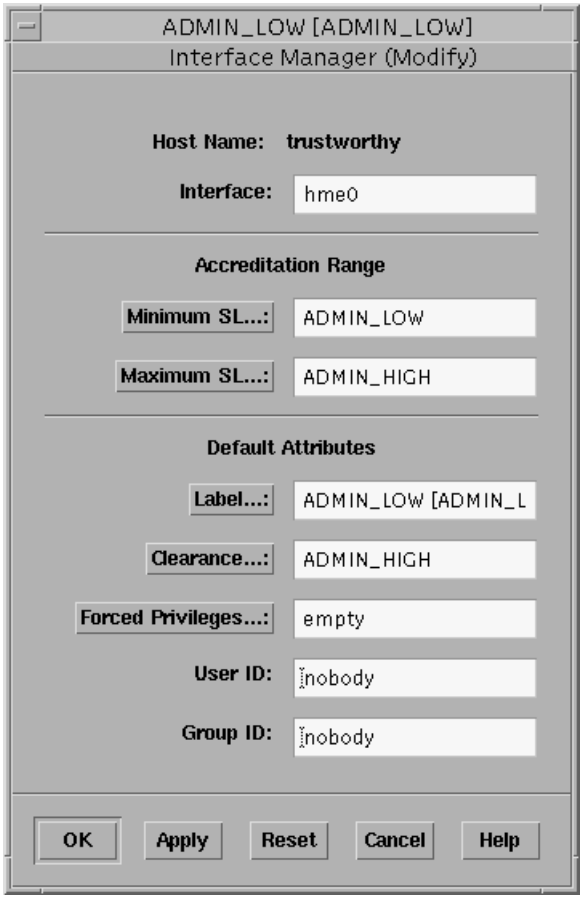


Figure 10–29 Interface Manager (Modify) Dialog Box

5. If the name of the interface is not in the default list, add it, and specify the attributes desired.

a. Choose Add from the Edit menu.

Figure 10–30 shows the le0 interface name and the Add option highlighted on the Edit menu.

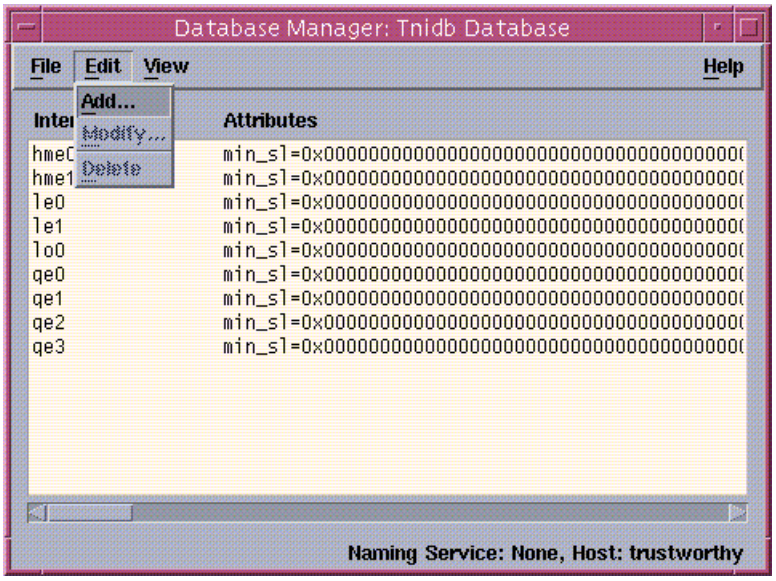


Figure 10–30 Add Option Selected from the Tnidb Edit Menu

b. Specify the name of the added interface and its desired attributes.
Figure 10–31 shows the Interface Manager (Add) Dialog Box.

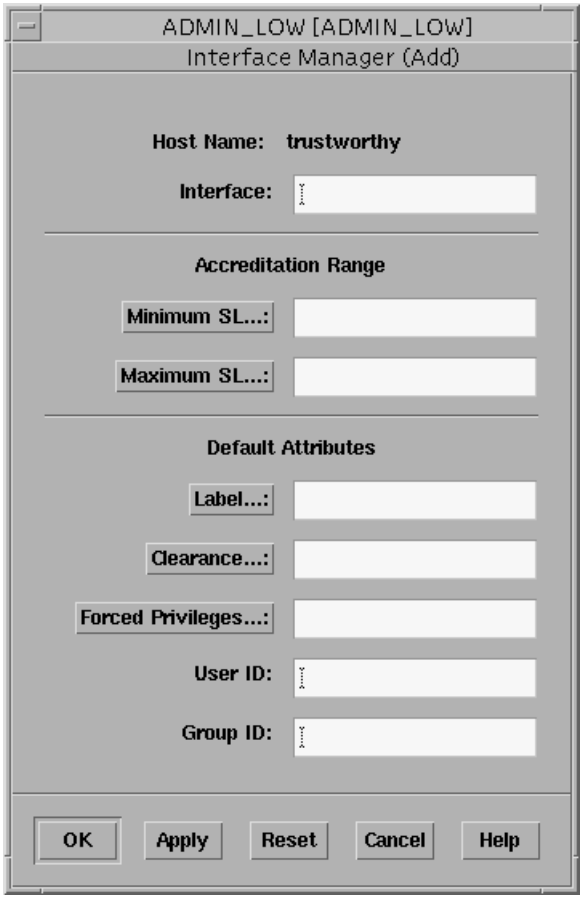


Figure 10–31 Interface Manager (Add) Dialog Box

▼ To Substitute a Valid CIPSO Label for the ADMIN_HIGH Sensitivity Label

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

   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Use the Admin Editor action to open the `/etc/system` file for editing.**

See “To Use the Admin Editor Action to Edit a File” on page 29, if needed.

3. **Add a line to set the tsol_admin_high_to_cipso= flag equal to 1.**

```
set tsolsys:tsol_enable_il=1
```

The default set in the kernel, which is not shown in the `system` file, is set to 0.

4. **Write and quit the file.**

```
:wq
```

▼ To Set Up a Simple Default Route for a Network with One Gateway

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

See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Use the Set Default Routes action to create an `/etc/defaultrouter` entry with the hostname of the router.**

See “To Launch Administrative Actions” on page 29, if needed. See also the route(1MTSOL) man page for more about the syntax and use of `/etc/defaultrouter`. The following example shows an entry for a default router called merlot.

```
merlot
```

3. **Make sure there is an entry for the gateway(s) in the local `/etc/hosts` file.**

```
129.150.113.36 merlot
```

4. **Make sure there is an entry for the gateway(s) in the local `/etc/security/tsol/tnrhdb` file.**

```
129.150.113.36:tsol
```

5. **Run the `tnctl`(1MTSOL) command for any host added to the `tnrhdb` file to update the trusted networking cache.**

```
$ tnctl -h merlot
```

▼ To Set Up Static Routes with Optional Emetrics for Specific Hosts or Networks

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

   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Use the Set TSOL Gateways action to open the `/etc/tsolgateways` file for editing.**

   See “To Launch Administrative Actions” on page 29, if needed. See also the tsolgateways(4TSOL) and route(1MTSOL) 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 fourth line shows a host entry set up 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 gateways(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.**

```
129.150.113.36:tsol
```

▼ To Set Up Trusted Routing

Note - Do the steps below to set up the Tnrhtp template and assign it to the sending host, to each of the trusted gateways through which the message should pass, and to the destination host.

---

1. **Access the Database Manager.**

   See "To Access the Trusted Network Databases from the Database Manager" on page 313 if needed.

2. **Double-click on the name of the Tnrhtp database, (as shown in Figure 10–32) to bring up the Template Manager, shown in Figure 10–33.**

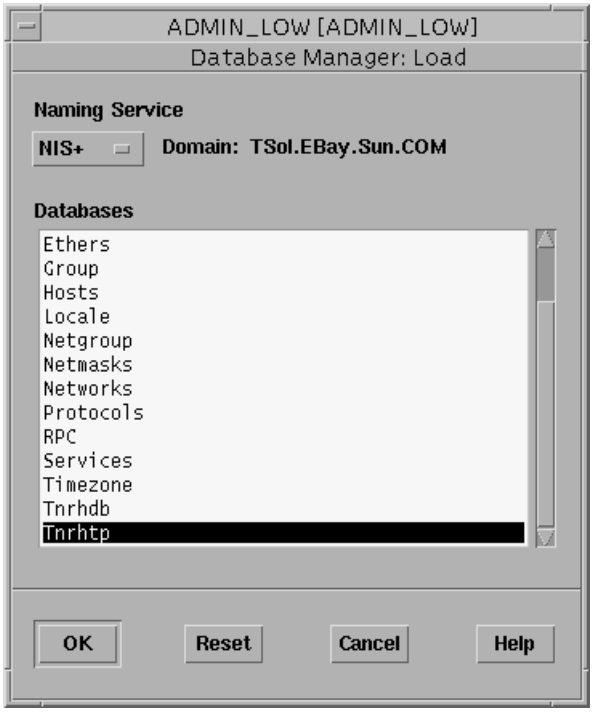


Figure 10–32 Database Manager: Load List with Tnrhtp Selected

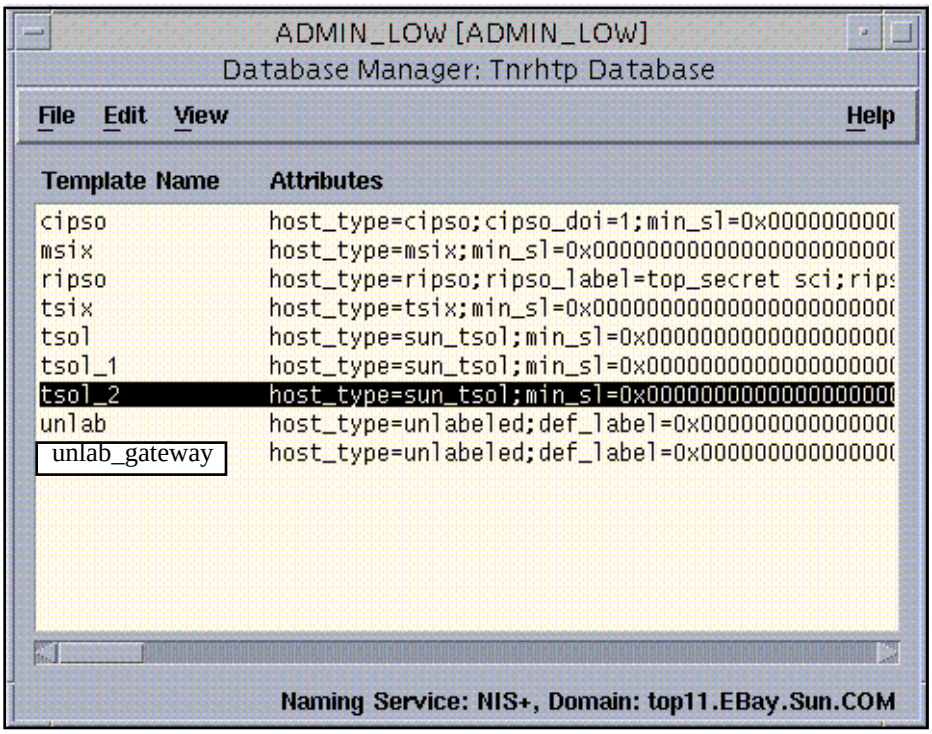


Figure 10–33 Database Manager: Tnrhtp Database Dialog Box with the tsol_2 Template Name Selected

3. Double-click on the name of the template to bring up the Template Manager, shown in Figure 10–34.

The tsol_2 template is selected in Figure 10–33.

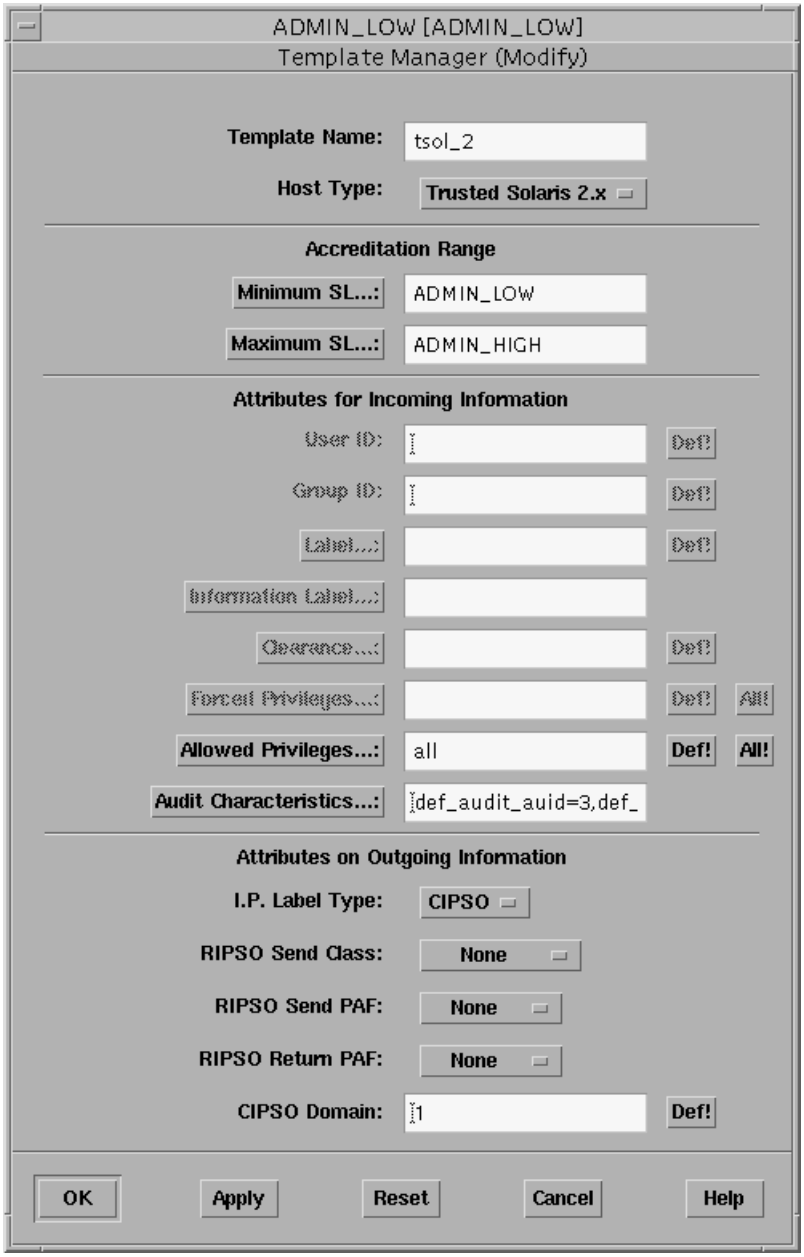


Figure 10–34 Trusted Network Template Manager Modify Dialog Box

4. Change any of the fields.

In the template for a gateway, you can use either the RIPSO, CIPSO, Trusted Solaris or TSIX host type.

To use a CIPSO label go to Step 5 on page 340. To use a RIPSO label, go to Step 6 on page 340.

5. **To use a CIPSO label, which is derived from the actual sensitivity label on the data, supply a CIPSO DOI.**

   Currently, the only defined CIPSO DOI is 1. Make sure to specify the same CIPSO DOI for the sender, all gateways, and the destination host.

6. **To define a RIPSO label, select one of the classifications from the RIPSO Send Class menu, choose zero or more of the supported Protection Authority flags from the RIPSO Send PAF menu, and choose one of the supported RIPSO errors in the RIPSO Return PAF menu.**

   Note - Make sure to specify the same RIPSO label and RIPSO error for the sending host, all gateways, and the destination host.

   ---

7. **Click OK to apply your changes and exit the Template Manager.**

8. **Use the Set Routes action from the System_Admin folder in the Application Manager to open the** `/etc/defaultrouter` **file.**

9. **List each gateway in the** `/etc/defaultrouter` **file by name or IP address one line per entry, using either the host's IP address or its host name.**

   Note - If the hostname is used, the host must also be listed with its IP address in the local `/etc/inet/hosts` file.

   The first example shows a gateway listed in `/etc/defaultrouter` by its hostname:

```
routeman
```

   The next example shows the same gateway called routeman listed in `/etc/defaultrouter` by its IP address:

```
127.13.104.10
```

10. **Write and quit the file.**

```
:wq
```

▼ To Set Up Tunneling

A forwarding host is any Trusted Solaris 2.5.1 gateway being set up to tunnel through one or more non-Trusted Solaris 2.5.1 gateway(s) to advertise the emetrics of its routes to Trusted Solaris 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 Login and Assume an Administrative Role" on page 15, 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 29, 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 steps Step 1 on page 341 through Step 4 on page 341 on the remote gateway(s), specifying the IP address for the local network.**

CHAPTER 11

Managing Files and File Systems

This chapter gives the background needed to understand how to manage and mount files, directories and file systems in the Trusted Solaris system. This chapter covers the following topics:

This chapter provides the following procedures.

Overview of Trusted Solaris Files, Directories, and File Systems

The Trusted Solaris system supports the same files and directories, most of the file system types, and all the file system management commands that are supported by the base Solaris system.

Whenever a process tries to access a file or directory, security attributes obtained from various sources are used in making access control decisions. New commands have been added to work with the extended Trusted Solaris security attributes.

This chapter describes what the security administrator needs to know about:

- How security attributes are obtained,
- How to set them and change their settings and
- How to specify attributes at mount time for filesystems that do not have them

Review of File, Directory, and Filesystem Access Terminology

The terms defined in this section are used in the discussions of how to manage files, directories, and file systems throughout this chapter. These terms and concepts are introduced in the *Trusted Solaris User's Guide* and the *Trusted Solaris Administration Overview*. This review is included here for convenience. You may choose to read through these definitions before going on to the rest of this chapter. Or, you can skip to "Security Attributes on Files and File Systems" on page 356 and then refer back to this section if you do not understand a term.

Within a definition, special terms used for the first time and terms that need emphasis are *italicized*. Words and phrases in **bold** within the definitions have their own definitions in this section.

Access Control List

An access control list (ACL) provides a type of **discretionary access control** based on a list of entries that the owner can specify for a file or directory. An access control list can restrict or permit access to any number of individuals and groups, allowing finer-grained control than provided by the standard UNIX **permission bits**.

Access Permissions

Read, write, and execute or search **permissions** on a file or directory are granted to a process if the **mandatory access control** checks are passed as described in **access policy for files, directories, and file systems**, and if one of the following tests is true.

- If an **ACL** exists for a file or directory, then the following tests are performed in order until one is true, and then the requested access is granted.
  - If the effective UID of the process is equal to the UID of the owner of the file or directory and if the ACL grants the desired type of access to the owner.
  - If the effective UID of the process is explicitly listed in the user list in the ACL and if both the ACL assigned to the named user and the ACL mask grant the desired type of access to the named user.
  - If the effective GID or a supplementary GID of the owner of process is equal to the GID of the file or directory and if both the ACL entry for the owner's group and the ACL mask both grant the desired type of access to the owner's group.

- If the effective GID or a supplementary GID of the owner of the process is named in the ACL group list, and if both the ACL entry for the named group and the ACL mask grant the desired type of access to the group.
- If the ACL's *other* entry grants the desired type of access to the owner of the process.

- If an ACL does not exist for the file or directory being accessed, then the following tests are performed in order, and if one of them is passed the desired access is granted.

  - If the effective UID of the process is equal to the UID of the owner of the file or directory and if the owner portion (0700) of the file's permissions is set to allow the desired type of access.
  - If the effective GID is equal to the GID of the file or directory and if the group portion (0070) of the file's permissions is set to allow the desired type of access.If one of the groups in the supplementary group list of the process is equal to the GID of the file or directory and if the group portion (0070) of the file's permissions is set to allow the desired type of access.
  - If the other portion (0007) of the file's permission bits is set to allow the desired type of access to all others.

Otherwise, access is denied, unless the process asserts the appropriate DAC override privilege(s), which are described in "Access Policy for Files, Directories, and File Systems" on page 346" and in "Execution Profile Mechanism " on page 349."

Access Policy for Files, Directories, and File Systems

Because in UNIX systems just about everything (including a spreadsheet, a printer, a letter, a chapter of a book, or a mail message) is handled as a file that is stored in a directory—to do just about anything the user must access files and directories. The conditions for access are described here. (Even though devices are treated as files in the UNIX system, devices have slightly different mandatory access rules than files or directories do, and these rules are separately described in this section.)

A file, directory, or device may be accessed in three ways:

- The *name* may be *viewed*,
- The *contents* may be *viewed*, or
- The *contents* or the *attributes* may be *modified*.

In the Trusted Solaris system, each of these types of access is granted or denied based on:

- Whether the basic UNIX **discretionary access control** checks have been passed, and

- Whether the **mandatory access control** checks have been passed.

All types of access require that the **sensitivity label** of the process **dominate**s the sensitivity label of all directories in the pathname and that the owner of the process (the person who executed the command) has discretionary search access for each directory in the pathname. View access to the name of the file, directory or device requires only that this part of the check is passed.

For view access (read access) to the contents or attributes of a file or a directory, the process' sensitivity label must dominate the sensitivity label of the file or directory. For view access to the contents of a device (for example, so you can read information stored on a tape in a tape drive), the process' sensitivity label must be equal to the sensitivity label of the device. The owner of the process also must have discretionary read access to the file, directory, or device.

For a process to write into a file or to modify the file's attributes, the sensitivity label of the file must dominate the sensitivity label of the process and must be within the process' clearance, which is set to be the **session clearance**. For a process to write into a directory (create a file), the sensitivity label of the process must equal the sensitivity label of the directory. For a process to write to a device (for example, store information on a tape in a tape drive), the sensitivity label of the process must also equal the sensitivity label of the device. The owner of the process must have discretionary write access to the file, directory, or device.

The security policy for device files can differ from the policy for regular files based on how the policy for devices is defined in the device_policy(4TSOL) file, which can be changed by the security administrator.

For each type of failure of a MAC or DAC check, a specific override **privilege** may be applied to the command, depending on the type of access being denied. A privilege can be made available to a command only by the action of a security administrator, because the security administrator must ensure that the user who executes the command is cleared, or that the command may be trusted, to use the privilege in a trustworthy manner.

These conditions and the listed override privileges apply to any type of access:

- The sensitivity label of the process must dominate the sensitivity label of all directories in the pathname, or the process must have the privilege to search up (search a directory whose sensitivity label dominates the sensitivity label of the process), which is *file_mac_search*.
- The user executing the command has discretionary search permission for all directories in the pathname, or the process must have the privilege to override search restrictions when accessing a directory, which is *file_dac_search*.

These conditions and the listed override privileges apply to view (read) access:

- The sensitivity label of the process must dominate the sensitivity label of a file or equal the sensitivity label of a directory or device, or the process must have the privilege to override MAC read restrictions, which is *file_mac_read*.

- The user executing the command must have discretionary read permission for the file or directory, or the process must have the privilege to override DAC read restrictions, which is *file_dac_read*.

These conditions and the listed override privileges apply to modify (write) access:

- The sensitivity label of file dominates or if the sensitivity label of a directory or device equals the sensitivity label of the process, or the process has the privilege that overrides MAC write restrictions, allowing the user to write up and to write above the user's clearance, which is *file_mac_write*.
- The user executing the command must have discretionary write permission for the file or directory, or the process must have the privilege to override DAC write restrictions, which is *file_dac_write*.

Accreditation Range

An accreditation range is actually not a range, but a set made up of labels. See "Label Range " on page 350, "User Accreditation Range " on page 356**,** and "System Accreditation Range" on page 355 for more about the types of label and accreditation ranges in the Trusted Solaris system.

Adorned Name

The text string *.MLD*. is the default adorned name for multilevel directories (MLDs). The adornment is alternately called the MLD prefix, and is used when accessing the MLD itself.

CMW Label

Consists of an **information label** followed by a **sensitivity label** in brackets, in the form: INFORMATION LABEL [SENSITIVITY LABEL].

Classification

The classification is the hierarchical portion of a **sensitivity label**, **information label**, or **clearance**, each of which has only one classification. In a sensitivity label assigned to a file or directory, a classification indicates a relative level of protection based on the sensitivity of the information contained in the file or directory. In a clearance assigned to a user and to processes that execute applications and commands on behalf of the user, a classification indicates a level of trust.

Clearance

The clearance is the upper bound of the set of labels at which a user may work. The lower bound is the minimum label assigned by the security administrator as the initial label. There are two types of clearance, the **user clearance** and the **session clearance**.

Compartments

Compartments are an optional set of words that the site's security administrator may specify to appear in a **sensitivity label**, **information label**, or **clearance**. The compartment represents areas of interest or work groups associated with the labels that contain them and with the files that are assigned the labels and the individuals that work with them.

Discretionary Access Control

Discretionary access control (DAC) is a type of access granted or denied by the owner of a file or directory at the *discretion* of the owner. The Trusted Solaris system provides two kinds of discretionary access controls (DAC): **permission bits** and **access control lists.**

Dominate

When any type of label (**sensitivity label**, **information label**, or **clearance**) has a security level equal to or greater than the security level of another label to which it is being compared, the first label is said to dominate the second. The **classification** of the dominant label must equal or be higher than the classification of the second label, and the dominant label must include all the words (**compartments** and **markings**, if present) in the other label. Two equal labels dominate each other. Sensitivity labels are compared for dominance when MAC decisions are being made. See **strictly dominate**.

Execution Profile Mechanism

The execution profile mechanism allows security administrators to bundle commands, CDE actions, **security attributes** that may be associated with those commands and actions, and user authorizations into an execution profile, which may then be assigned to one or more users based on the tasks that they need to perform. Security attributes that may be specified in execution profiles include inheritable privileges for commands and actions, which may in some cases be used to override

discretionary or mandatory access policy for files and directories and file systems. See “Overview of Trusted Solaris Files, Directories, and File Systems” on page 344.

Information Label

An information label is a type of label that signifies the actual security level of the information contained in a file or directory, and which may be used in deciding whether to downgrade the **sensitivity label** of the file or directory. The information label can also provide guidance for how to physically label information stored on backup media, and how to handle printed output or mail.

Information Label Floating

Information label floating is a conjoining of two **information label**s that occurs when a file or directory with one information label is accessed by a process that has another information label. The resulting conjoined information label reflects the combined security level of both information labels.

Label

A label is a security identifier assigned to a file or directory based on the level at which the information being stored in that file or directory should be protected. Depending on how the security administrator has configured the system, users may see the complete **CMW label**, only the **sensitivity label** portion, only the **information label** portion, or no labels at all.

Label Range

A label range is actually a *set* of **sensitivity label**s, which is specified by designating a maximum label and a minimum label. When applied to a command or an action in execution profiles, a label range limits the sensitivity labels at which the command or action may be executed. When applied to a file system, a label range limits the sensitivity labels at which information may be stored on the file system. When applied to a host or network in trusted networking databases, a label range restricts communications between the local host and the remote host or network, based on the remote’s host’s or network’s label range. A remote host that does not recognize labels is assigned a single sensitivity label, by making the host’s maximum sensitivity label equal to its minimum sensitivity label. A label range specified for a file system mounted from a remote host has to be within the remote host’s label range. For allocatable devices, a label range limits the sensitivity labels at which a device may be allocated and thereby restricts the sensitivity labels at which information can be

stored or processed using the device. For non-allocatable devices, such as a host's frame buffer or audio device, the label range restricts access to the host or to the audio device based on the accessing account's clearance. See also **network accreditation range**.

Mandatory Access Control

Mandatory access control (MAC) is a type of control based on comparing the **sensitivity label** of a file, directory, or device or another other thing being accessed to the sensitivity label of the **process** that is trying to access it. (The sensitivity label of the process is generally the same as the sensitivity label of the workspace where the command was invoked.) Even though directories and devices are managed like files in the UNIX system, the mandatory access control (MAC) rules that apply to directories and devices are different from the MAC rules that apply to files. Before a file may be accessed for writing, MAC checks ensure that the sensitivity label of the file dominates the sensitivity label of the process—enforcing a policy called *write up*. A process cannot write to a file whose sensitivity label is higher than the process' **clearance**, which is set to be equal to the session clearance. (The write up policy also allows for the sensitivity labels to be equal.) Before a directory or a device may be accessed for writing, MAC checks ensure that the sensitivity label of the directory or device is equal to the sensitivity label of the process—a policy called *write equal*. Before a file or directory may be accessed for viewing (reading or searching), MAC checks ensure that the sensitivity label of the process dominates the sensitivity label of the file or directory—enforcing a policy called *read down*. Before a device may be accessed for viewing, MAC checks ensure that the sensitivity label of the process equals the sensitivity label of the device, enforcing a policy called *read equa*l. (The read down policy also includes read equal.)

The rule that applies when a process at one sensitivity label attempts to read or write a *file* at another sensitivity label is write up, read down (WURD). The rule that applies when a process at one sensitivity label attempts to write a *directory* at another sensitivity label is write equal, read down. The rule that applies when a process at one sensitivity label attempts to write a *device* at another sensitivity label is read equal, write equal.

Markings

The codewords, handling caveats, control and release markings and the associated bits that apply to labeled information, markings are only contained in **information label**s.

Minimum Label

For users, the minimum label is the lower bound of the **sensitivity label**s at which a particular user can work, which is specified by the security administrator role while setting the user's account. For the system, the sensitivity label specified in the minimum label field by the security administrator in the `label_encodings` file sets the lower bound for all users.

Multilevel Directory

A multilevel directory (MLD) is a directory in which information at differing sensitivity labels is maintained in separate subdirectories called single-level directories (SLDs). An MLD appears to most interfaces to be a single directory under a single name.

In the Trusted Solaris system, certain directories (`/tmp`, `/var/mail`, and users' $HOME directories) are created as MLDs to accommodate standard applications that may run at differing labels and expect to be able to write files into these standard locations.

A user working in an MLD can only see and work with files at the sensitivity label of the current process. If a user has the upgrade file authorization or the downgrade file authorization, that user can upgrade or downgrade a file or directory's sensitivity label after it is created within an MLD. Names of upgraded files or directories are visible. The security administrator may change the switch tsol_hide_upgraded_names in the system(4) file from the default setting of 0 to 1 to hide the names of upgraded files.

The adorned name of an MLD, also called the MLD prefix, is a file system attribute that may be changed by the system administrator role using setfsattr(1MTSOL) with the `-m` option followed by a new prefix. The adornment may also be set on a file system at creation, using `newsecfs` with the `-m` option followed by a prefix. To find out the current MLD prefix, use getfsattr(1MTSOL) with the `-m` option. (See also "Working with MLDs" on page 52 in Chapter 2.")

When a file is created in an MLD in which an SLD at the correct sensitivity label does not already exist, Trusted Solaris creates the SLD and assigns to it the process' sensitivity label.

Users can access an MLD two ways: either *without* using the adorned name or with using the adorned name. Interfaces that access directories [such as cd(1), mkdir(1TSOL), and ls(1)] accept either form. When an MLD is referred to without using the adorned name, the Trusted Solaris system transparently extends the reference to the single-level directory (SLD) that corresponds to the sensitivity label of the window in which the command is invoked. For example, entering **`cd /tmp`** puts the user into the SLD that corresponds to the sensitivity label of the window in which the command is invoked, such as `/.MLD.tmp/.SLD.1`. Use of the adorned name allows programs to refer directly to the MLD instead of to the SLD that has the

same sensitivity label as the process. So if the user enters **cd /.MLD.tmp**, the user changes into the top level MLD, and the user can then list all the SLDs that the current window's sensitivity label dominates.

Permission Bits

Permission bits are a type of discretionary access control. The owner specifies permissions that are stored as a set of bits signifying who can read, write, or execute a file or directory. Three different sets of permissions are assigned to each file or directory: one set for the owner; one set for all members of the group specified for the file or directory; and one set for all others. See also **access control list**s.

Privilege

A privilege is a right granted to a process executing a command that allows the command or one or more of its options to bypass some aspect of security policy. A privilege should only be granted by a site's security administrator after the command itself or the person using it has been judged to be able to use that privilege in a trustworthy manner. Privileges are made available to an executing command either from the forced and allowed privilege sets assigned to the executable file or from inheritance of privileges. Privilege inheritance is managed by the execution profile mechanism. See also "Execution Profile Mechanism " on page 349.

Process

A process is the action that executes a command on behalf of the user who invokes the command. Each process receives a number of security attributes from the user, including the user ID (UID), the group ID (GID), the supplementary group list, and the user's audit ID (AUID). Security attributes received by a process include any privileges available to the command being executed, the process clearance (which is set to be the same as the session clearance), the sensitivity label of the current workspace, and an information label. If the option RESET IL ON EXEC is selected [see system(4)], the information label is set to be the lowest viewable label in the system when a new process is started. The information label floats if any information at a higher information label is accessed by the process.

Security Administrator

In an organization where sensitive information must be protected, the security administrator role is assigned to the person or persons who define and enforce the site's security policy and who are cleared to access all information being processed at

the site. In the Trusted Solaris software environment, the security administrator role is an administrative role that is assigned to one or more individuals who have the proper clearance and whose task is to configure the security attributes of all users and hosts so that the software enforces the site's security policy.

Security Attribute

Security attributes are used in enforcing the Trusted Solaris security policy. Sometimes Trusted Solaris-specific security attributes are called *extended* attributes because they are extensions to the security attributes used in the base Solaris operating environment. Various sets of security attributes, both from the base Solaris and the Trusted Solaris environments, are assigned to processes, users, files, directories, file systems, hosts on the trusted network, allocatable devices, and other entities. Security attributes for users from the base Solaris environment include the user ID (UID), audit ID (AUID), group ID (GID), supplementary group IDs (SGIDs), and access control lists (ACLs). Security attributes for users from the Trusted Solaris environment include the clearance, minimum label (initial label), and possible authorizations.

An important Trusted Solaris security attribute for files and processes is the CMW label. The sensitivity label portion of the CMW label is used in access decisions and the information label portion of the CMW label may be used to track the real sensitivity of the information contained in the file.

A sensitivity label range security attribute is assigned to file systems, to allocatable devices and to printers. Security administrators can associate a UID, GID, a label range, and one or more privileges with commands and CDE actions in execution profiles.

The other mentioned security attributes along with a sensitivity label range called the network accreditation range are assigned to hosts in Trusted Network databases, which are used to enforce MAC on communications in between hosts and networks.

The sensitivity labels at which NFS mounts may be performed are limited by the sensitivity label(s) assigned to the NFS server in the trusted network databases. See also "Attributes on File Systems" on page 361.

Security Policy

In the Trusted Solaris environment, the security policy is the set of DAC, MAC, and information-labeling rules that define how information may be accessed. For customers, the security policy is the set of rules that define the sensitivity of the information being processed at their site and the measures that are used to protect the information from unauthorized access.

Sensitivity Label

A sensitivity label is a security label assigned to a file or directory or process, which is used to limit access based on the security level of the information contained therein.

Session Clearance

A session clearance is a clearance in effect only during a particular login session, and it is set by the user when starting a session. Each process started during a session has a process clearance equal to the session clearance. The session clearance may be set either to be the same as or lower than the user clearance.

Single-level Directory

A single-level directory (SLD) is a directory within an MLD containing files at only a single sensitivity label. When a user working at a particular sensitivity label changes into an MLD, such as `/tmp`, the user's working directory actually changes to a single-label directory within the MLD, such as /.MLD.tmp/.SLD.1, whose sensitivity label is the same as the sensitivity label at which the user is working. SLD names are created using the .SLD. prefix followed by a number indicating the sequence in which they were created.

Strictly Dominate

When any type of label (sensitivity label, information label, or clearance) has a security level greater than the security level of another label to which it is being compared, the first label strictly dominates the second label. Strict dominance is dominance without equality, which occurs either when the classification of the first label is higher than that of the second label and the first label contains all the compartments in the second label or when the classifications of both labels are the same while the first label contains all the compartments in the second label plus one or more additional compartments.

System Accreditation Range

The system accreditation range is the set of all valid (well-formed) labels created according to the rules defined by each site's security administrator in the `label_encodings` file, plus the two administrative labels that are used in every Trusted Solaris system, ADMIN_LOW and ADMIN_HIGH.

User Accreditation Range

The user accreditation range is the set of all possible labels at which any normal user may work on the system, as defined by each site's security administrator. The rules for well-formed labels that define the system accreditation range are additionally restricted by the values specified in the ACCREDITATION RANGE section of the site's label_encodings(4TSOL) file: the upper bound, the lower bound, the combination constraints and other restrictions.

User Clearances

The user clearance is assigned by the security administrator to set the upper bound of the set of labels at which a particular user may work at any time. The user may decide to accept or further restrict that clearance during any particular login session, when setting the session clearance after log in.

Security Attributes on Files and File Systems

Attributes may be specified:

- At the level of an the individual file or directory within a file system

  See "Security Attributes on Files and File Systems" on page 356.

- At the level of the file system

  See "Attributes on File Systems" on page 361.

- At mount time

  See "Specifying Mount Time Security Attributes " on page 370.

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 371.

Attributes on Files and Directories

The security attributes shown in the following table are from the base Solaris operating system.

TABLE 11–1 Base Solaris Security Attributes

Security Attributes from Solaris Base
User Id
Group Id
Permission Mode
Access ACL (optional)
Default ACL (optional)

Along with the attributes of the base Solaris file system, Trusted Solaris files and directories have extended security attributes stored in a shadow inode off the object's inode. The following table gives the extended, security attributes required by Trusted Solaris security policy.

TABLE 11–2 File and Directory Attributes from the Base and from Trusted Solaris Operating System

Added Trusted Solaris Security Attributes	Description of Extended Trusted Solaris Attributes
Sensitivity Label	The sensitivity label of the file or directory.
Information Label	The information label of the file or directory. For directories, MLDs, and SLDs the information label may be undefined.
Forced Privileges	The set of privileges that an executable is guaranteed to have available at start of execution. Optional. Must be a subset of the allowed privileges.
Allowed Privileges	The maximum set of privileges that this executable is allowed to use during its execution. Optional. (Because editing executable files causes them to lose all their privileges, 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.

TABLE 11–2 File and Directory Attributes from the Base and from Trusted Solaris Operating System *(continued)*

Added Trusted Solaris Security Attributes	Description of Extended Trusted Solaris Attributes
File Attribute Flag	Optional. The only supported file attribute flag is public, whose effect is that when certain read operations are performed on any object in the file system on which this flag is set, audit records are not generated even when the operations are part of a preselected audit class, with the following exception. If the audit pseudo event for use of privilege (AUE_UPRIV) 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 read operations for which audit records are not generated when the public flag is set are: `access`(2TSOL), `fgetcmwlabel`(2TSOL), `fgetsldname`(2TSOL), `fstatvfs`(2TSOL), `getcmwfsrange`(2TSOL), `getcmwlabel`(2TSOL), `getfpriv`(2TSOL), `getmldadorn`(2TSOL), `getsldname`(2TSOL), `lgetcmwlabel`(2TSOL), `lstat`(2TSOL), `mldlstat`(2TSOL), `mldstat`(2TSOL), `open`(2TSOL) read only, `pathconf`(2TSOL), `preadl`(2TSOL), `readl`(2TSOL), `readlink`(2TSOL), `stat`(2TSOL), and `statvfs`(2TSOL).
Directory Attribute Flag	Optional. Flag that indicates a directory is an MLD

Unlike many systems implementing MAC, the Trusted Solaris system does not impose any order on the sensitivity labels of directories in a pathname.

Files and directories can be created only at the same sensitivity label as the containing directory. Privileged subjects can create files and directories and relabel existing files and directories at any valid sensitivity label to create *upgraded* or *downgraded objects*. See “To Change Labels and Privileges on Files and Directories ” on page 375.

The system may be configured so the names of upgraded files and directories are not visible. The Customize Trusted Solaris dialog that displays during installation allows the install team to choose `Hide upgraded names in directories`. This setting can be changed after installation by changing the setting of the `tsol_hide_upgraded_name`s switch in the `system` file as described in “`tsol_hide_upgraded_names`” on page 398 of Chapter 13” and rebooting.

Directory name 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 MAC and information label attributes.

A sensitivity label is set at creation and can be changed. The information label is set at creation, and can be changed without causing the process’s information label to float.

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 sensitivity labels. The mldpwd(1TSOL) and mldrealpath(1TSOL) commands are used to get the name either of the current working directory or of any other MLD with the MLD adornment. Mounting MLDs does not require the use of the adorned name.

When a command such as vi(1) tries to do something like create (write) a new file in a directory, the security attributes are compared to the corresponding security attributes (in this example the UID and sensitivity label) of the directory where the file is being created. The operation succeeds or fails based on whether the required MAC and DAC checks are passed. The user ID would have to meet the DAC requirements specified in the directory's permission bits and in the ACL, if one exists, and the sensitivity label of the process would have to be dominated by the sensitivity label of the directory.

If an MLD is automounted by a single-label host, the SLD that corresponds to the sensitivity label administratively assigned to the host in the trusted networking databases [tnrhtp / tnrhdb(4TSOL)] is mounted instead of the MLD. If, for example, a user's home directory is automounted on an unlabeled host, only the SLD that is at the sensitivity label assigned to the host is mounted.

Changing 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. Override privileges may be required if attempted accesses are outside MAC or DAC policy.

Changing Labels and Privileges

The File Manager Selected menu has a Change Labels options to set the sensitivity label and information label, which may also be done on the command line by any account that has the setlabel(1TSOL) command in one of its profiles. The Selected menu's Change Privileges option lets you 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(1TSOL) command in one of its profiles.

The following authorizations are required in order to set privileges and labels, either on the command line or through the File Manager Selected menu options:

- Setting privileges requires the *set file privileges* authorization.

- Upgrading file and directory labels requires the *upgrade file sensitivity label* authorization
- Downgrading file and directory labels requires the *downgrade file sensitivity label* authorization.

  See Figure 11–1 for the Selected menu. The options that allow the setting of privileges and labels are new in Trusted Solaris 2.x. See “To Change Labels and Privileges on Files and Directories ” on page 375 for how to change labels and permissions.

Changing File and Directory Attribute Flags

The getfattrflag(1TSOL) command gets the attribute flag of a file or directory and the setfattrflag(1TSOL) command sets the public object flag on a file and sets the MLD flag on a directory.

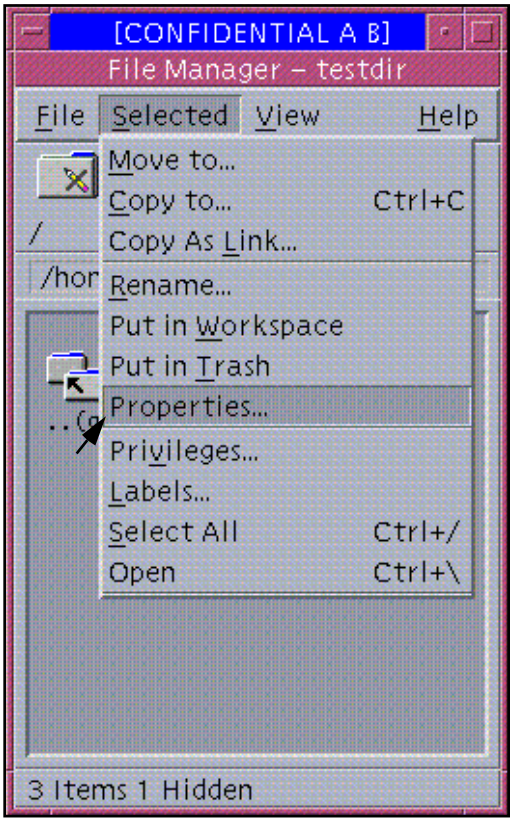


Figure 11–1 File Manager Selected Menu

Attributes on File Systems

File systems with the *tsol_attr* flag have a full set of Trusted Solaris extended security attributes already defined. Because the attributes can be changed on these file systems, they are called *variable* file systems. For example, the sensitivity label on a file in a mounted variable file system can be changed by an authorized user.

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 changed. For example, the sensitivity label specified for a mounted fixed-attribute file system cannot be changed on any of the objects in that file system. Only if an object is moved from the fixed file system can it be changed.

File systems supported by the Trusted Solaris system are characterized by whether their attributes can be changed or whether they are unchangeable. The two classifications are:

- Variable attribute file systems, and
- Fixed attribute file systems

  A file system with the extended Trusted Solaris security attributes is referred to as a *variable attribute* file system or *variable file system*. A file system that does not have the Trusted Solaris security attributes, but has only the standard Solaris attributes or some subset is referred to as a *fixed attribute file system*.

  All file systems installed with the Trusted Solaris system are variable attribute file systems, but there is nothing to stop the administrator from creating a fixed attribute file system on a disk or floppy device connected to the Trusted Solaris system. Therefore, both variable attribute file systems and fixed attribute file systems can be locally connected to a Trusted Solaris host, and can even coexist on the same local disk.

Variable Attribute File Systems

All Trusted Solaris file systems are variable attribute file systems and have the *tsol_attr* flag as one of their extended security attributes, while other types of file systems from other UNIX systems do not have this flag and do not recognize it. On a variable attribute file system with the *tsol_attr* flag:

- The *Trusted Solaris security attributes are present* and
- The security attributes *may be changed.*

  The following table shows the security attributes for variable-attribute file systems, with the default values that are used when none are specified at file system creation.

TABLE 11–3 Trusted Solaris File System Security Attributes with Defined Settings

Attribute	Description	Defaults
Attribute Flag	Only supported option is tsol_attr flag	tsol_attr
MLD prefix	The characters to use for the MLD prefix for MLDs on this file system	.MLD
Sensitivity Label Range	The minimum and maximum sensitivity level for files and directories created on this file system	ADMIN_LOW to ADMIN_HIGH
Sensitivity Label	Sensitivity label to infer for all files and directories on this file system that do not have an explicit sensitivity label	none
Information Label	Information label to infer for all files and directories on this file system that do not have an explicit information label (default *none*).	none
Access ACL	Access ACL to infer for files and directories on this file system that do not have an explicit access ACL (see setfacl(1) for the format)	none
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

Specifying Variable Attributes on File Systems

All file systems created during Trusted Solaris installation have the *tsol_attr* flag. Newly-created systems can be used just as they are with the defined set of security attributes. However, when site security policy requires, the security administrator can:

- Use the getfsattr(1MTSOL) command to get the security attributes of a file system.
- Use the newsecfs(1MTSOL) command to supply security attributes for a new Trusted Solaris file system while creating it.

See “To Specify Alternative Security Attributes While Creating a Local File System” on page 379

- Use setfsattr(1MTSOL) to tune the attributes set on an already-existing file system

  See “To Set Security Attributes on a Standard File System or Reset Security Attributes for an Existing Trusted Solaris File System ” on page 379.

Warning - 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.

Fixed Attribute File SystemsBecause they are configured to have a single sensitivity label when mounted on Trusted Solaris hosts, fixed attribute file systems are also referred to as single-label file systems.

Security administrators may associate a single CMW label (information label with sensitivity label) and a set of attributes with a fixed attribute file system, with the result that these attributes are applied to all the files and directories in the mounted file system. On fixed attribute file systems, neither the information label or the sensitivity label can change as long as an object resides in the file system. The security attributes are specified for the mounted fixed file system either on the command line using `mount` with the `-S` option or in the vfstab_adjunct(4TSOL) file.

The following example shows the entry to NFS_mount a fixed attribute file system called `/spare` from a NFS server running Solaris operating environment. The service is called *outside*. /spare is mounted with a CMW able of PUBLIC[INTERNAL_USE_ONLY] using `mount` with the `-S` option on the command line as shown here:

```
$ mount -F nfs -S ''slabel=INTERNAL_USE_ONLY; ilabel=PUBLIC'' outside:/spare /spare
```

If the mounted file system `/spare` contains a file called `test`, no one can change the sensitivity label or information label of `/spare/test`, and its information label can never float. However, if `/spare/test` is copied into another directory such as `/tmp` or `/export/home/secadmin`, its label can be changed.

When a fixed attribute file system (such as a file system from a Solaris system) is being mounted, any security attributes not specified at mount time are assigned default values. Table 11–4 shows the values used when a fixed attribute file system that does not support an attribute is being mounted if a mount-time value for the attribute has not been supplied either on the `mount` command line or in `vfstab_adjunct` entry for the file system.

TABLE 11–4 Attributes Assignable to Fixed File Systems

Attribute	Defaults	
UID	Must be assigned if not a UID is not defined for file system's objects, for example when mounting a DOS file system from a floppy.	
GID	Must be assigned if not a UID is not defined for file system's objects, for example when mounting a DOS file system from a floppy.	
mode	Must be assigned if a UID is not defined for file system's objects, for example when mounting a DOS file system from a floppy.	
Attribute Flag	None	
MLD prefix	empty string	
Sensitivity Label Range	ADMIN_LOW to ADMIN_HIGH	
Sensitivity Label	When a fixed file system is being mounted from a local hard disk	mounting process's sensitivity label.
	When a fixed file system is being mounted from a CD-ROM or floppy disk	allocating process's sensitivity label, or the sensitivity label of the existing mount point when mount is done separately from allocation
	When a fixed file system being mounted from a NFS server	the default sensitivity label administratively assigned to the server in its trusted network entries
Information Label	ADMIN_LOW	
Access ACL	None	
Forced Privilege Set	None	
Allowed Privilege Set	None	

Note - Empty means that the attribute has all bits off. None means that the attribute has no effect.

Types of File Systems that Can Be Mounted in the Trusted Solaris System

Trusted Solaris mount(1MTSOL) can be used to mount the following types of file systems:

- FDFS
- HSFS
- LOFS
- NFS
- PCFS
- PROCFS
- TMPFS
- UFS

The CACHEFS filesystem type is not supported.

Multiple `mount_*` mount man pages are available, such as mount_nfs(1MTSOL) and mount_ufs(1MTSOL), but there is only one mount(1MTSOL) command. The `mount` man page describes security attributes that may be set at mount time and gives the privileges, UID and GID that `mount` needs in order to succeed. See also “Attributes on File Systems” on page 361 and following for more about security attributes.

The options for doing mounts of the various file system types are described in the appropriate mount_* man pages. The vfstab_adjunct(4TSOL) man page describes the `/etc/security/tsol/vfstab_adjunct` file, where mount time security options can be entered to be used when file systems specified in the `/etc/vfstab` are being mounted. The differences between fixed attribute and variable file systems are described in “Fixed Attribute File SystemsBecause they are configured to have a single sensitivity label when mounted on Trusted Solaris hosts, fixed attribute file systems are also referred to as single-label file systems. ” on page 363.

Table 11–5 lists the various types of mounts with examples and notes.

TABLE 11–5 Mount Types, Examples, and Notes

Type	When Used	Notes
FDFS	A pseudo file system type that allows a program to access its own file descriptor 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 sensitivity label and information label are those of the backing file or directory. This is a fixed attribute file system.
HSFS	Mounts a file system from a CD device.	See mount_hsfs(1MTSOL). 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 lofs(7FS). 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 mount_nfs(1MTSOL). See also “Trusted Solaris NFS Mounts” on page 373. NFS mounts can be performed on fixed and variable attribute file systems.
PCFS	Mounts DOS file systems from a diskette.	See pcfs(7FS). 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 /proc 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 `/proc` file, which are imputed from the underlying process’s DAC and MAC attributes.If the calling process has the proc owner privilege, then the process can get information at the same sensitivity label about processes not owned by the caller. If the calling process has proc_mac_read privilege, the process can get information about a process that is owned by the caller when the process’s sensitivity label dominates that of the caller or is disjoint. The restrictions for modifying are more granular than the ones for reading. See the proc(4TSOL) man page.

TABLE 11–5 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 `/tmp` 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 mount_tmpfs(1MTSOL).
UFS	Mounts a file system from a local disk	See mount_ufs(1MTSOL). UFS file systems can have fixed mount time attributes assigned or variable attributes assigned at creation or later when the file system is given the tsol_attr flag. See “Variable Attribute File Systems” on page 361. Note: On unlabeled (fixed attribute) file systems, the MLD prefix generally has no useful effect—with the following exception. An mld_prefix should be supplied if another file system that has the tsol_attr flag is being mounted on the unlabeled file system and if the root of that file system is an MLD. If no prefix is supplied; the default is an empty string.

UFS and NFS mounts are the types of mounts most commonly done.

MLDs are supported only by the following file system types:

- `ufs` -variable (with the tsol_attr attribute)
- `nfs-variable` (with the tsol_attr attribute)
- `lofs`, and
- `tmpfs`

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 vfs`tab(4TSOL)` 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 filesystem. The `mount` restrictions shown in the following table are supported on all filesystem types. The Default Values column shows the values used when the system administrator does not specify an option.

TABLE 11–6 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 specials, 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 2.5 version of mount.

Summary of Attributes on Various Filesystem Types

Depending on the filesystem, some attributes can be bound to individual objects (files or directories), while other exist only at the filesystem level. Attributes at the filesystem level can be provided by the filesystem itself, or can be specified at mount time. If a filesystem provides an attribute even if by use of a system default value), any mount-time specification of that attribute is ignored.

The mount-time specification of filesystem security attributes can be done on the command line using `mount` with the –`S` option, in the `vfstab_adjunct` file, or in `auto_master` and in `autofs` map entries with the –`S` option. If security attributes are not specified in either `auto_master` or an `autofs` map entry, but an entry for the mount point is in the `vfstab_adjunct` file, the security attributes in the `vftab_adjunct` are used. See the mount(1MTSOL), vfstab_adjunct(4TSOL), and automounat(1MTSOL) man pages.

The following table indicates how various filesystems support the various filesystem attributes. See the key in Table 11–8

TABLE 11–7 Attributes Supported by the Supported Filesystem Types

Attribute	TUFS/TNFS	TMPFS/FxUFS/SLNFS	PCFS/HSFS
Allowed privileges	FS	MT	MT
Forced privileges	FS	MT	MT
CMW label	FS	MT	MT
MLD prefix	FS	MT	MT
Label range	FS	MT	MT
Audit preselection mask	FS	MT	MT
Filesystem attribute flags	FS	none	none
Object attribute flags	FS	MT	MT
Mount flags	MT	MT	MT
Access ACL	OBJ	OBJ	MT
File mode	OBJ	OBJ	MT*
File owner	OBJ	OBJ	MT*
File group	OBJ	OBJ	MT*

TABLE 11–8 Key to Table 11–7

Filesystem Type		Where Attribute Obtained	
TUFS	a ufs filesystem with tsol attributes	FS	Attributes specified by filesystem
TNFS	a tnfs filesystem from a Trusted Solaris or TSIX server	MT	Attributes specified at mount time
TMPFS	a tmpfs filesystem	OBJ	Attributes are attached to all objects in filesystem
FxUFS	a fixed (single-label) ufs filesystem	MT*	for hsfs with Rock Ridge extensions: same as OBJ

TABLE 11–8 Key to Table 11–7 *(continued)*

Filesystem Type		Where Attribute Obtained
SLNFS	a NFSv2 filesystem or a NFSv3 filesystem from a single-label/unlabeled server	
PCFS	a pcfs filesystem	
HSFS	an hsfs filesystem	

Specifying Mount Time Security Attributes

Even though they can be changed during normal operations, security attributes on variable file systems cannot be overridden at mount-time.

Any security attributes that are not supported on a filesystem can be specified at mount-time for the following file systems:

- File systems that do not support any attributes (such as DOS file systems)
- File systems that support UIDs, GIDs, and modes, but that do not support the Trusted Solaris extended security attributes (such as sensitivity labels on file system objects)

  When an attribute is not specified, a default value is applied.

  The `mount` command is extended to allow default values to be specified for privilege sets, ACLs, the sensitivity label, and the information label when a fixed attribute file system is being mounted. For file systems with tsol_attr flag, the mount protocol is extended to retrieve the extended attributes from the file system whether it is local or remote.

  The security administrator can specify security attributes at mount time either by using the mount -S option or setting the attributes in the file system's entry in the vfstab_adjunct(4TSOL) file. See the mount(1MTSOL) man page and:

  - "To Specify Mount-time Security Attributes on the Command Line " on page 380
  - "To Specify Mount-time Security Attributes in the Mount Table " on page 381

Allowing the system administrator to specify a set of attributes at mount time:

- Supports mounting of fixed attribute file systems that do not support imbedded security attributes at either the file system or file and directory level

  When the file system is mounted, any specified mount attributes are first checked against the list of supported attributes for the file system type mounted. If a mount attribute is not supported for the file system being mounted then the mount fails and an error is returned to the caller. For example, on ufs file systems the mount user ID is not supported, because user IDs are supported on ufs file systems. The second check is made on the validity of the attribute value. If the attribute value is not valid then the file system is not mounted and an error will be returned to the caller. For example, the mount forced privilege set must be a subset of the mount allowed privilege set.

  A single-label fixed attribute file system allows access to files and directories only at the single sensitivity label range specified when the file system is mounted. Also, sensitivity labels and information labels are not changeable on individual files and directories within the single-label file system. Also, because there is no information labels support, the information labels of all files and directories in a single-label file system are treated as equal to the value specified at `mount` time, do not float and cannot be changed by user action. It is the administrator's responsibility to ensure that the label range and the CMW label specified accurately represent the label of all data.

  A file system mounted from a host configured as an unlabeled host supports UIDs, GIDs, and optional ACLs and default ACLs on the files and directories it contains, but has no other security attributes. The security administrator specifies a sensitivity label range that is actually a single label for the file system along with other default security attributes.

Trusted Solaris Attribute Precedence Rules

A file or directory's attributes take precedence over the attributes on the containing file system. If a file system has Trusted Solaris extended attributes, attributes specified at mount-time are ignored. Any attributes not obtainable from the file system are given from the defaults.

Figure 11–2 illustrates the rules.

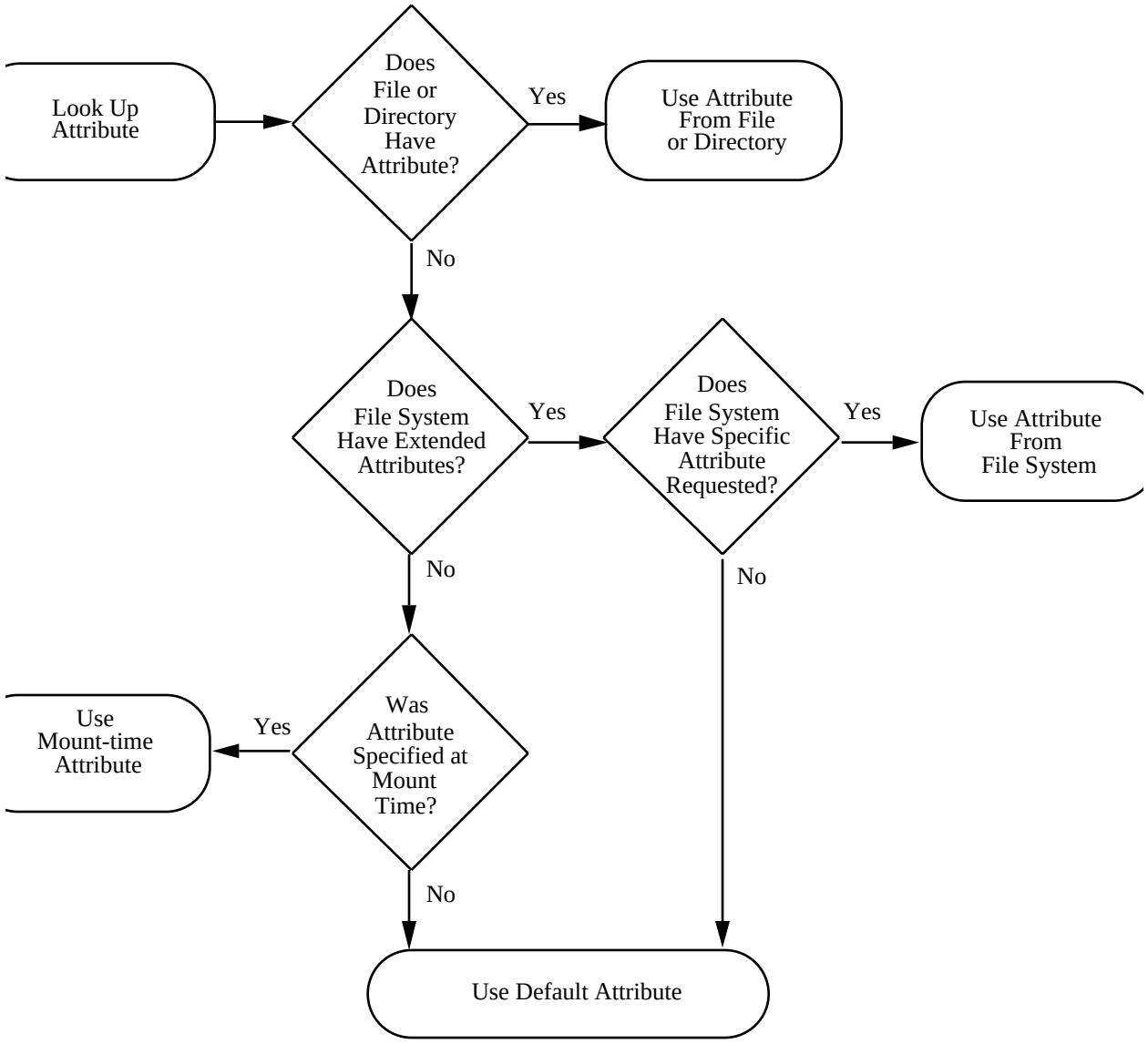


Figure 11–2 Trusted Solaris Attribute Precedence Rules

Example of Specifying Security Attributes for a Fixed Attribute File System Mounted from an Unlabeled Host

In this example, the security administrator role has specified the following security attributes for the `/public` file system being mounted from a Solaris host in the `vfstab_adjunct` entry for the file system:

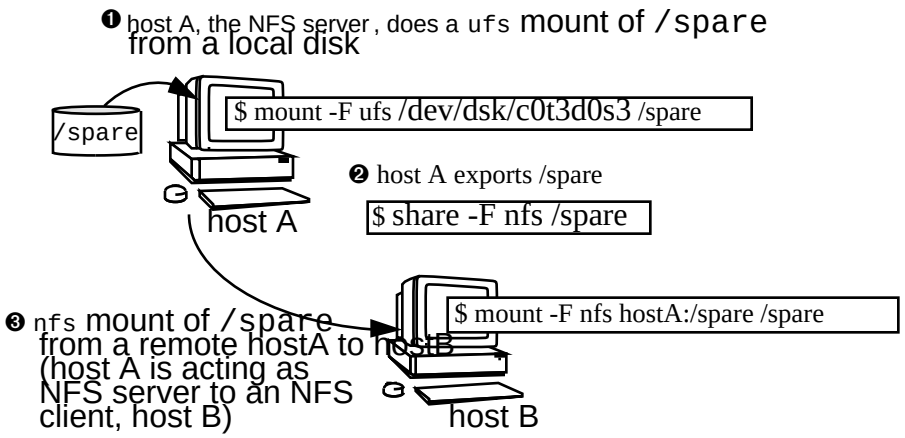
- A minimum sensitivity label and a maximum sensitivity label of PUBLIC
- A default sensitivity label of PUBLIC
- A default information label of PUBLIC
- An allowed privilege set of none
- A forced privilege set of none

  Normal DAC rules apply to attempts to access files and directories in this file system. With the listed defaults set for the file system:

- A user can access a file in that file system (such as `/public/stock.quotes`) only while in a window with a sensitivity label of PUBLIC.
- No action of the user can change the information label or sensitivity label of the file as long as it resides in the `/public` file system.
- If the user copies or moves a file from the `/public` file system to a variable file system with the tsol_attr flag, the file's CMW label changes to that of the process that copies it, and the rest of the file's security attributes can then be changed.

Trusted Solaris NFS Mounts

When mount(1MTSOL) is invoked to do an NFS-type mount, `mount` is invoked by definition on a Trusted Solaris NFS client to mount a file system from an NFS server. The NFS server may or may not be running the Trusted Solaris operating environment. The NFS server has the file system locally-mounted, and the file system can be any local file system type, hsfs, pcfs, tmpfs, ufs, and so forth. The NFS server exports the file system using share(1MTSOL). The `share` command and its options are either entered on the command line or in the `/etc/dfs/dfstab` file.



Trusted Solaris and NFS

Trusted Solaris 2.5 supports both of the Network File System (NFS) protocols supported in the base Solaris operating environment and the previous major release of Trusted Solaris:

- NFS Version 2 (V2) (from the Solaris 1.x environment)
- NFS Version 3 (V3) (from the Solaris 2.5 environment)

When a Solaris host exports a file system using the NFS protocol, a Trusted Solaris 2.5 host can use the corresponding NFS protocol version to access the file system at a single label.

A Trusted Solaris host can also use the NFS protocol to export its own file systems to unlabeled client hosts. The unlabeled client ignores the Trusted Solaris security attributes. A file or directory exported to an unlabeled client is *writable* if its sensitivity label equals the sensitivity label associated with the client host in its trusted networking database entries. A file or directory exported to an unlabeled client is *readable* only if its sensitivity label is dominated by the sensitivity label associated with the client host.

To support data sharing with Trusted Solaris 1.x hosts, Trusted Solaris 2.5 partially supports the 1.x version of Trusted NFS. Only the client portion of these protocols is supported so that Trusted Solaris 1.x hosts can export file systems to Trusted Solaris 2.5 hosts. Since some of the extended attributes (such as privileges) are different between the different versions of Trusted Solaris operating environment, not all attributes are honored. Specifically, the extended attributes are limited to sensitivity label, information label, and permission bit DAC. No guarantee is made that ACLs are compatibly implemented with Trusted Solaris 1.x. No guarantee is

made with respect to execution of files from a Trusted Solaris 1.x host. In particular, no privileged programs can be executed from a Trusted Solaris 1.x host.

Note - And privileged processes on Trusted Solaris 2.5 may not have their privileges interpreted on a Trusted Solaris 1.x server.

Any file system being mounted from a NFS server running Solaris 2.4 or earlier versions of Solaris or running Trusted Solaris 1.x needs to 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 host'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 for Mounting by Other Hosts

Exporting directories (sharing) for mounting by other hosts is done the same way it is done in the base Solaris system. Two new Trusted Solaris mount options nodev and nopriv may also be used when sharing file systems. See "To Share a Directory for Mounting by Other Hosts " on page 383.

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 Trouble Shoot Mount Failures " on page 385.

File and File System-related Procedures

▼ To Change Labels and Privileges on Files and Directories

1. **Assume the security administrator role.**

"To Login and Assume an Administrative Role" on page 15, 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.**
   a. **On the File Manager Privileges dialog box shown in Figure 11–4, 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.**
   a. **On the File Manager Label Builder dialog box shown in Figure 11–4, check the button for SL or IL, as needed, and enter a label**

      Do either of Step 4 on page 376 or Step 4 on page 376.

      i. **Type in the text entry field under Update With.**
      ii. **Click the desired classification, compartments or markings, as appropriate.**

   a. **Click OK.**

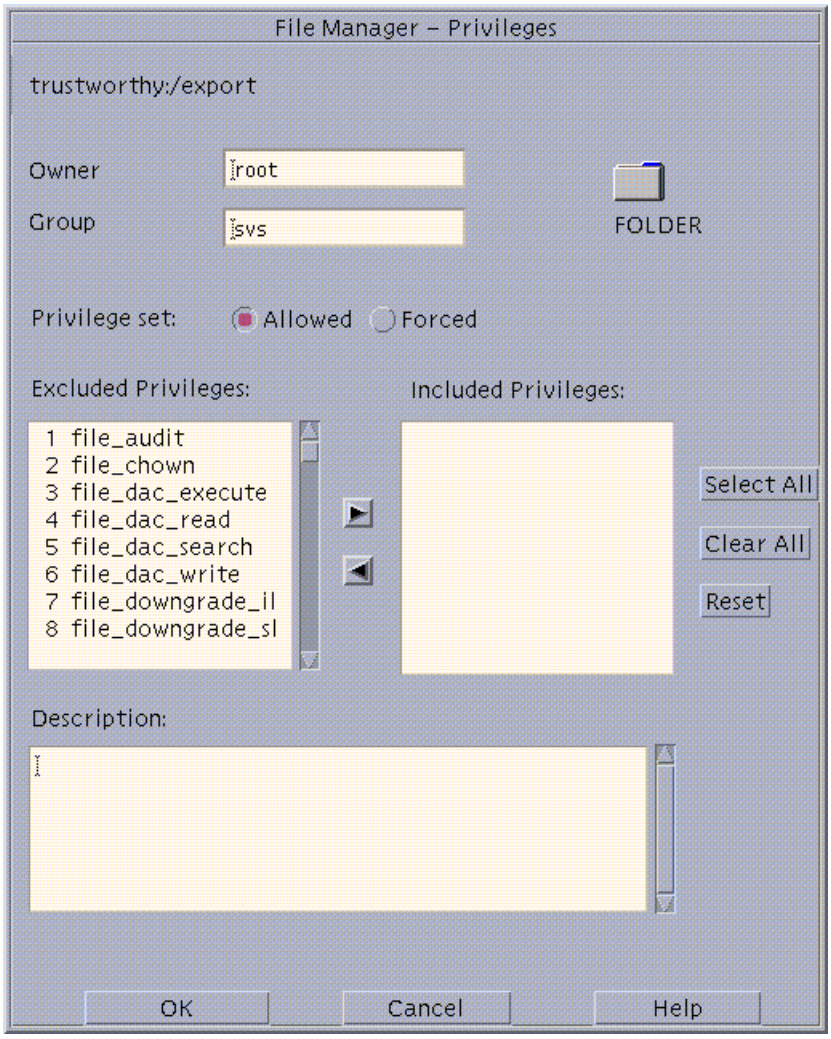


Figure 11–3 File Manager Privileges Dialog Box

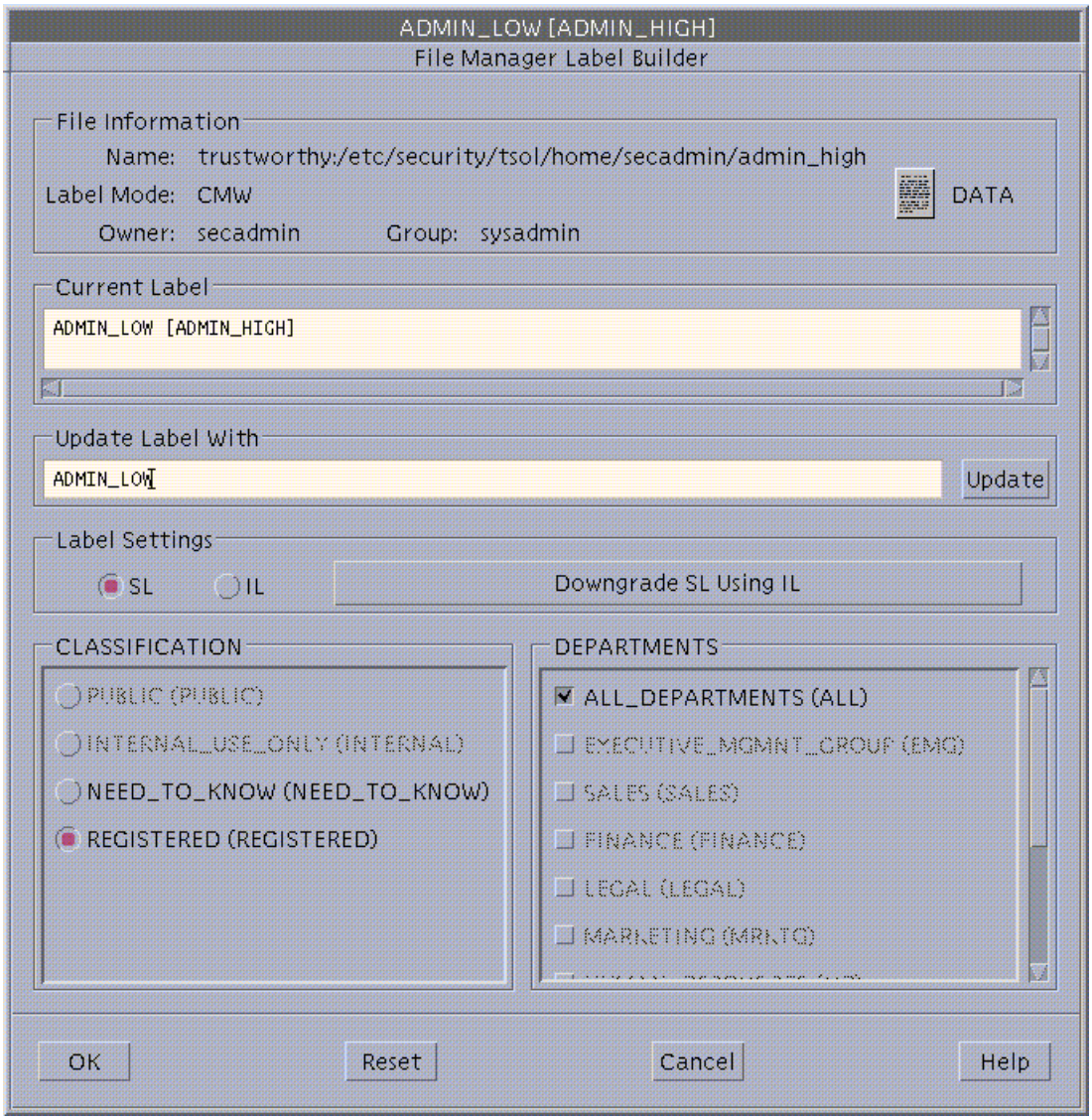


Figure 11–4 File Manager Label Builder

▼ To Specify Alternative Security Attributes While Creating a Local File System

1. **Assume the administrator role and go to an ADMIN_LOW workspace.**
   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Using the File Manager or the** `mkdir(1TSOL)` **command in a profile shell [pfsh(1MTSOL)make the mount point directory.**

```
$ mkdir /newpublic
```

3. **Use the Set Mount Points Action to open the vfstab(4TSOL) file for editing.**
   See “To Launch Administrative Actions” on page 29, if needed.

4. **Make an entry for the file system in the vfstab(4TSOL) file.**

```
/dev/dsk/c0t3d0s3       /dev/rdsk/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(1MTSOL)], execute the newsecfs(1MTSOL) command with the options that specify the desired alternative security attributes, then mount the file system.**

```
$ newsecfs -l ‘‘Secret;Secret’’ /newpublic
   $ mount /newpublic
```

▼ To Set Security Attributes on a Standard File System or Reset Security Attributes for an Existing Trusted Solaris File System

1. **Assume the administrator role and go to an ADMIN_LOW workspace.**
   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **In a profile shell [pfsh(1MTSOL)], enter** `umount` **to unmount the file system.**

```
$ umount /spublic
```

3. **Use the Set Mount Points Action to open the vfstab(4TSOL) file for editing.**
   See “To Launch Administrative Actions” on page 29, if needed.

4. **Make sure that an entry exists for the file system in the** `vfstab`**(4TSOL) file.**

```
/dev/dsk/c0t3d0s4       /dev/rdsk/c0t3d0s4      /spublic  ufs     2    yes     -
```

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

6. **In a profile shell [pfsh(1MTSOL)], execute the setfsattr(1MTSOL) command with the appropriate arguments, then remount the file system.**
   The following example sets a sensitivity label range of SECRET to SECRET.

```
$ setfsattr -l ‘‘Secret;Secret’’ /spublic
   $ mount /spublic
```

Warning - 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 Login and Assume an Administrative Role” on page 15, 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;mld_prefix=hidden-’’ swap /mnt
```

   The example mounts a tmpfs-type file system, swap, on `/mnt`.

▼ To Specify Mount-time Security Attributes in the Mount Table

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

   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Use the Set Mount Points administrative action to open the vfstab(4TSOL) file for editing.**

   See “To Launch Administrative Actions” on page 29, if needed.

3. **Make an entry for the file system in the vfstab(4TSOL) file.**

```
/dev/dsk/c0t3d0s4   /dev/rdsk/c0t3d0s4   /spublic  ufs     2    yes     nodevices,nopriv,nosuid
```

   - The `nosuid` mount flag prevents users from using `setuid` in the file system.
   - The `nodevices` mount flag disallows opens on device special files.
   - The `nopriv` mount flag disallows using privileges on files in the file system.

4. **Save and close the file.**

```
:wq
```

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

6. **Use the Set Mount Attributes Action to open the vfstab_adjunct(4TSOL) file for editing.**

7. **Copy and paste the template entry, and modify the copy.**

   The example on Code Example 11–1 gives the following security attributes to `/spublic`:

   - All files in the file system gets an slabel (sensitivity label) of SECRET A so they can only be accessed at that sensitivity label or at a sensitivity label that strictly dominates them.

CODE EXAMPLE 11–1 Example `vfstab_adjunct` Entries

```
#
   #       Yank the following entry and use as a template.
   #
   #<mount point>; \
   #acc_acl=; \
```

```
        .
        .
        .
#audit_psa=;
#
#       attributes for an unlabeled filesystem
#
/spublic;\
acc_acl=; mode=; attr_flg=; gid=; uid=; ilabel=;\
slabel=''Secret A''; forced=; allowed=; low_range=''Secret A'';\
hi_range=''Secret A''; mld_prefix=; mnt_flg=; audit_psa=;
#
#       attributes for an HSFS file system to mount from a
#       CD-ROM
#
/cdrom;\
acc_acl=; mode=; attr_flg=; gid=; uid=; ilabel=;\
slabel=; forced=127; allowed=all; low_range=;hi_range=;\
mld_prefix=hidden-; mnt_flg=; audit_psa=;
#
#       automatically mounted by /etc/init.d/MOUNTFSYS
#
/tmp;\
acc_acl=; mode=; attr_flg=; gid=; uid=; ilabel=;\
slabel=; forced=127; allowed=all; low_range=;hi_range=;\
mld_prefix=hidden-; mnt_flg=; audit_psa=;
```

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

```
:wq
```

▼ To Share a Directory for Mounting by Other Hosts

1. **Assume the administrator role in an ADMIN_LOW workspace.**
   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Use the Share Filesystems action to open the** `/etc/dfs/dfstab` **file for editing.**
   See “To Launch Administrative Actions” on page 29, if needed.

3. **Make an entry for the file system you wish to export.**

```
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 profile shell, run** `shareall` **to tell the NFS daemon nfsd(1MTSOL) to reread** `/etc/dfs/dfstab`**.**

```
$ shareall
```

▼ 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 Login and Assume an Administrative Role” on page 15, 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;mld_prefix=hidden-’’ 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, 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. If the answer is yes, the file system is automatically mounted.

▼ To Automatically Launch a CD Player for an Audio CD-ROM

As described in Chapter 15," under "Handling of CD-ROM Devices" on page 469, 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 administrator role in an ADMIN_LOW workspace.**
   See "To Login and Assume an Administrative Role" on page 15, 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 29, if needed.

3. **Add an action to automatically launch a CD player.**

```
action cdrom action_workman.so /path/to/workman_app
```

   The following example shows how the security administrator can put an action for the commonly-used CD player, `workman`, in `rmmount.conf` to automatically bring up `workman` from `/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 384 has been done.**

3. **Allocate the audio and the cd_rom_*N* devices at your working sensitivity label.**

4. **When prompted, insert the audio CD into the device.**

The specified CD player program will automatically be launched.

▼ To Trouble Shoot Mount Failures

1. **Make sure that the IP address of the host sharing the file system is in the tnrhdb(4TSOL) file on the Trusted Solaris host doing the mounting.**

   If the NIS+ naming service is being used, the IP address of the host sharing the file system must be listed in the `tnrhdb` table in the NIS+ domain; if no naming service is being used, the IP address of the host sharing the file system must be in the `/etc/security/tsol/tnrhdb` file on the Trusted Solaris host doing the mounting.

2. **If the host is not running the Trusted Solaris system, make sure the host has a valid sensitivity label assigned in the tnrhtp(4TSOL) template and that the same sensitivity label is being used for the mount, whether the label range is entered after `mount -S` on the command line or in the `/etc/security/tsol/vfstab_adjunct` file.**

3. **If the remote single-label host is newly added to the trusted network databases or its template has been changed since the local host has been rebooted, make sure the entry for the single-label host has been updated with `tnctl -h` *hostname*.**

4. **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 specifies the security attributes of mounts while the system administrator takes care of the normal Solaris aspects of mounting.

5. **When mounting any file system from a NFS server running Solaris 2.4, earlier versions of Solaris, or Trusted Solaris 1.x, with *vers=2* and *proto=udp* options to mount(1MTSOL).**

TABLE 11–9 File and Directory Attributes from the Base and from Trusted Solaris Operating Environments

Security Attributes	Defaults
User Id	
Group Id	

TABLE 11–9 File and Directory Attributes from the Base and from Trusted Solaris Operating Environments *(continued)*

Security Attributes	Defaults
Permission Mode	
Access ACL (optional)	none
Default ACL (optional)	none
Sensitivity Label	
Information Label	
Forced Privileges	none
Allowed Privileges	none
File Attribute Flag	none
Directory Attribute Flag	none

CHAPTER 12

Managing NIS+

To achieve the desired uniformity of user, host, and labels information within a security domain with multiple Trusted Solaris hosts, Sun's naming service (NIS+) is used to distribute most configuration information.

Administering NIS+ is described in the base documentation in:

- *NIS+ Transition Guide*
- *NIS+ and DNS Setup and Configuration Guide*
- *NIS+ and FNS Administration Guide*
- *Name Services Administration Guide*
- *Name Services Configuration Guide*

Setting up the NIS+ master server and NIS+ clients is described in the *Trusted Solaris Installation and Configuration* manual.

This chapter describes the differences in managing NIS+ in a Trusted Solaris environment. This chapter includes the following major topics:

This chapter includes the following procedure.

Managing Multiple Trusted Solaris Hosts in a Security Domain

A Trusted Solaris 2.5.1 NIS+ master can be used to manage data for Trusted Solaris 2.x NIS+ clients or Solaris 2.x NIS+ clients. Hosts running the 1.x version of the Trusted Solaris operating environment cannot be clients of a Trusted Solaris 2.x NIS+ master because Trusted Solaris 1.x uses NIS, not NIS+. Trusted Solaris 2.x hosts cannot be clients of Solaris NIS+ masters.

A security domain is generally administered as a single NIS+ domain with a single NIS+ master. Multiple security domains may be administered together in a hierarchy of subdomains with multiple NIS+ non-root masters under a single NIS+ root master, which is the server at the top level of a hierarchy of NIS+ domains. There can be only one NIS+ root master. NIS+ replica servers may also be created to provide backup NIS+ query services; the replica is associated with a particular NIS+ master (root or non-root) and responds to NIS+ requests in the event that the primary master is unable to respond.

Configuration files that for one reason or another cannot be administered through NIS+ must be centrally administered and duplicated on individual hosts by other means.

Managing Standalone Trusted Solaris Hosts

Trusted Solaris hosts may or may not be connected to a network with hosts running other operating environments. A standalone Trusted Solaris host may either be configured as its own NIS+ master server or configured with no naming service. If a Trusted Solaris standalone host is configured without a naming service, the configuration information is maintained in `/etc`, `/etc/security`, and `/etc/security/tsol`. The administrative tools in the Trusted Solaris version of Solstice AdminSuite allow the administrative role to specify no naming service so that the information is stored locally.

NIS+ Constraints on Using the Root Role to Use Solstice System Administration Tools

Sites that wish to combine administrative tasks into a single administrative role similar to the root user (superuser) in standard UNIX systems cannot use the root role as the single unified role because:

- The root role cannot be added to the NIS+ admin group, and
- The root role cannot run Solstice from any other host than the NIS+ Master

Trusted Solaris version of the Solstice administration tools allow the system administrator and administrator administrative roles to update files on the NIS+ master. But because the root administrative role can only run Solstice when on the NIS+ master, a combined root role would be similarly constrained.

New Trusted Solaris NIS+ Tables and Files Not Administered by NIS+

New Trusted Solaris NIS+ tables are listed in the following table.

TABLE 12–1 New NIS+ Tables

Map Name	Definition
`tsoluser`	Stores Trusted Solaris attributes and execution profiles specified for user and role accounts
`tsolprof`	Stores execution profiles.
`tnrhdb`	Stores assignments between host and network addresses and templates in tnrhtp(4TSOL)
`tnrhtp`	Stores templates assigned to hosts and networks in tnrhdb(4TSOL)

Table 12–2 shows the configuration files that ordinarily must be the same on all machines and that are not administered through NIS+. These files can be automatically distributed only when a new NIS+ client is being set up using net install or Custom JumpStart installation from an install server, as described in *Trusted Solaris Installation and Configuration*. Changes to these files after the system is configured must be distributed administratively.

TABLE 12–2 Configuration Files Not Administered Through NIS+

File Name
label_encodings(4TSOL)
system(4)

Distributing these files during initial configuration of the Trusted Solaris environment is described in the *Trusted Solaris Installation and Configuration* manual. Distributing updated copies of these files if later changes are made is discussed in the *Trusted Solaris Label Administration* manual. See also Chapter 13," in this manual.

Adding Trusted NIS+ Tables

As in the base Solaris, administrator can add NIS+ tables with protected data fields. See the NIS+ administration manuals.

Caution - The protection features provided by NIS+ are not included in the Trusted Solaris access controls. To meet evaluation requirements, do not extend the default NIS+ tables or modify the access rules defined for table fields.

Adding a New Host and Giving It Credentials

The system administrator adds a new NIS+ client host using the Host Manager, which configures a host's relationship with the NIS+ master and sets up the host's credentials, while at the same time entering the host's name and IP address into the `hosts` database. The default Secure RPC password assigned to a new host by the Host Manager is *nisplus*.

The Database Manager should be used only to set up communications with a host that is not under local control, because the Database Manager only puts the host's name and IP address into the `hosts` database and does not do any additional configuration. For example, to allow users at a site to use `ftp` to download files from a `ftp` server at MIT, the system administrator would add the MIT host to the `hosts` database using the Database Manager. Since no one at the local site can log into the host, it would not be necessary to set its credentials.

NIS+-Related Procedures

▼ To Save NIS+ Tables and Restore Them After Reinstalling the Trusted Solaris Environment

1. **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+.

   Code Example 12–1 shows a script the system administrator can create to do the dumps and to create a list of group members for later re-creation of the groups table.

CODE EXAMPLE 12–1 nisscript for Dumping NIS+ Tables into ASCII Files *(of)*

```
#!/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 aliases bootparams ethers group hosts netgroup \
netmasks networks passwd protocols rpc services \
timezone tnrhdb tnrhtp tsolprof tsoluser 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
done
```

2. **Assume the root role and run the** `nisscript` **at ADMIN_LOW.**

3. **For each group, execute** `nisgrpadm -l` ***group_name*** **to list each of its members.**

```
$ nisgrpadm -l group_name
```

4. **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.**

5. **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

  .

  .

  .
```

6. **At the appropriate point in Chapter 5, "Configuring the NIS+ Root Master," in the *Trusted Solaris Installation and Configuration* manual, recreate the NIS+ environment.**

```
# nisserver -r -d domain-name.
```

   Make sure to include the final period (.) in the domain's name.

7. **As security administrator 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
```

8. **Re-create the NIS+ groups and add members manually from the list of group members created in the** `nisscript` **shown in .**

   There is no easy way to recreate the NIS+ groups automatically.

CHAPTER 13

Changing Configurable Trusted Solaris Kernel Switches and Window System Behavior

The settings of a number of Trusted Solaris kernel switches and the behavior of certain windows can be changed by a site's security administrator role. This chapter includes in the following major topics and procedures sections:

Behaviors Controlled by Configurable Trusted Solaris Kernel Switches Sites can set several Trusted Solaris kernel switches to control the following behaviors:

- Whether information labels are displayed at all on the system, and if so:
  - Whether information labels float
  - Whether information labels are reset to 0 upon exec
- Whether the names of files or subdirectories are displayed when their sensitivity labels strictly dominate the sensitivity label of the containing directory
- Allow the use of runpd(1MTSOL) to determine which privileges an application needs to run
- Whether attempts to shut down the system using the keyboard abort sequence are ignored

Needed Terms and Concepts

This section defines the customer-configurable kernel switches and provides definitions for other related terms.

Note - Even though the information label-related switches are initially set as described in this section and summarized in Table 13–1, the settings can be changed during installation. (See “How Kernel Switches Are Set and Changed ” on page 399.) The default settings of the switches that are not information label-related are overridden only by the editing of the system(4) file, as described in “To Change Kernel Switch Setting in the `/etc/system` File” on page 401.”

---

tsol_admin_high_to_cipso

This switch is not in the default version of the /etc/system file, but it can be added if needed. This switch must be set to 1 to enable communications with TSIX-type hosts that have the IP Label Field specified as CIPSO.

As explained in Chapter 10," when a tnrhtp template assigned to a destination host is specified with one of the CIPSO label indicators, the trusted networking software derives a CIPSO label from the message's sensitivity label and inserts the CIPSO label into the IP options portion of the message's packets. The ADMIN_HIGH sensitivity label is too big to map to a CIPSO label, so, by default, a message sent at the ADMIN_HIGH sensitivity label to a CIPSO-identified host is always dropped.

The security administrator can add the tsol_admin_high_to_cipso switch set equal to 1 in the /etc/system file. This causes the ADMIN_HIGH sensitivity 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.

tsol_enable_il

This is the master switch for enabling or disabling information labels. By default, tsol_enable_il= 0, and therefore information labels are not enabled. This switch allows information labels to be turned on.completely.

tsol_enable_il_floating

This switch is only looked at if tsol_enable_il= 1. This switch is provided for sites that want to use information labels but not allow them to float. If this variable is not enabled, the floating of information labels is disabled. by default, tsol_enable_il_floating= 0, and therefore information label floating is disabled.

tsol_reset_il_on_exec

This switch is only looked at if tsol_enable_il=1. If this switch is not set, the information label of an executed program is set to be the conjunction of both the information label of the calling process, the information label on the program file that is executed, and the information label of any other shared libraries that are read. If this switch is set, the information label of an executed program is reset to ADMIN_LOW at the beginning of execution—which assumes that the arguments and environment passed to the process are not to be labelled with the information label of the caller. When this switch is not set, information labels float even when a file or directory is merely listed, and for this and purely operational reasons, information labels have a tendency to quickly float to the highest information label

and thus lose their significance. This switch is set, and therefore information labels are reset to ADMIN_LOW by default.

Upgraded Names

Actions by users with the upgrade file sensitivity label authorization and by processes with the file_mac_write and file_upgrade_sl privileges can either create a new file or subdirectory or relabel an existing file or subdirectory at a sensitivity label that strictly dominates the sensitivity 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 *upgraded names*.

tsol_hide_upgraded_names

At sites that consider *upgraded names* to be sensitive information, the tsol_hide_upgraded_names kernel 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 getdents(2TSOL). 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.

tsol_privs_debug

Allows the administrative use of runpd(1MTSOL) to characterize a program‘s use of privilege. Requires additional setup; for the complete procedure, see Chapter 16” under “To Find Out Which Privileges an Application Needs” on page 530. After the application(s) have been privileged debugged, this variable should be reset and the machine rebooted. This switch is set to 0, and therefore privilege debugging is disabled by default.

audit_load

By default, auditing is enabled in the Trusted Solaris system, and this switch is set to 1. Turning off auditing is done by setting this switch to 0 and performing a number of other steps described in *Trusted Solaris Audit Administration* under “Audit Shutdown and Startup Procedures” in Chapter 2, “Auditing Setup.”

abort_enable

This feature, which was ported from the Solaris 2.6 operating environment, allows a site to choose whether to allow anyone to shut down the system using the keyboard abort sequence (usually “Stop (L1) A” or “Bread”). In the default Trusted Solaris `/etc/system` file, this switch is set to 0, and, therefore, systems can only be shut down by authorized users through the Shut Down option from the Trusted Path menu.

On hosts that are used by administrators for debugging, this switch can be enabled to allow access to the kadb(1M) kernel debugger at the monitor level.

How Kernel Switches Are Set and Changed

Table 13–1 summarizes information about each of the Trusted Solaris kernel switches that are available for setting by customers. The kernel switches are initially set to the defaults shown in Table 13–1 by the Trusted Solaris installation process.

TABLE 13–1 Default Kernel Switch Settings and Values Definitions

Switch Name	Value Definitions	Default Settings
`tsol_admin_high_to_cipso`	0 messages with ADMIN_HIGH SLs sent to TSIX CIPSO hosts are dropped 1 SL of ADMIN_ HIGH messages is mapped to a CIPSO-compatible label	This switch is implicitly set to 0, and it is not defined in the default system file. A site can add this switch, set to 1,to achieve the goals described in “`tsol_admin_high_to_cipso`” on page 397.
`tsol_enable_il`	0 ILs are disabled 1 ILs are enabled	`tsol_enable_il=1`
`tsol_enable__il_floating`	0 IL floating is disabled 1 IL floating is enabled	`tsol_enable_il_il_floating=0`
`tsol_hide_upgraded_names`	0 Show upgraded names 1 Hide upgraded names	`tsol_hide_upgraded_names=0`

TABLE 13–1 Default Kernel Switch Settings and Values Definitions *(continued)*

Switch Name	Value Definitions	Default Settings
`tsol_privs_debug`	0 Disable debugging using `runpd`(1MTSOL) 1 Enable debugging using `runpd`(1MTSOL)	`tsol_privs_debug=0`
`tsol_reset_il_on_exec`	0 Set IL to the conjunction of the ILs of the calling process, the exec'd process, and any shared libraries. 1 Reset IL to ADMIN_LOW upon exec.	`tsol_reset_il_on_exec=1`
`abort_enable`	0 Disable keyboard abort sequence 1 Enable keyboard abort sequence	`abort_enable=0`

During installation, the install team may accept or reset the default information label-related kernel switches in the Customize Trusted Solaris Configuration dialog box, shown in Figure 13–1. See also the *Trusted Solaris Installation and Configuration* manual. Any changes to the default settings are made in the kernel and recorded in the `/etc/system` file, which is then read when the system is booted.

After installation, any of the kernel switches shown in Table 13–1 may be changed by the security administrator, by modifying the system(4) file, as described in "To Change Kernel Switch Setting in the `/etc/system` File" on page 401.

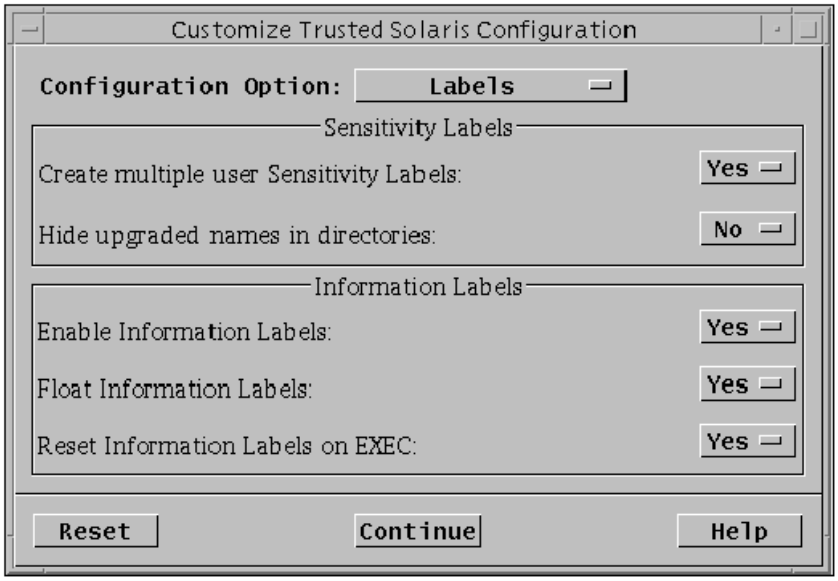


Figure 13–1 Label Configuration Defaults

▼ To Change Kernel Switch Setting in the `/etc/system` File

1. **As security administrator, use the Admin Editor action from the System_Admin folder in the Application Manager to open `/etc/system` for editing.**

   See "To Login and Assume an Administrative Role" on page 15 and "To Use the Admin Editor Action to Edit a File" on page 29, if needed.

2. **To set the variable that substitutes a CIPSO label for the ADMIN_HIGH sensitivity label, add `tsol_admin_high_to_cipso=1` (on).**

3. **Configure information labels and their behavior.**

   a. **Turn on or off the variable for enabling information labels, find `tsol_enable_il=` and set the value to `1` (on) or `0` (off).**

      Because the switches for information label floating and for the reset of the information label when a new program is executed are looked at only when the switch shown in Step 3 on page 401 is set to 1, do either of Step 3 on page 401 or Step 3 on page 401 only if you have enabled information labels.

   b. **To turn on or off the variable for enabling information label floating, find `tsol_enable_il_floating=` and set the value to `1` (on) or `0` (off).**

   c. **To turn on or off the variable for resetting information labels on `exec`, find `tsol_reset_il_on_exec=` and set the value to `1` (on) or `0` (off).**

4. **To turn on or off the variable for hiding the names of files whose sensitivity labels have been upgraded, find** `tsol_hide_upgraded_names=` **and set the value to** `1` **(on) or** `0` **(off).**

5. **To turn on or off the variable for allowing privilege debugging, find** `tsol_privs_debug=` **and set the value to** `1` **(on) or** `0` **(off).**

6. **To turn on or off the variable that loads the audit system call module, find** `audit_load=` **and set the value to** `1` **(on) or** `0` **(off).**

7. **To turn on or off the variable for allowing the keyboard abort sequence, find** `abort_enable=` **and set the value to** `1` **(on) or** `0` **(off).**

8. **Write and quit the file.**

```
:wq
```

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

```
> boot
```

10. **To distribute the changes to all hosts in the distributed system, follow the steps in "Distributing Changed Kernel Switch Settings to Hosts Across the Network" on page 402 of *Trusted Solaris Label Administration***

    The security administrator can set up the boot server so that any changes to the `/etc/system` file are passed to each host in the distributed system from a file consulted at boot.

Distributing Changed Kernel Switch Settings to Hosts Across the Network

Modifications seldom need to be made to the system file after a site has been installed and configured, except for changing the switch that enables privilege debugging, and enabling privilege debugging is usually done on a single host. If need be, the security administrator can use the rdist(1) command to automatically distribute identical copies of the file to all machines in the distributed system. See Chapter 2," under "To Remotely Distribute Configuration Files" on page 40.

Modifying the Front Panel and Workspace Menu

The *CDE Advanced User's Guide and System Administrator's Guide* describes how to modify the Front Panel and Workspace Menu. Much of the information in those manuals also apply in Trusted Solaris environments. Exceptions are discussed in this section.

On Trusted Solaris systems, ordinary users cannot make modifications to the configuration files that change the following menus on the Front Panel:

- Trusted path menu or
- Subpanel menus

On Trusted Solaris systems, ordinary users can:

- Modify personal copies of files that configure the Workspace (root) menu
- Drag and drop actions onto the Front Panel without security risks and therefore without restrictions

In general, actions and executables invoked in most of the Front Panel and Workspace-related menus are limited by your profile, UID and label.

The use of privileges by commands and actions and their UID, GID, and sensitivity label are controlled through the profile mechanism. Whether something works or not depends on your profile. For example, the Solstice applications actions in the Solstice_Apps folder within the Application Manager are executable only when the account using the current workspace has the Solstice actions in its profile with the needed privileges.

Trusted Desktop items and the Exit button run as root with all privileges because they are session wide. The rest of the Front Panel, its Subpanels, and the Workspace menu run with the same UID as and the sensitivity label of the current workspace. Table 13–2 summarizes the main differences that apply to modifying the workspace under Trusted Solaris.

TABLE 13–2 Differences in Modifying the Workspace Under Trusted Solaris Restraints

Base CDE	Trusted Solaris
Any user can copy the system-wide version of the `dtwmrc`(4) file into `$HOME/.dt` and modify the workspace menu to bring up actions and invoke commands.	Same as base CDE. If commands require privileges, the privileges are only available by inheritance. The command needs to be specified with the needed privileges in a profile. The account needs: 1, the profile shell as its default (login) shell; 2, a profile specified with the privileged command.
Any action or command invoked through the workspace menu or from the Front Panel runs with the UID of the invoking user.	Any action or command invoked through the workspace menu or from the Front Panel runs with the UID of the invoking user within the profiles of the user. Actions and commands can use privileges and other attributes specified for them in the account's profile. commands can run with privileges and other attributes specified in a profile if the account's default shell is the profile shell and the profile applies to the current user

Modifying the Front Panel

Only the security administrator can modify 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, one to create an alternate mail application that can run with privilege in the Front Panel and one to add an administrative action that can run with inherited privileges to the System_Admin folder for the purpose of editing another configuration file. See "To Create a New Administrative Action for Editing an Administrative File" on page 30 of Chapter 1 and "Substituting an Alternate Mail Application" on page 187 of Chapter 6.

Modifying the Workspace (root) Menu

The workspace menu is the menu that is accessed by clicking and holding the right mouse button on the background of the workspace.For example, the default Workspace menu has the options shown in the following figure.

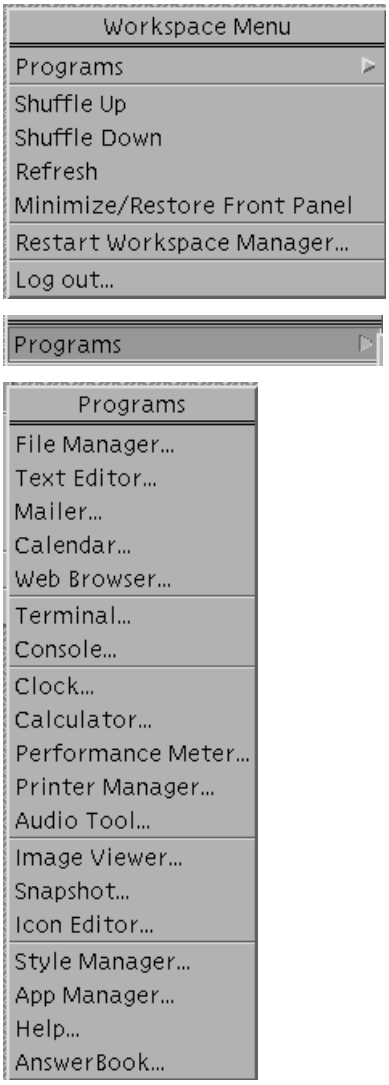


Figure 13–2 Default Workspace Menu and Programs Sub-menu

The administrator role can change the workspace menu for all accounts by copying `/usr/dt/config/C/sys.dtwmrc` to `/etc/dt/config/C/` and making modifications to the copied file. Any version of the `dtwmrc` in an account's $HOME`/.dt` directory can override the system-wide workspace root menu for its own normal user sessions. The modified `dtwmrc` file must be in the $HOME`/.dt` directory in the SLD at the account's *session clearance* for it to apply to any workspace at any sensitivity label.

To set up modified `dtwmrc` files to be copied to individual user's $HOME directories, the administrator role can specify a skeleton directory that contains both a modified `dtwmrc` and a `copy_files` file that lists the `dtwmrc`. Both the modified

`dtwmrc` and `copy_files` file will be automatically copied into the $HOME directory SLD at each account's minimum sensitivity label, and propagated into other $HOME directory SLDs that are subsequently created. However, a modified `dtwmrc` file needs to be in effect at the session clearance for it to apply to any workspace at any sensitivity label. Unfortunately, using the skeleton path method does not ensure that the dtwmrc will be copied into the home directory SLDs that correspond to each account's session clearances.

When an account is able to work at only a single sensitivity label, the session clearance always equals the minimum sensitivity label. For example, for an account whose minimum and maximum sensitivity labels are defined as PUBLIC, the only possible session clearance is PUBLIC.

When an account is allowed to use multiple labels, the account may only choose one session clearance. For this type of account, as for the account of a single-label user, only one $HOME`/.dt/dtwmc` needs to be created in the SLD that corresponds to the minimum sensitivity label SLD (.SLD.0) for the account.

For accounts that are able to work a multiple sensitivity label, creating a `dtwmrc` file at the session clearance requires some thought, because these accounts have multiple options when selecting a session clearance after login.

The possible methods to use for each of the two following cases are described below:

- For an account on a newly installed system
- For an account that has already been working on the system

Modifying `dtwmrc` for an Account on a Newly-installed System

Before any home directory SLDs have been created with sensitivity labels that equal any of the account's session clearances, the administrator or the user can do the following in the SLD in $HOME`/.MLD.`*username* whose sensitivity label is equivalent to the account's minimum sensitivity label.

- Create a `.dt/dtwmrc` file
- Put the pathname of the `dtwmrc` file into the `.copy_files` file
- Create and relabel a workspaces at each of the possible session clearances
- Restart the workspace manager

  This ensures that `.copy_files` copies the modified `dtwmrc` file to each home directory SLD as the workspace at the corresponding sensitivity label is created. The copy only occurs when an SLD is first created. See "To Modify the Workspace Menu (Method 1)" on page 407 if the account has never been logged into.

Modifying `dtwmrc` *for an account that has already been working on the system*

If any workspaces have been created with sensitivity labels that equal any of the account's session clearances, SLDS with the corresponding sensitivity labels already have been created with sensitivity labels. The administrator or the user can do the following.

Do the following in the account's home directory MLD.

- Find out the adorned SLD names of every SLD whose sensitivity label is equivalent to a session clearance at which the account works
- Go to the .dt subdirectory in a SLD whose sensitivity label is equivalent to a session clearance
- Create a `.dt/dtwmrc` file and modify it as desired

  If more than one session clearance is possible, go on and do the following.

- Copy the modified `dtwmrc` file to `/tmp`
- Go to the .dt subdirectory in another SLD whose sensitivity label is equivalent to a session clearance
- Copy the /tmp/dtwmrc file from /tmp, using the adorned name of the SLD.

  To put the changes into effect:

- Restart the workspace manager

  See "To Modify the Workspace Menu (Method 2)" on page 409, if SLDs already have been created with sensitivity labels that are equal to the session clearances available to the account.

▼ To Modify the Workspace Menu (Method 1)

Do this method before any home directory SLDs have been created with sensitivity labels that equal any possible session clearances. See "Modifying the Front Panel and Workspace Menu" on page 403.

1. **Select a session clearance at the account's minimum sensitivity label.**

   In the example, the account has a minimum sensitivity label of PUBLIC, so the user or administrator selects a session clearance of PUBLIC.

2. **In the initial workspace created at the account's minimum sensitivity label, change to the home directory `.dt` directory.**

   .

```
trusted11% cd /home/roseanne/.dt
```

3. **Copy the `/usr/dt/config/C/sys.dtwmrc` file to the $HOME/`.dt` directory.**

```
trusted15%  cp /usr/dt/config/C/sys.dtwmrc dtwmrc
```

4. **Use your favorite text editor to edit the** `dtwmrc` **file and change the Workspace Menu as desired.**

```
trusted16%  vi dtwmrc

   . . .

   ''Workspace Menu''   f.title

       ''Bug Menu''    f.menu BugMenu

       ''FrameMaker''                    f.menu FrameMenu

       ''Games Menu''   f.menu GamesMenu

       ''Sun Local Applications''  f.menu LocalMenu

       ''Programs''    f.menu ProgMenu

       ''Shuffle Up''   f.circle_up

       ''Shuffle Down''   f.circle_down

       ''Refresh''    f.refresh

       ''Minimize/Restore Front Panel'' f.toggle_frontpanel

        no-label    f.separator

       ''Restart Workspace Manager...'' f.restart

        no-label    f.separator

       ''Log out...''   f.action ExitSession

   }


   Menu BugMenu

   {

       ''Bug Creating/Changing Tool'' f.exec /usr/dist/exe/bugtraq

    ''Bug Finding Tool''  f.exec /usr/dist/exe/bugfinder

   }

   . . .
```

The example above adds a number of new menus to the Workspace Menu definition and shows the beginning of the definition for the first menu called Bug Menu.

5. **Use your favorite text editor to open the $HOME/**`copy_files` **file and add the pathname of the** `dtwmrc` **file.**

```
trusted17%  cd /home/roseanne
   trusted18%  vi copy_files
   /home/roseanne/bin/dtwmrc
```

6. **For each session clearance available to the account, create a workspace whose sensitivity label is the same as one of the account's session clearances.**

   For example, to create the first `bin/dtwmrc` file, select Add Workspace and Change Workspace SL from the TP menu, changing the sensitivity label to equal one of the account's session clearances.

7. **To put the changed menu into effect, choose Restart Workspace Manager from the Workspace Manager.**

   The new menu will go into effect only if the `dtwmrc` is in the SLD corresponding to the current session clearance.

8. **After the `dtwmrc` is modified in the home directory SLD of the user's session clearance, a new menu displays.**

   The screen example shows the menu with new options.

▼ To Modify the Workspace Menu (Method 2)

Do this method if SLDs already have been created with sensitivity labels that are equal to the session clearances available to the account.

1. **Go to a workspace at the account's maximum sensitivity label, change to the home directory MLD, and find out the SLD number for the SLDs whose sensitivity labels are equal to the session clearances available to the account.**

In the example, the account has a label range from PUBLIC to INTERNAL_USE_ONLY, so the account can choose a session clearance of either PUBLIC or INTERNAL_USE_ONLY. The following commands are entered in a workspace at the account's maximum sensitivity label, INTERNAL_USE_ONLY.

```
trusted11% cd /home/.MLD.roseanne
   trusted12% ls -la
   ./   ../  .SLD.0/  .SLD.1/ .SLD.2

   trusted13% getlabel .*
   . . .

   .SLD.0:   ADMIN_LOW [PUBLIC]

   .SLD.1:   ADMIN_LOW [INTERNAL]
```

The result of `getlabel` shows that there are two SLD directories where a copy of the `dtwmrc` file should be installed, `.SLD.0` (for PUBLIC) and `.SLD.1` (for INTERNAL_USE_ONLY).

2. **Change to a workspace whose sensitivity label is the same as one of the account's session clearances.**

   For example, to create the first `bin/dtwmrc` file, change to the workspace whose sensitivity is PUBLIC.

3. **Change to the `.dt` directory in the home directory SLD whose sensitivity label is the same as the current workspace.**

   Because the code example in shows that the name for the PUBLIC SLD is `.SLD.0`, the following example shows the directory being changed to `.SLD.0/.dt`.

```
trusted14% cd /home/.MLD.roseanne/.SLD.0/.dt
```

4. **Copy the `/usr/dt/config/C/sys.dtwmrc` file to the $HOME/`.dt` directory.**

```
trusted15%  cp /usr/dt/config/C/sys.dtwmrc dtwmrc
```

5. **Use your favorite text editor to edit the `dtwmrc` file and change the Workspace Menu as desired.**

```
trusted16%  vi dtwmrc

   . . .

   ''Workspace Menu''   f.title

       ''Bug Menu''    f.menu BugMenu

       ''FrameMaker''                    f.menu FrameMenu

       ''Games Menu''   f.menu GamesMenu
```

```
    ''Sun Local Applications''  f.menu LocalMenu
    ''Programs''    f.menu ProgMenu
    ''Shuffle Up''   f.circle_up
    ''Shuffle Down''   f.circle_down
    ''Refresh''    f.refresh
    ''Minimize/Restore Front Panel'' f.toggle_frontpanel
     no-label    f.separator
    ''Restart Workspace Manager...'' f.restart
     no-label    f.separator
    ''Log out...''   f.action ExitSession
}


Menu BugMenu
{
    ''Bug Creating/Changing Tool'' f.exec /usr/dist/exe/bugtraq
 ''Bug Finding Tool''  f.exec /usr/dist/exe/bugfinder
}
. . .
```

The example above adds a number of new menus to the Workspace Menu definition and shows the beginning of the definition for the first, Bug Menu.

6. **To put the changed menu into effect, choose Restart Workspace Manager from the Workspace Manager.**

   The new menu will go into effect only if the `dtmrc` is in the SLD corresponding to the current session clearance.

7. **To continue creating one or more `dtwmrc` files for an account with multiple session clearances, copy the modified `dtwmrc` to `/tmp` and continue with Step 8 on page 412 and following steps. If there is only one possible session clearance, go to Step 12 on page 412.**

```
trusted14% cp dtwmrc /tmp
```

8. **Change to a workspace whose sensitivity label is the same as another one of the account's session clearances.**

   To create the second `bin/dtwmrc` file in our example, change to the workspace whose sensitivity label is INTERNAL_USE_ONLY.

9. **Change to the `.dt` directory in the home directory SLD whose sensitivity label is the same as the current workspace.**

   The code example in above shows that the name for the INTERNAL_USE_ONLY SLD is .SLD.1

```
trusted15% cd /home/.MLD.roseanne/.SLD.1/.dt
```

10. **Copy the `dtwmrc` from the correct SLD in the `/tmp` MLD, which has the same SLD name as the home directory SLD where the last change was made.**

    In our example, the `dtwmrc` was created in `.SLD.1`, so the `dtwmrc` is copied from `/.MLD.tmp/.SLD.1`.

```
trusted16%  cp /.MLD.tmp/.SLD.1/dtwmrc .
```

11. **If any SLDs remain that correspond to session clearances for the account, repeat Step 7 on page 411 through Step 10 on page 412.**

12. **To put the changed menu into effect, choose Restart Workspace Manager from the Workspace Manager.**

    The new menu will go into effect only if the `dtmrc` is in the SLD corresponding to the current session clearance.

Configuring the Rules for Upgrades and Downgrades

The three following types of operations in the window system may result in an upgrade or downgrade of either or both of the sensitivity label and information label:

- Cut and paste
- Copy and paste
- Drag and drop

The `/usr/dt/config/sel_config` file is consulted to determine:

- Whether certain types of operation should be automatically confirmed or

- Whether a selection confirmer dialog should be displayed.

  Figure 13–3 shows the selection confirmer for drag and drop operations between File Managers. Other selection confirmers display for cut and paste and copy and paste operations between File Managers and between windows at varying labels.

  The `sel_config` file also specifies the following:

- For each type of operation whether the information label of the destination is floated by the information label of the source
- What to do with an invalid information label when information labels are configured to be hidden for a specific user
- A list of selection types to which automatic replies are given

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

  This section gives the needed background and describes the fields in the file for the system administrator who may wish to change the defaults.

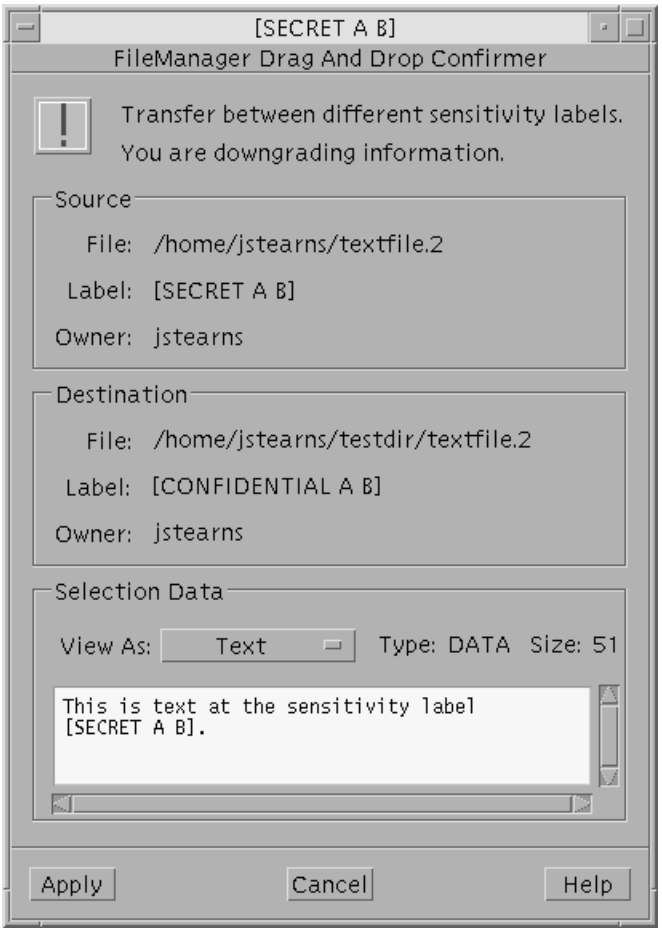


Figure 13–3 File Manager Selection Confirmer

Review of Selection Management Concepts

By default, normal users can perform cut and paste, copy and paste, and drag and drop operations of the following types *as long as the source and destination have the same sensitivity label and have the same user ID*:

- Files between File Managers
- Selections between windows

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 same or differing owners and the same or differing sensitivity labels

- *Cut and paste* and c*opy and paste* of *selections* between *windows* having the same or differing owners and the same or differing sensitivity labels

Note - *Drag and drop* of *selections* always requires equality of sensitivity labels and ownership.

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

TABLE 13–3 Rules and Authorizations Required for File Manager Copy and Paste, Cut and Paste and Drag and Drop

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

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

TABLE 13–4 Rules and Authorizations Required for Copy and Paste, Cut and Paste and Drag and Drop of

TABLE 13–4 Rules and Authorizations Required for Copy and Paste, Cut and Paste and Drag and Drop of Selections Between Windows *(continued)*

Selections Between Windows

Transaction Description	SL Relationship	Owner Relationship	Authorizations
Copy and paste, or cut and paste, of selections between windows	Same SL	Same UID	None required
	Downgrade	Same UID	Paste to a downgraded window
	Upgrade	Same UID	Paste to an upgraded window
	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 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 of selections between windows, which are mediated by the `/usr/dt/bin/sel_mgr` application.

Note - Because drag and drop of selec*tions* always requires equality of sensitivity labels and ownership, the rules in the `sel_config` file do not apply to drag and drop of selections.

The `sel_config` file has three sections described below:

- Automatic confirmation
- Hidden information label action

- Automatic reply

Automatic Confirmation Section

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

TABLE 13–5 Format of Automatic Confirmation Section in `sel_config`

Transfer Type	**Automatically confirm?** **n= display the selection confirmer**	**IL From?** **s= Use the destination IL** **d= float the destination IL with the source IL**
SL-relation.IL-relation:	<space>y | n,	<space>s | d

SL-relation refers to the relationship between the SL of the source and the SL of the destination. *IL-relation* refers to the relationship between the IL of the source and the IL of the destination. *The relationship* can be one of:

```
upgrade|downgrade|equal|disjoint
```

The relationship compares the label of the source with the label of the destination:

upgrade	source label < destination label
downgrade	source label > destination label
equal	source label = destination label
disjoint	source label <> destination label

For example, see the following:

```
downgradesl.downgradeil: n,s
```

The example above means that when an operation results in a downgrade of the sensitivity label and a downgrade of the information label:

- The selection confirmer dialog displays for user confirmation and
- The information label of the source floats the information label of the destination.

Hidden Information Label Action Section

The hidden_il_action field specifies how to handle an invalid information label when information labels are hidden for a user.

If automatic confirmation is specified for the operation, the setting for "IL From" in the automatic confirmation section (see Table 13–5) is used if it results in a valid information label. If using the "IL From" setting produces an invalid information label, the hidden_il_action setting is used instead.

When a manual confirmation is specified for an operation, and if the information label would be invalid for the destination, the hidden_il_action setting is used.

If the information label is still invalid after the hidden_il_action setting is used, the transfer is not allowed and an error is logged.

The following table defines the options for the hidden IL actions.

TABLE 13–6 hidden_il_action = Values and Definitions

Value	Action	Application
c	display a confirmer with ILs shown	dtfile and sel_mgr
d	use IL of destination window	sel_mgr
	use IL of destination directory	dtfile
l	set IL to ADMIN_LOW	dtfile and sel_mgr

By default, the hidden IL action is to display a confirmer showing the information labels:

```
hidden_il_action: c
```

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 for yes, 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.

Default `sel_config` Settings

The comments and the default settings in the file `/usr/dt/config/sel_config` are shown in the following table.

CODE EXAMPLE 13–1 `sel_config` Selection Configuration File *(of)*

```
#pragma ident ‘‘@(#)sel_config 5.6 98/03/17 SMI; TSOL 2.x’’

# Copyright (c) 1998 by Sun Microsystems, Inc.

#  All rights reserved.

# Auto settings default file

# This file controls the action of the selection

# manager and the file manager drag and drop confirmers.

# There is an entry for the 16 possible types of transfer based

# upon the source label and destination label.

# The file has three sections, the auto confirm section, the

# hidden information label action section and the auto reply section.

# Auto Confirm Section

#   Specifies for each transfer type whether the action should

#   be automatically or manually confirmed.

# Auto confirm?     y -> confirm transfer without displaying confirmer

#                        (note: user must still have proper authorizations)

#                   n -> display confirmer before processing transfer

# IL From = specifies the IL to use if auto_confirm is true.

#                   ‘s’ = use IL of source data (sel_mgr)

#                         use IL of source file (dtfile)
```

```
#                   'd' = use IL of destination window (sel_mgr)
#                         use IL of destination directory (dtfile)
#                             Auto     IL
#  Transfer Type             Confirm?  From
downgradesl.downgradeil: n,  s
downgradesl.equalil:  n, s
downgradesl.upgradeil:  n, s
downgradesl.disjointil:  n, s
equalsl.downgradeil:  n, s
equalsl.equalil:  y, s
equalsl.upgradeil:   y, s
equalsl.disjointil:   n, s
upgradesl.downgradeil:   n, s
upgradesl.equalil:   n, s
upgradesl.upgradeil:   n, s
upgradesl.disjointil:   n, s
disjointsl.downgradeil:  n, s
disjointsl.equalil:   n, s
disjointsl.upgradeil:   n, s
disjointsl.disjointil:  n, s
# Hidden IL action Section
#   If ILs are being hidden, it is possible that the IL can be
#   invalid but would not be displayed for the user to correct.
#   The hidden_il_action field specifies the action to be taken
#   in that case.
#   If ILs are being hidden, the IL will be set as follows:
#   If an auto-confirm is being processed the auto confirm ''IL From''
#   setting will be used if it results in an IL that is valid.  If that
#   produces an IL that is invalid, the hidden_il_action setting will
#   be used.
```

```
#   If it is not an auto-confirm and the IL would be invalid
#   for the destination, the hidden_il_action setting will be used to
#   set the hidden IL.  If the IL is still invalid after the hidden_il_action
#   setting is used, the transfer will not be allowed and an error will be
#   logged.
#   hidden_il_action =
#                   'c' = display a confirmer with ILs shown
#                         (dtfile and sel_mgr)
#                   'd' = use IL of destination window (sel_mgr)
#                         use IL of destination directory (dtfile)
#                   'l' = set IL to admin_low (sel_mgr and dtfile)
hidden_il_action: c
# Auto Reply Section
autoreply: y
replytype: TARGETS
replytype: Pixel Sets
replytype: LENGTH
replytype: Type Of Monitor
```

▼ 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`**(1MTSOL)**

   See "Configuring the Rules for Upgrades and Downgrades " on page 412" 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.

Configurable Window Settings

Window behavior in the Trusted Solaris operating environment can be modified by changing the settings in these files:

- `/usr/dt/bin/Xsession` – This script starts the session and window managers. It sets the font path and other session-wide default values.
- `/usr/dt/bin/Xtsolusersession` – This script establishes the context for starting applications in a workspace. For example, the script contains lines that source the $HOME`/.dtprofile` for a user with any login shell except the `pfsh` while not sourcing `.dtprofile` for an account whose login shell is the `pfsh`. A site's security administrator can change this behavior by modifying the script.
- `/usr/openwin/server/tsol/config.privs` – The purpose of this file is to relax the security for specific security checks. Security checks are not enforced for those privileges contained in this file. For example, including win_fontpath in the file relaxes the restriction on loading fonts.
- `/usr/openwin/server/tsol/property.atoms` – The property atoms specified in this file are not polyinstantiated.
- `/usr/openwin/server/tsol/public.atoms` – The atoms specified in this file can be accessed by the XGetAtomName routine.
- `/usr/openwin/server/tsol/selection.atoms` – The selection atoms in this file are polyinstantiated.

CHAPTER 14

Managing Printing

Standard Solaris print utilities and databases and the Solstice Printer Manager have been modified to meet Trusted Solaris requirements. The system administrator and the security administrator role both share printer administration duties. Both administrators need an understanding of printer administration concepts, as described in the Solaris *System Administration Guide*, and of how to use the Solstice Print Manager, as described in the *Solstice AdminSuite 2.3 Print Administration Guide*.

This chapter provides background information on the following topics for understanding and managing the unique aspects of printing in the Trusted Solaris environment:

This chapter also provides the following procedures:

Needed Terms

Banner / Trailer Pages

In the base Solaris printing system, jobs print with an optional banner page that contains information on, for example:

- Who submitted the job
- The time of printing, and
- The name of the host from which the job was submitted

In the Trusted Solaris system, jobs print with a trailer page paired with a banner page. The banner and trailer pages contain additional security-related information about the printed output. The print service guarantees the content and accuracy of the information contained in the banner / trailer pages. Users cannot suppress the printing of banner and trailer pages unless the security administrator grants them a special authorization.

Body Pages

The body pages for a print job are the pages that contain the data submitted by a user for printing. In the Trusted Solaris system, the body pages are printed with labels at the top and bottom of each page. Users cannot suppress the labels on body pages unless the security administrator grants them a special authorization.

Information Labeling and Access Control for Printers

By default, the Trusted Solaris printing subsystem provides the following features:

- Banner / trailer pages on all print jobs have the print job's sensitivity label, information label, and a unique job ID, along with site-specified special handling instructions that are based on the label of the print job
- The print job's information label is printed on the top and bottom of body pages
- Security-sensitive information about the print job and the print data itself are protected
- Status information provided to users is controlled by MAC and DAC
- Each printer can be configured with a restricted sensitivity label range to control which jobs it prints

The banner and trailer pages and the printing of labels at the top and bottom of body pages can be suppressed by command line options used by authorized user or role accounts or by administrative action to turn off these features for everyone.

The system's MAC and DAC policies are enforced upon the data contained in each print request. Enforcement is from the point at which the print request is submitted until the requested data has been printed on a physical page. Attempts to subvert or circumvent the protection provisions are detectable and generate an audit trail.

The handling of printer output is controlled by each site according to its policy and procedures.

All labeling of printer output from hosts running the Trusted Solaris operating system is *automatically done* according to the site's requirements. Access to printers is controlled by comparing the label of the print job to the label range of the printer.

Assigning Labels to Print Jobs

Each print job is automatically assigned a sensitivity label that corresponds to the sensitivity label at which the user is working. Figure 14–1 shows an employee reading email at a sensitivity label of `INTERNAL_USE_ONLY`. When he sends the email to the printer by selecting the Print option from the Message menu, the print job is automatically assigned the sensitivity label `INTERNAL_USE_ONLY`.

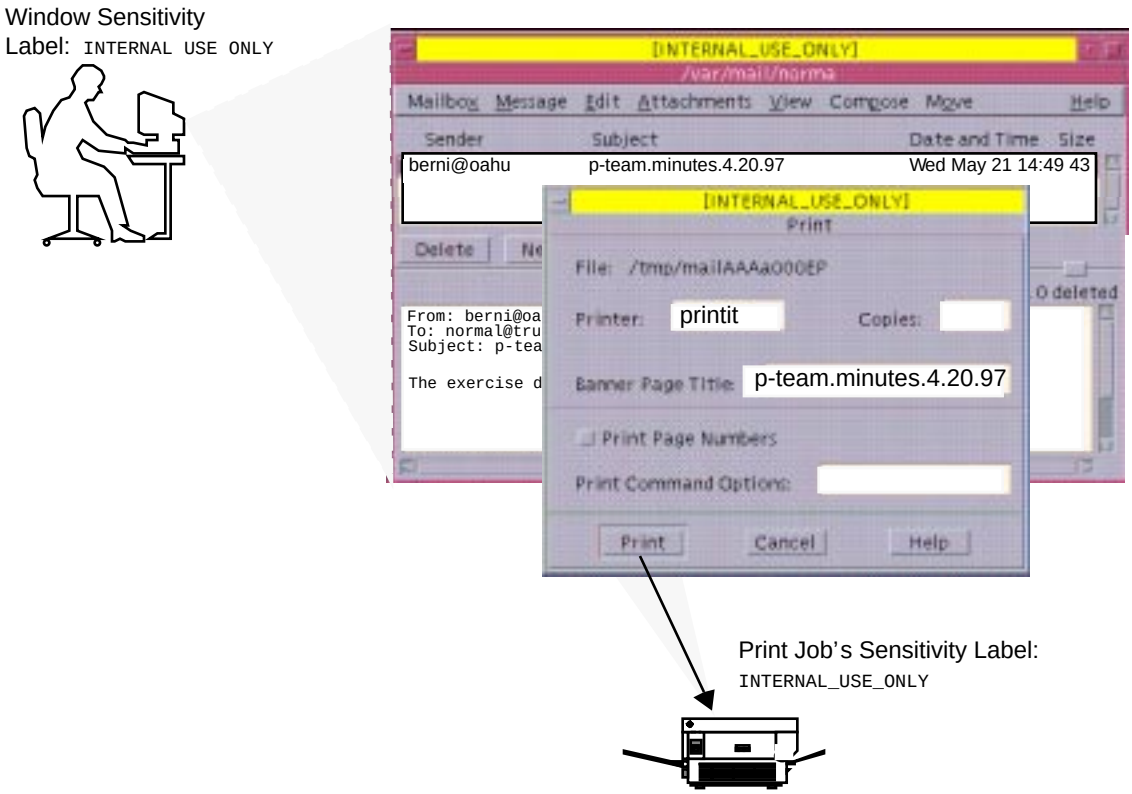


Figure 14–1 Automatic Labeling of Print Jobs

Using a Label Range on Printers to Control Which Jobs Can Print

Printers may be configured to print only jobs whose labels are within a restricted label range. Email notifies the sender when a job does not go through.

For example, the legal department's printer is set up to print jobs that are sent at any of the following three labels:

- NEED_TO_KNOW LEGAL
- INTERNAL_USE_ONLY, and
- PUBLIC

The legal department's printer set up as described above would reject jobs at any other sensitivity label. See Figure 14–2.

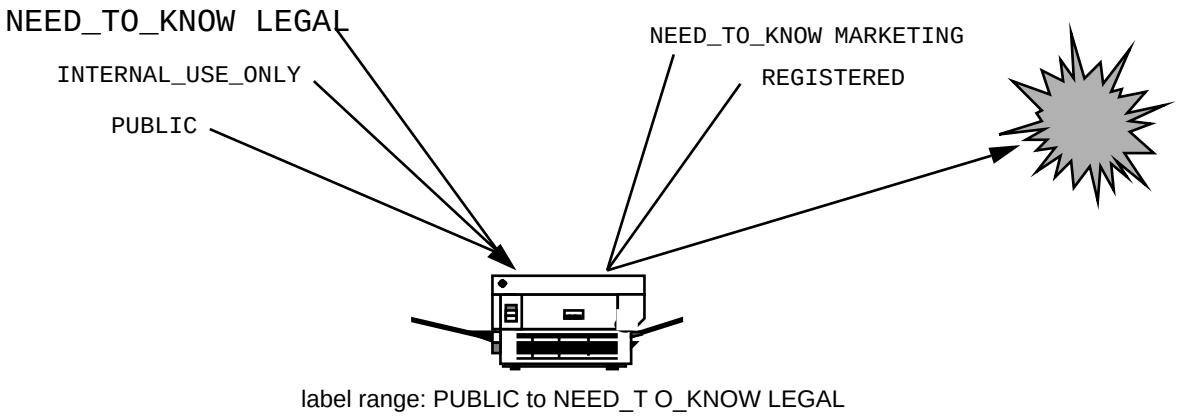


Figure 14–2 Example of a Printer with a Restricted Label Range

The setting of a restricted label range for a printer is done by the administrator using the Device Allocation Manager, as described in:

- "To Configure a Restricted Label Range for a Printer" on page 444

Printing of Labels on Printer Output

This section describes:

- Labels printed on body pages
- Labels and other information printed on banner and trailer pages
- How the default labels can be changed

Which labels print are determined by whether information labels have been disabled. (See Chapter 13," for how information labels are enabled or disabled in the system(4) file by setting `tsol_enable_il=` to `1` or `0`.)

As described in Chapter 5," the security administrator specifies for each account whether information labels and sensitivity labels are hidden or not. The individual accounts's settings to hide or show sensitivity labels or information labels *do not* affect what labels are printed. So, for example, if information labels are enabled and an account has the hide ILs setting, an information label is still printed where appropriate.

The security administrator can override any of the default labels for printer output by editing the `tsol_separator.ps` file. "Changing the Default Labels on Body Pages" on page 429 describes how to change which label prints by modifying a file on the print server.

Labels Printed on Body Pages

By default, all jobs are printed with the *information label* of the job.at the top and bottom of every body page. Figure 14–3 shows a job whose information label is PUBLIC and sensitivity label is INTERNAL_USE_ONLY. The job is printed with the PUBLIC information label.

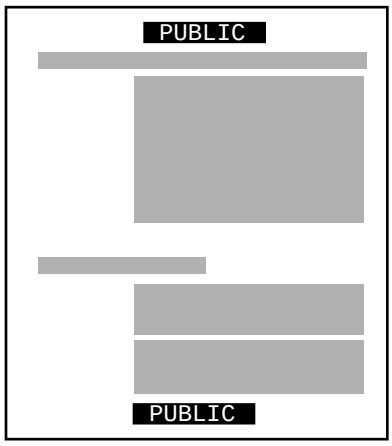


Figure 14–3 Information Label Automatically Printed by Default on a Body Page

If information labels are disabled in the system file (`tsol_enable_il=0`), the job's sensitivity label is printed instead of its information label. The following figure shows a print job with the PUBLIC[INTERNAL_USE_ONLY} label when information labels are disabled. Instead of the job's information label, the sensitivity label INTERNAL_USE_ONLY is printed.

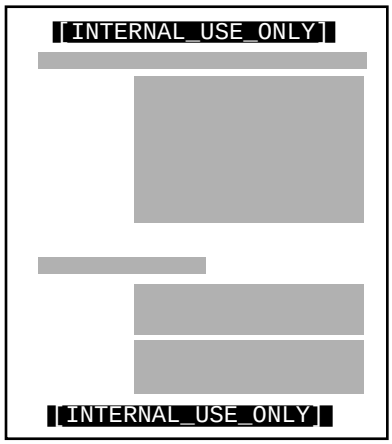


Figure 14–4 Sensitivity Label Printed on Body Pages when Information Labels Are Disabled

Changing the Default Labels on Body Pages

The security administrator can change the default for the printing of labels on body pages in the following ways:

- Give one or more users an authorization to allow them to print jobs without labels on the body pages
- Change a configuration file on the print server so that:
  - Labels are not printed on body pages for any users
  - Some labels other than information labels are printed on body pages for all users

See "`/usr/lib/lp/postscript/tsol_separator.ps`" on page 431. Also see these related procedures:

- "To Allow Some Users to Print Jobs Without Banners and Trailers " on page 449
- "To Suppress the Printing of Page Labels on All Print Jobs" on page 451

Labels, Job Numbers, and Handling Information on Banner and Trailer Pages

This section describes the types of labels and text that are printed by default on the banner and trailer pages. Figure 14–5 shows a banner page. Figure 14–6 shows the differences on the trailer page. A thick black line appears instead of a gray frame, and the page type identifier changes from JOB START to JOB END.

Changing the Default Labels and Warnings on Print Jobs

All the text and the labels and warnings that appear on print jobs are site-configurable. The labels that appear on the print job itself and the information shown in Figure 14–5, Figure 14–5," which appears on both banner and trailer pages, are configured in two places:

- label_encodings(4TSOL)
- `/usr/lib/lp/postscript/tsol_separator.ps`

`label_encodings`(4TSOL)

The following portions of printer trailer and banner pages are configured in the label_encodings(4TSOL):

- The “protect as classification,”
- The “access-related” words
- The handling instructions specified in the PRINTER BANNERS line(s)
- The handling instructions specified in the CHANNELS line(s)

See Chapter 3, “Specifying Labels and Handling Guidelines for Printer Output,” in the *Trusted Solaris Label Administration* manual for how to configure these portions of the banner and trailer pages.

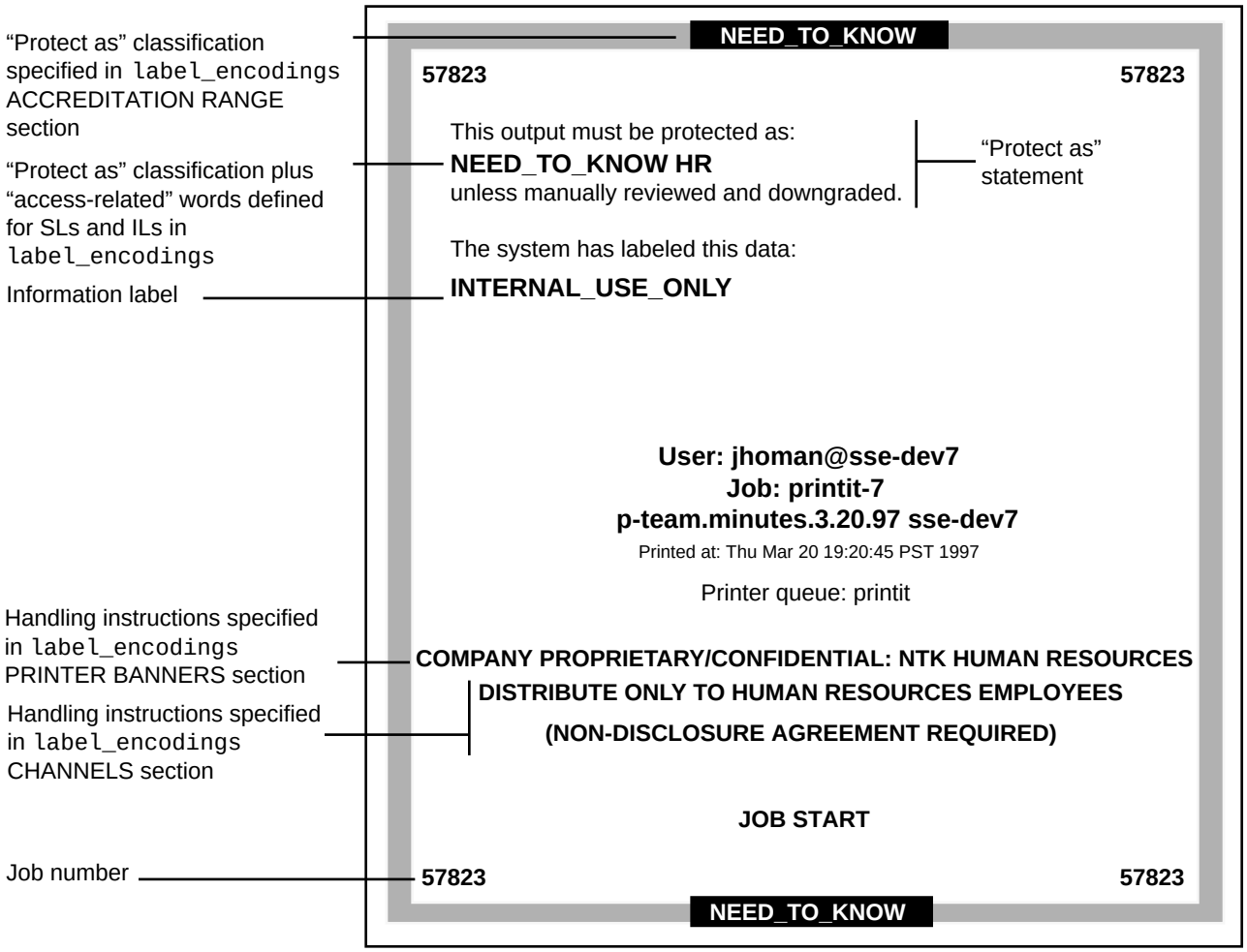


Figure 14–5 Typical Print Job Banner Page

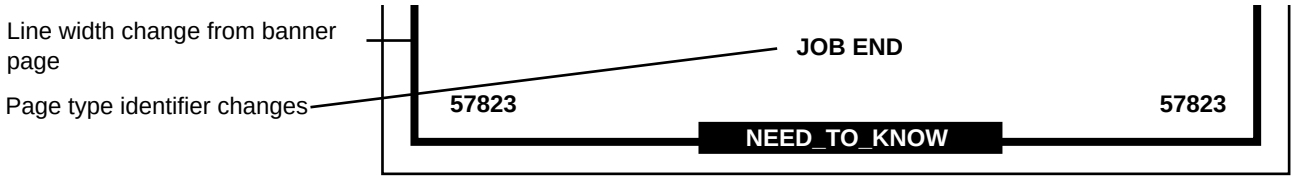


Figure 14–6 Differences on a Trailer Page

/usr/lib/lp/postscript/tsol_separator.ps

The security administrator may modify the `tsol_separator.ps` file in the `/usr/lib/lp/postscript` directory to do the following:

- Internationalize 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, or
- Change or omit any of the text or labels

  The most-common substitution that sites choose to make is to specify that the sensitivity label prints instead of the information label at the top and bottom of banner pages. See:

  - Step 1 on page 448

For how to do any other customizations or internationalization, see the comments in the `tsol_separator.ps` file. Code Example 14–1 shows the comments and modifiable fields in the `tsol_separator` file, with the rest of the programming code removed.

CODE EXAMPLE 14–1 `tsol_separator.ps` Comments and Configurable Values *(of)*

```
%!
%ident ‘‘@(#)tsol_separator.ps 5.2 97/05/28 SMI; TSOL 2.x’’

%% Copyright (c) 1997 by Sun Microsystems, Inc.
%% All rights reserved.

%% This PostScript file is normally used as input to the lp.tsol_separator
%% program, which will prepend code to set the values of a number of
%% variables.  lp.tsol_separator is called by the printer interface script.

%% This PostScript file may be modified for local customizations or
%% internationalization.  Comments marked ‘‘INTERNATIONALIZE:’’ show
%% places where changes may be made for internationalization.  Comments
%% marked ‘‘CUSTOMIZE:’’ show places where some typical customization
%% changes may be made.
```

```
%% The following comments describe variables set by lp.tsol_separator
%% These variables are from the print job information that can be
%% displayed with lpstat or lpq.
%%
%%   /Job_HeadLabel   The classification (from the sensitivity label) to
%%                    be displayed at the top and bottom of the banner
%%   /Job_Printer     Printer Name
%%   /Job_Host        Host job was submitted from
%%   /Job_User        User who submitted the job
%%   /Job_JobID       Job number
%%   /Job_Title       Job title
%%
%% This variable is NO if an authorized user used the lp -o nobanner option
%% and the printer was set up to allow bannerless jobs.  Otherwise it is YES.
%%
%%   /Job_DoPageLabels Print page labels YES/NO.
%%
%% These variables are generated from the system clock value.
%%
%%   /Job_Date        Date and time the job is being printed, in the
%%                    locale's default format
%%   /Job_Hash        A randomly generated identifying number for
%%                    matching up the banner and trailer pages of the job
%%
%% The following variables are the job's Sensitivity Label and
%% Information Label as interpreted by the bcltobanner(3TSOL) library
%% routine.
%%
%%   /Job_HeadLabel   The classification (from the sensitivity label) to be
%%                    displayed at the top and bottom of the banner page.
%%   /Job_Protect     The sensitivity label to be displayed in the protect-as
%%                    field.
%%   /Job_Information The information label to be displayed in the ''the system
%%                    has labeled this data as'' field.  If ILs are turned off
%%                    in secconf, this variable will be an empty string.
%%   /Job_PageLabel   The information label to print at the top and bottom of
%%                    each page.  If ILs are turned off in secconf, this
%%                    variable will contain the sensitivity label instead.
%%   /Job_Caveats     The caveats from the information label.
%%   /Job_Channels    The channels from the information label.
%%
%% The following variables are the job's Sensitivity Label and
%% Information Label as interpreted by the bsltos and biltos library
%% routines.
%%
%%   /Job_SL_Internal The sensitivity label in internal view format.
%%   /Job_SL_External The sensitively label in external view format.
%%   /Job_IL_Internal The information label in internal view format.
%%   /Job_IL_External The information label in external view format.
  . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
%% CUSTOMIZE: The text between parentheses may be changed
%% to use different wording, or changed to empty
%% parentheses to eliminate the text.
/ILabelText (The system has labeled this data:) def
%% CUSTOMIZE: To not display the information label, change
%% this line to: /ILabel () def
/ILabel Job_Information def
  . . .
```

```
%% CUSTOMIZE: To use a different string at the top and
%% bottom of each page, change the following line.  For
%% instance, to use the sensitivity label in external view
%% format, change the line to: /PageLabel Job_SL_External def
%% To eliminate page labels complete, change this line to
%% set the page label to an empty string: /PageLabel () def
/PageLabel Job_PageLabel def
  . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
/Protect Job_Protect def
/Protect_Text1 (This output must be protected as:) def
/Protect_Text2
 (unless manually reviewed and downgraded.) def
  . . .
%% CUSTOMIZE:  To not print the caveats, change
%% this line to /Caveats () def
/Caveats Job_Caveats def
%% CUSTOMIZE:  To not print the channels, change
%% this line to /Channels () def
/Channels Job_Channels def
%% CUSTOMIZE:  To not print the hash number, change
%% this line to /Hash () def
/Hash Job_Hash def
%% CUSTOMIZE:  To not print the head label, change
%% this line to /HeadLabel () def
%% You may also substitute another string.  For example, to use
%% the SL in internal view format: /HeadLabel Job_SL_Internal def
/HeadLabel Job_HeadLabel def
  . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
(User:  ) User (@) Host append
  . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
(Job:  ) JobID append
  . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
(Printed at:  ) Date append
  . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
(Printer queue:  ) Printer append
  . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
{ TSOLJobInfo (JOB START) JobHashInfo}
 . . .
%% INTERNATIONALIZE: Replace the text between
%% parentheses with the appropriate text.
{ TSOLJobInfo (JOB END) JobHashInfo}
  . . .
%% End of tsol_separator.ps
```

Supported Printers

PostScript printers are the only types of printers supported. Non-PostScript printers function correctly but they do not have support for page labels or banner and trailer pages, so they do not meet Trusted Solaris printing requirements.

Issues About the Printing of PostScript Files

By default, users may not print PostScript files. This restriction exists because a knowledgable PostScript programmer could create a PostScript file that modifies the labels on the printer output.

To bypass this restriction, the security administrator role can assign the authorization called “print a PostScript file” to trusted users and role accounts. The security administrator should do so only if the account doing the printing can be trusted not to spoof the labels on printer output and if allowing anyone to print PostScript files is within the site’s security policy. See “Authorizations to Bypass Printing Defaults” on page 437.

Supported and Unsupported File Contents

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 administrator can install additional filters, which then would be trusted to have authentic labels and banner and trailer pages. See the Solaris 2.5.1 *System Administration Guide* for how filters are added.

Printers Connected to Non-trusted Print Servers

A printer connected to a host (print server) that is not running Trusted Solaris can print jobs sent from a Trusted Solaris host. However, the jobs print without labels and without trailer pages and without security information on any banner pages. This type of printing does not fully meet Trusted Solaris requirements.

If consistent with a site's security policy, the security administrator can set up printing to a printer connected to a host (print server) that is not running Trusted Solaris. Both of the following must be done:

- The security administrator configures the print server host with a specific sensitivity label, and
- Users send print jobs to the single-label printer at the same sensitivity label assigned to the print server.

  See Chapter 10," for how the security administrator assigns a single sensitivity label to an unlabeled host.

Permitting Publicly-readable Jobs to Be Printed by Default Without Labeled Pages

Certain users or groups of users, such as technical writers, need to produce publicly readable documents without labels printed on the top and bottom of the pages. If there is a printer connected to a Solaris print server available, the security administrator can set up the users' environments so that the publicly-readable jobs go to the printer connected to the Solaris host while jobs at all other labels go to Trusted Solaris hosts. The procedure requires understanding of how to set up user accounts as described in Chapter 3," and host network entries as described in Chapter 10." See:

- "To Set Up Publicly-Readable Print Jobs from an Unlabeled Print Server" on page 452"To Set Up Publicly-Readable Print Jobs from an Unlabeled Print Server" on page 452

Configuring Printers

The system administrator configures printers using these administrative applications:

- The modified Solstice Printer Manager—to set up local and remote printers
- The Device Allocation Manager: Administration dialog box—to specify a restricted label range for any printer

  Access the Solstice Printer Manager through the Application Manager Folder from the Solstice_Apps folder. See “To Access the Printer Manager” on page 438.

  Access the Device Allocation Manager through the Trusted Desktop subpanel in the Front Panel or through the Allocate Device option on the Trusted Path menu. “To Configure a Restricted Label Range for a Printer” on page 444

  See “To Install a Printer on a Print Server” on page 440 and “To Add Access to a Remote Printer” on page 445.

Modified Utilities and Man Pages

The commands in the following list are modified to work within Trusted Solaris security policy. To use the administrative commands identified by the 1MTSOL man page suffix, the account must have the *administer printing* authorization. For details on these commands and the Trusted Solaris differences, see the man(1TSOL) pages.

TABLE 14–1 Printing-related Commands

Command	Description
accept / reject(1MTSOL)	allow or prevent the queueing of print requests
enable / disable(1TSOL)	enable, disable LP printers
lp / cancel(1TSOL)	send / cancel requests to an LP print service
`lpadmin(1MTSOL)`	configure the LP print service
lpc(1BTSOL)	(BSD) line printer control commands
lpq(1BTSOL)	(BSD) display the queue of printer jobs

TABLE 14–1 Printing-related Commands *(continued)*

Command	Description
lpr(1BTSOL)	(BSD) send a job to the printer
lprm(1BTSOL)	(BSD) remove jobs from the printer queue
lpsched / lpshut / lpmove(1MTSOL)	start / stop the LP print service
lpstat(1TSOL)	print information about the status of the LP print services
lpsystem(1MTSOL)	register remote systems with the print service
lptest(1BTSOL)	(BSD) generate line printer ripple pattern

Authorizations to Bypass Printing Defaults

Table 14–2 defines the authorizations that are related to printing. In order for a user or role to be able to do the action described under the Purpose heading, the authorization must be in one of the profiles assigned to the user or role. See “To Assign Printing-related Authorization(s) to an Account” on page 449.

TABLE 14–2 Authorizations Related to Printing

Name	Purpose	Default Profile	Default Role Assigned Profile
administer printing	Permits the user or role to administer printing using administrative utilities to start and stop printing daemons, list and cancel other users' print jobs, and so forth.	Printer Security All Authorizations	secadmin N/A
print without banners	Permits the user or role to submit a print request using the **lp -o nobanner** option, to suppress the printing of banner and trailer pages. NOTE: For this to option to work, the Always Print Banners check box on the Printer Manager entry for the printer *must no*t be checked.	Printer Security	secadmin
print a PostScript file	Allows a user to print a PostScript file.	Printer Security Convenient Authorizations	secadmin N/A
print without labels	Allows a user to submit a print request that specifies (by means of the **lp** command with the **-o nolabels** option) that labels will not be printed on the top and bottom of the print job's body pages.	Printer Security Convenient Authorizations	secadmin N/A

Printing-related Procedures

▼ To Access the Printer Manager

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

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **Click the Application Manager icon in the Front Panel to open it.**

3. Double-click the Solstice_Apps icon.

4. Double-click the icon for the Printer Manager.

The Printer Manager: Load dialog box displays with None as the only Naming Service option, as shown in the following figure.

Figure 14–7 Printer Manager: Load Dialog Box with None as the Only Naming Service Option

5. Choose None for the naming service.

The Printer Manager displays, as shown in Figure 14–8.

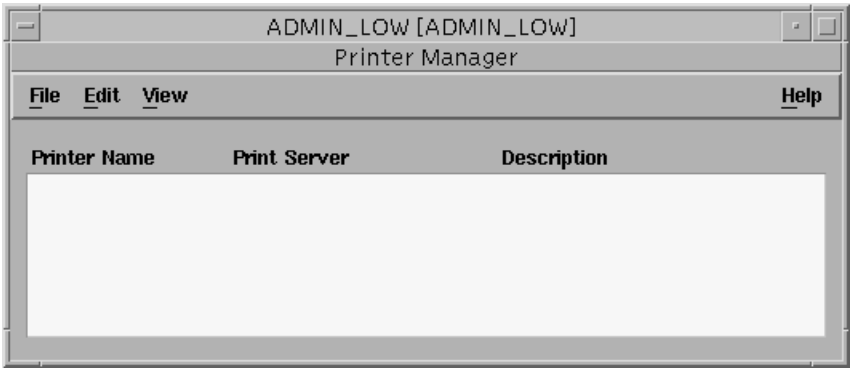


Figure 14–8 Printer Manager

Go to “To Install a Printer on a Print Server” on page 440 or “To Add Access to a Remote Printer” on page 445.

▼ To Install a Printer on a Print Server

1. **Connect the printer to a serial or parallel port on a print serve using the appropriate cable, as described in the printer’s installation manual.**

   The printer in the screen examples is connected to the serial port `/dev/term/b`.

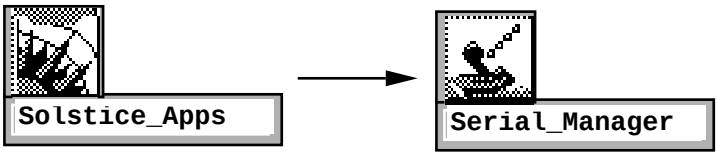
2. **Assume the security administrator role on the print server, and go to an ADMIN_LOW workspace.**

   See “To Login and Assume an Administrative Role” on page 15, if needed.

3. **If the printer is connected to a serial port, make sure the correct baud rate is set, using the Serial Manager from the Solstice_Apps folder.**

   a. **Double-click the icon for the Serial Manager in the Solstice_Apps folder.**

      See “To Launch Solstice Administration Tools” on page 27, if needed.



   b. **On the Serial Port Manager dialog box, double-click the entry for the port to which the printer is connected.**

The Serial Port Manager: Modify dialog box displays. See Figure 14–9.

c. On the Serial Port Manager: Modify dialog box, verify that the baud rate specified for the port is set correctly.

See the printer documentation for the correct baud rate.

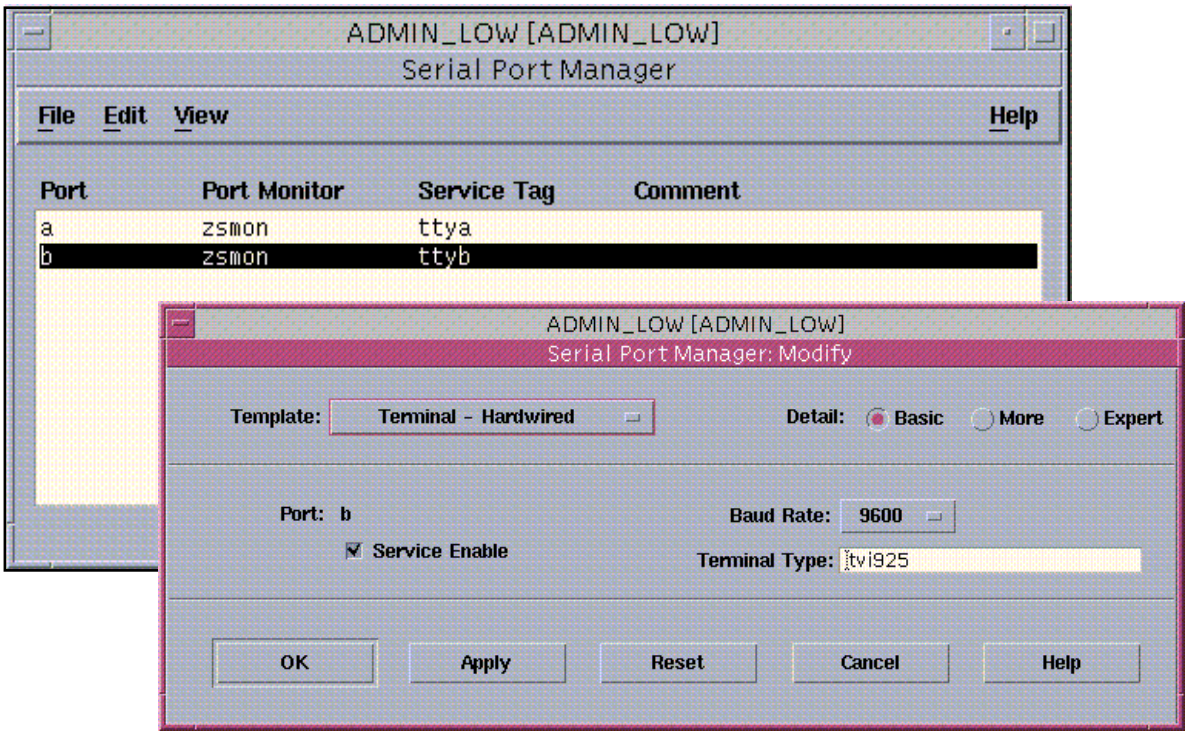


Figure 14–9 Serial Port Manager and Serial Port Manager: Modify Dialog Boxes

4. Access the Printer Manager from the Solstice_Apps folder.

See “To Access the Printer Manager” on page 438, if needed.

5. Choose Install Printer from the Edit menu, as shown in Figure 14–10

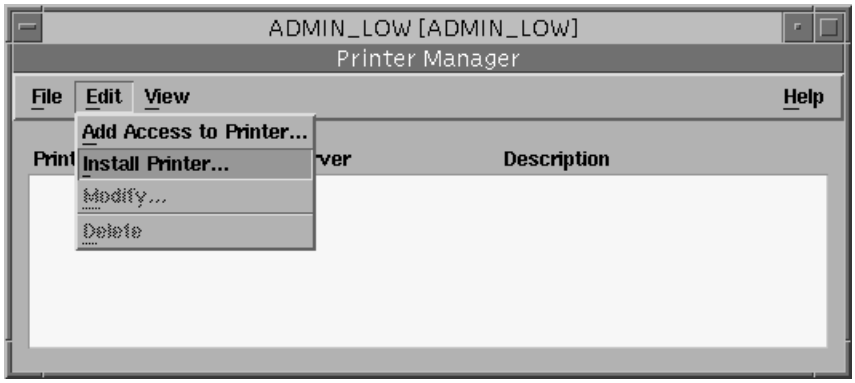


Figure 14–10 Printer Manager: Selecting Install Printer from the Edit Menu

The Printer Manager: Install Printer dialog box displays as shown in Figure 14–11.

6. **Configure the printer.**
   a. **Supply a name for the printer.**
   b. **Supply a description if desired.**
   c. **From the Printer Port menu, select the name of the port where the printer is connected, or select Other and enter the name of an alternate port, if the printer is connected to a non-standard port.**
      ttya and ttyb are for serial cabling; bpp0 is for parallel cabling.
   d. **Leave Printer Type and File Contents set to PostScript.**

Warning - If you change the Printer Type and File Contents settings from the default value of PostScript, printing does not work.

   e. **Check the box next to Default Printer or not, as desired.**
   f. **If your site does not give to any users the authorization to print without banner and trailer pages, check the box next to Always Print Banners.**

Note - If your site gives any users the authorization to print without banner and trailer pages, do not check the box next to Always Print Banners.

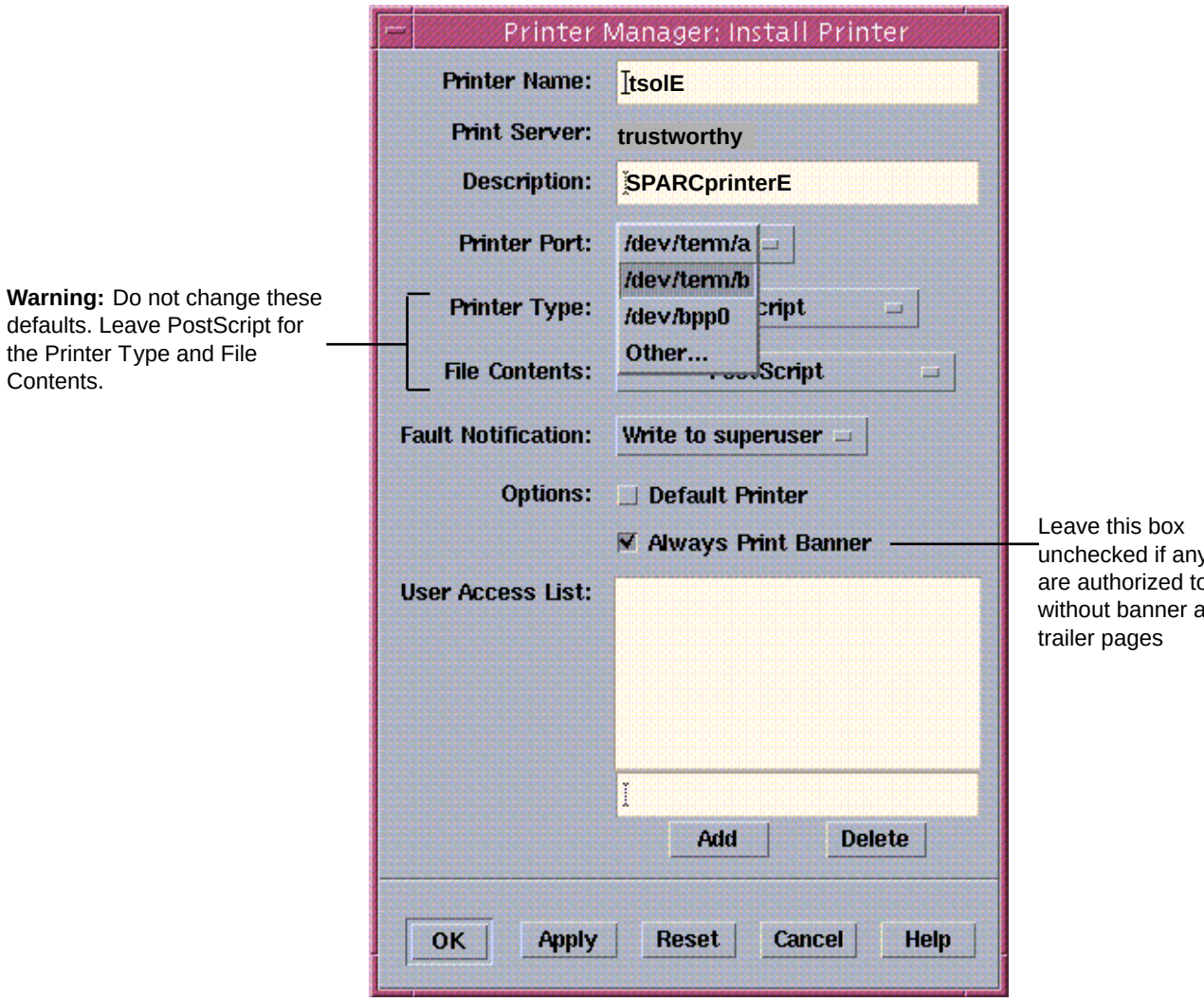


Figure 14–11 Printer Manager: Install Printer Dialog Box

g. Add users to the User Access List or not, as desired.
Adding the username of any one user excludes all others.

h. Click OK to save the changes and close the dialog box.

7. **To change the label range of the printer from ADMIN_LOW to ADMIN_HIGH, if desired, follow the steps under "To Configure a Restricted Label Range for a Printer" on page 444.**

▼ To Configure a Restricted Label Range for a Printer

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

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **Install the printer using the Printer Manager.**

   See "To Install a Printer on a Print Server" on page 440.

3. **To change the label range from the default range of ADMIN_LOW to ADMIN_HIGH, use the Device Allocation Manager.**

   a. **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 new printer in the Devices list.**

   d. **Click the Configure button to display the Device Allocation: Configuration dialog box.**

      See Figure 14–12.

   e. **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 sensitivity label.**

   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.**

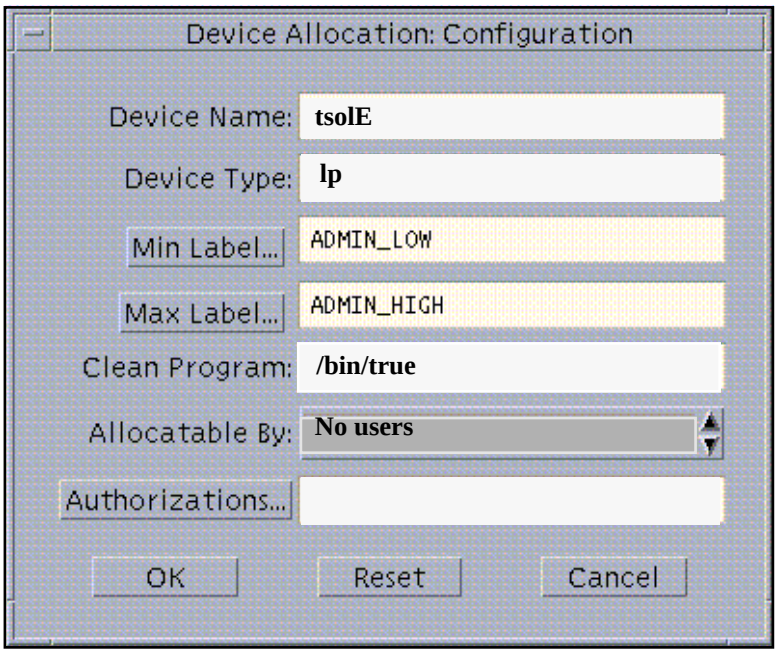


Figure 14–12 Device Allocation: Configuration Dialog Box

▼ To Add Access to a Remote Printer

1. **Assume the security administrator role on the local host. where printing to a remote printer is being set up.**
   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Access the Printer Manager.**
   See “To Access the Printer Manager” on page 438, if needed.

3. **Choose Add Access to Printer from the Edit menu, as shown in Figure 14–13.**

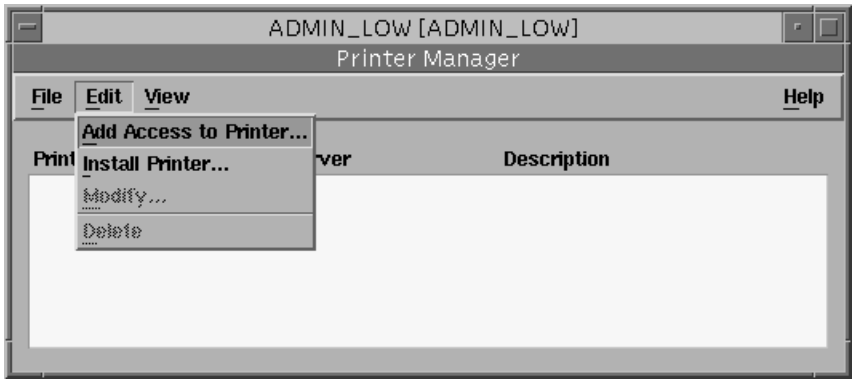


Figure 14–13 Printer Manager: Selecting Add Access to Printer from the Edit Menu

The Printer Manager: Add Access to Printer dialog box displays, as shown in Figure 14–14.

4. Specify access to a remote printer from the local host.

a. Type in the name of the remote printer in the Printer Name field.

b. Type in the print server's name in the Print Server field.

c. OPTIONAL. Type a description in the Description field.

d. OPTIONAL. Check the box next to Default Printer.

e. Click OK to save the changes and close the dialog box.

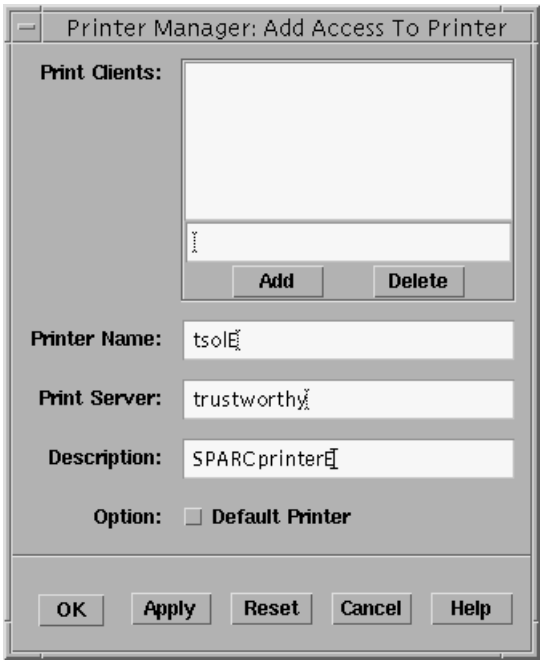


Figure 14–14 Printer Manager: Add Access to Printer Dialog Box

5. Specify access to a printer for a remote print client, if desired.

a. In the field above the Add and Delete buttons, type in the name of a remote print client's name.

b. Type in the name of the printer in the Printer Name field.

c. Type in the print server's name in the Print Server field.

d. OPTIONAL. Type a description in the Description field.

e. OPTIONAL. Check the box next to Default Printer.

f. Click the Add button to add the name of the print client to the list.
If you are done, go to Step 5 on page 447. To add more clients go back to Step 5 on page 447.

g. Click OK to save the changes and close the dialog box.

▼ To Specify SLs to Print Instead of ILs on Body Pages

1. **Assume the security administrator role on the print server.**
   See "To Login and Assume an Administrative Role" on page 15, 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 29, if needed.

3. **Find the following line:**

```
/PageLabel Job_PageLabel def
```

4. **Change the value of** `/PageLabel` **with whichever of the following settings you wish to use:**

Setting	What Prints
`Job_SL_Internal`	`sensitivity label in internal view format`
`Job_SL_External`	`sensitivity label in external view format`
`Job_IL_Internal`	`information label in internal view format`
`Job_IL_External`	`information label in external view format`

   The following line causes the external label view of the sensitivity label to be printed on body pages.

```
/PageLabel Job_SL_External def
```

5. **Save and close the file.**

```
:wq
```

▼ To Allow Some Users to Print Jobs Without Banners and Trailers

Warning - If the Always Print Banner check box on the Printer Manager is checked, banner and trailer pages always print, even if the user has the print without banners authorization and uses the `-o nobanner` option to `lp`.

1. **Assume the security administrator role on the print server.**
   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **Bring up the Printer Manager.**
   See "To Access the Printer Manager" on page 438, if needed.

3. **Make sure that the Always Print Banner check box on the Printer Manager is *not* checked.**

4. **Exit the Printer Manager.**

5. **Make sure that the *print without banners* 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 449, if needed.

6. **Make sure that the user or role submits jobs using `lp` 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 Login and Assume an Administrative Role" on page 15, if needed.

2. **Bring up the User Manager.**

See “To Launch Solstice Administration Tools” on page 27. See also Chapter 5,” in this manual if needed.

3. **Click the Profiles button to open the profiles dialog box.**

   a. **Make sure that the desired print-related authorization is contained in one of the user’s execution profiles.**

   b. **Move a profile containing the print-related authorization to the Available list, if desired.**

      The following table shows the printing-related authorizations and the default profiles that contain them.

TABLE 14–3 Authorizations Related to Printing

Name	Default Profile	Default Role Assigned Profile
administer printing	Printer Security	secadmin
	All Authorizations	N/A
print without banners	Printer Security	secadmin
print a PostScript file	Printer Security	secadmin
	Convenient Authorizations	N/A
print without labels	Printer Security	secadmin
	Convenient Authorizations	N/A

If the defaults have not been modified, including one of the profiles shown above in the user’s list of profiles gives that user the listed authorization.

Note - If none of the default execution profiles are appropriate, the security administrator can create a new profile that includes the desired printing-related authorization(s), either by themselves, or along with any other commands needed by the profile’s users to perform the desired work (such as, for example, `lp` or `lpadmin`). How to create a new profile is described in Chapter 8,” in this manual.

c. **Store your changes by clicking Apply or OK at the bottom of the Profiles dialog box.**

   Clicking Apply saves the changes and leaves the dialog box displayed. Clicking OK saves the changes and closes the dialog box.

d. **Click Done or Save on the User Manager Navigator.**

4. **Choose the Exit option from the User Manager File menu to exit from all User Manager windows.**

▼ To Suppress the Printing of Page Labels on All Print Jobs

1. **Assume the security administrator role.**

   See "To Login and Assume an Administrative Role" on page 15, 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 29, if needed.

3. **Find the following lines:**

```
%% To eliminate page labels complete, change this line to
    %% set the page label to an empty string: /PageLabel () def
            /PageLabel Job_PageLabel def
```

Note - The value Job_PageLabel may have been changed at your site.

---

4. **Replace the value of** `/PageLabel` **with an empty parentheses.**

```
%% To eliminate page labels complete, change this line to
    %% set the page label to an empty string: /PageLabel () def
            /PageLabel () def
```

▼ To Allow Some Users to Print Jobs Without Page Labels

1. **Make sure that the *print without labels* 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 449, 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 allows an authorized user or role to print jobs without labels when working at any sensitivity label.

▼ To Set Up Publicly-Readable Print Jobs from an Unlabeled Print Server

1. **In the `tnrhdb/tnrhtp` entries that define a Solaris print server, assign to the print server the sensitivity label that identifies 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 sensitivity label range.**

   b. **Instruct each user how to define the PRINTER variable in the user's publicly-labeled home directory SLD.**

      i. **Go to the publicly-labeled home directory SLD.**

      ii. **Open the `.login` file for editing.**

      iii. **Define the PRINTER variable to be the name of the printer connected to a Solaris print server.**

      When a printer named nolabels is connected to a single-label print server whose label is PUBLIC, the `.login` file in the PUBLIC SLD directory would have the following environment variable defined.

```
setenv PRINTER nolabels
```

iv. Write and quit the file.

a. Instruct each user how to define the PRINTER variable in all other SLDs. Tell them to do the following for each remaining SLD at which they work:

i. Go to the home directory SLD.

ii. Open the `.login` file for editing.

iii. Define the PRINTER variable to be the name of a printer connected to a Trusted Solaris print server.

iv. Save 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 the publicly-labeled SLD.

CHAPTER 15

Managing Devices

This chapter describes how to meet an organization's goals for protection of information on devices. This chapter includes the following topics.

This chapter provides the following device-related procedures:

Device Access Policy

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 and the default policy for floating of information labels on devices. These policies automatically apply to any new devices added to the system.

TABLE 15–1 Default Device Access Policy

Policy Type Names	Description	Default Policy
`il_float_policy`	Whether to float the device's IL	For reads and writes, float the device's IL to the reading process' IL.
`data_mac_policy`	SL required to access the device	For reads and writes, the process' SL must equal the device's SL.
`attr_mac_policy`	SL required to access the device's attributes [by acl(2), chmod(2TSOL), chown(2TSOL), and stat(2TSOL)]	For read access to the device's attributes, the process' SL must dominate the device's SL. For write access to the device's attributes, the process' SL must equal the device's SL.

TABLE 15–1 Default Device Access Policy *(continued)*

Policy Type Names	Description	Default Policy
`open_priv`	Privilege required to open the device	No privileges are required.
`str_type type`	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 can change default policies and define new policies on each host by editing the device_policy(4TSOL) file. To only way to put changes to the `/etc/security/tsol/device_policy` file into effect is to reboot. See the device_policy(4TSOL)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 475.

Security Issues Addressed by Device Allocation

The device-allocation mechanism controls security risks associated with the use of certain input/output devices. For example, an unscrupulous user can covertly read information from another user’s tape in a commonly-accessible tape device.

The device-allocation mechanism allows the security administrator to control whether individual users are allowed to access certain devices by granting or withholding the needed authorization. The unauthorized user in the default Trusted Solaris distributed system cannot allocate devices such as tape drives, CD-ROM drives, or floppy disk drives.

Only one authorized user at a time can access a device. Before the device can be deallocated and made available to another user, the user is prompted to clear the device of information. Any removable media (such as a tape or floppy disk) must be ejected before deallocation completes.

Other devices (such as the frame buffer, computer memory, and disks) are automatically allocated and deallocated on behalf of all users, and any information contained on such devices is automatically cleared between allocations.

MAC Issues Associated with Device Label Ranges

A normal user with the device allocation authorization can import or export information only at a single sensitivity label, which is the sensitivity label at which the user allocates the device. (See “Device-related Authorizations” on page 461 for a description of the device allocation authorization.

Each allocatable device has a default sensitivity label range of ADMIN_LOW to ADMIN_HIGH that can be restricted by the security administrator using the Device Manager. Label ranges may also be set for two types of nonallocatable devices: framebuffers (to restrict console logins) and printers. Normal users are restricted to accessing devices whose label range includes the labels at which they allowed to work.

A restricted label range can be used to control access to a device that is not physically protected by limiting access to only those users with certain clearances. See “Min Label... and Max Label...” on page 464 for how the label range is set.

Label Range on a Host

To restrict direct login access through the console, the security administrator sets a restricted label range on the framebuffer.

Label Range on a Local Printer

When a host has a local printer, the security administrator can set a restricted label range for that printer to limit the label range of jobs that it will print.

Managing Device Allocation 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, if a new device is added, the security administrator must decide whether to make the new device allocatable.

By using the Device Maintenance dialog box accessed from the Device Allocation tool, the security administrator role can make the following changes to the defaults.

- Make a new device or existing device either:
  - Allocatable with one or more authorizations
  - Allocatable by anyone without an authorization
  - Not allocatable
- Restrict the label range on a device

Changes can be made at any time. See "To Access the Device Allocation Administration Dialog Box" on page 477 and the following procedures.

Decision to Make Before Configuring the System

1. **Find out which devices are listed in the default** `device_allocate(4TSOL)` **file and what their default definitions are.**
2. **Decide whether the defaults are consistent with the site's security policy, and if not, which defaults to change.**
3. **Decide whether to make additional devices allocatable.**
4. **Decide which normal users, if any, should be allowed to allocate devices.**
5. **Decide whether to accept or change the default label range settings on the listed non-allocatable devices.**

Understanding the Device Allocation Manager

User-allocatable devices and some non-allocatable devices are allocated, configured, and managed through the Device Allocation Manager. Figure 15–1 shows the Device Allocation Manager with the default list of devices that are available to the security administrator role for allocation.

The Device Allocation Manager can be used only by accounts that have the device allocation authorization, *allocate device*. Unauthorized users see an empty list under Available Devices. When an allocatable device is currently allocated by another user

or is otherwise not available, the authorized user does not see that device listed under Available Devices.

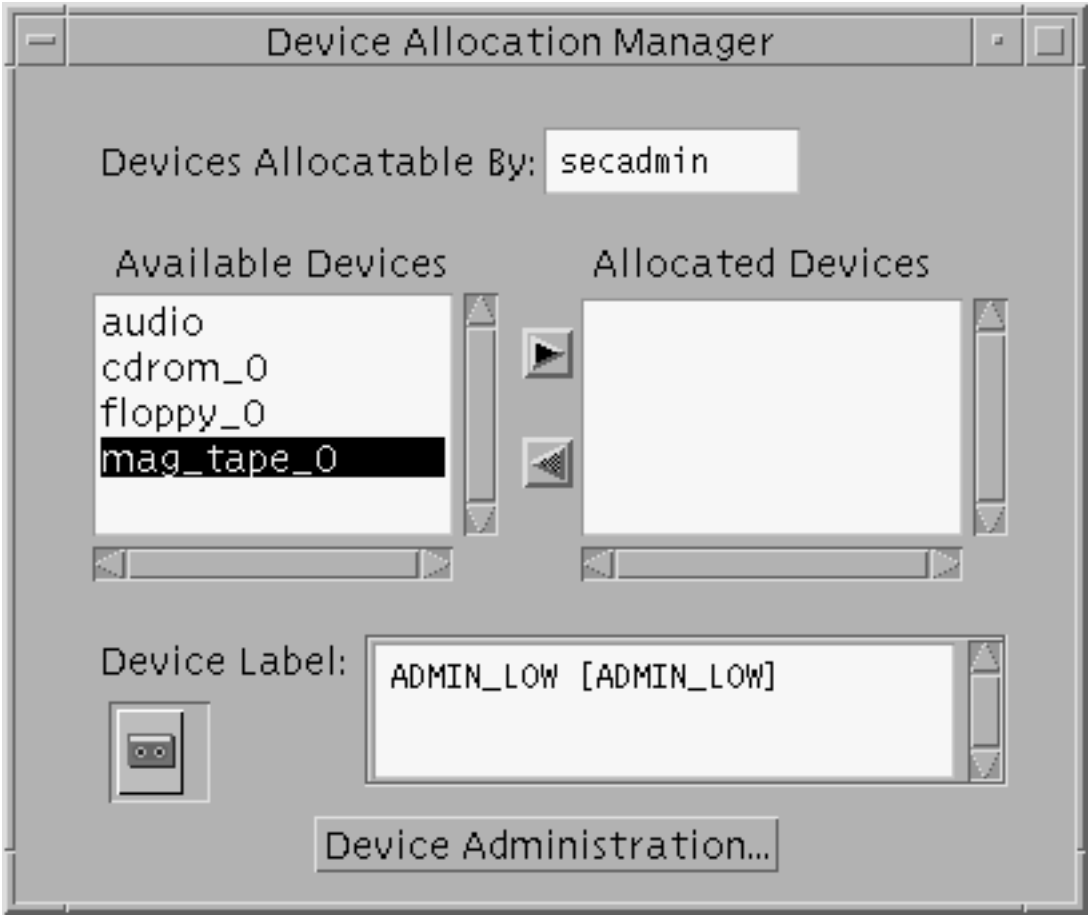


Figure 15–1 Device Allocation Manager

When a Device is Not Available

If a user who is attempting to allocate a device 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 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, both the security administrator and system administrator roles have the authorization to force deallocation of a device or to reclaim it from the error state.

Training Authorized Users, Defining, and Enforcing Security Procedures

By default, the security administrator decides who has the authorization to allocate devices. The 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 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 execution profiles for an account, and when the execution profile has been assigned to the account by the security administrator role using the User Manager. For more information, see Part 1 of this manual, . See also “To Assign Device-related Authorization(s) to an Account” on page 484.

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 44 and do the procedure under “To Add An Authorization ” on page 47 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. See “Authorizations...” on page 465.

The following table shows the device-related authorizations.

TABLE 15–2 Device Allocation, Configuration, and Management Authorizations

Authorization Name	Description
allocate device	Allows a user to allocate a device and to specify the CMW label to associate with information imported from it, or exported to it.
configure device attributes	Allows an administrator to configure a device. Device configuration includes setting the device name, type, label range, allocatable status, and allocation authorization list.
revoke or reclaim device	Allows an administrator to deallocate a currently allocated device or reset an allocate error state to make a device allocatable again.

Understanding the Device Allocation Manager: Administration Dialog

The Device Administration button shown in the following figure displays on the bottom of the Device Allocation Manager only for an administrative role account that has either one or both of the authorizations needed to administer devices, *configure device attributes* or *revoke or reclaim device*. (See Table 15–2 for the purposes of these authorizations.)

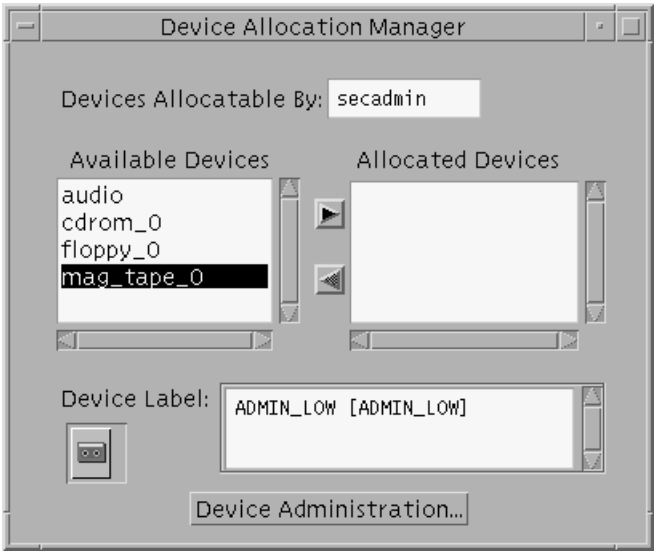


Figure 15–2 Device Allocation Manager

As shown in Figure 15–3, clicking the Device Administration button launches the Device Allocation: Administration dialog box.

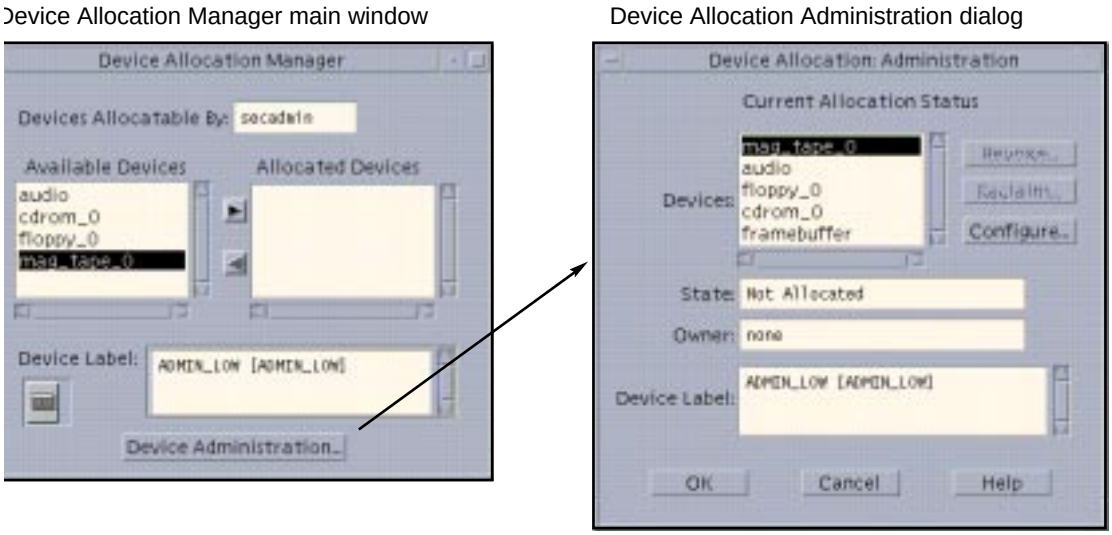


Figure 15–3 Device Allocation and Administration Dialogs

The dialog box displays the State, Owner, and the CMW Label (Information Label[Sensitivity Label]) of the highlighted device.

Revoke

If the Revoke button is active, the State: field. for the highlighted device displays: Allocated. If the account has the *reclaim or revoke 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: Allocate Error State. If the account has the *reclaim or revoke device* authorization, clicking the *Reclaim* button releases a selected device from the allocate error state and changes the state to Not Allocated.

Configure

If the account has the *configure device attributes* authorization, clicking the *Configure* button brings up the *Device Allocation: Configuration* dialog box. See the following figure.

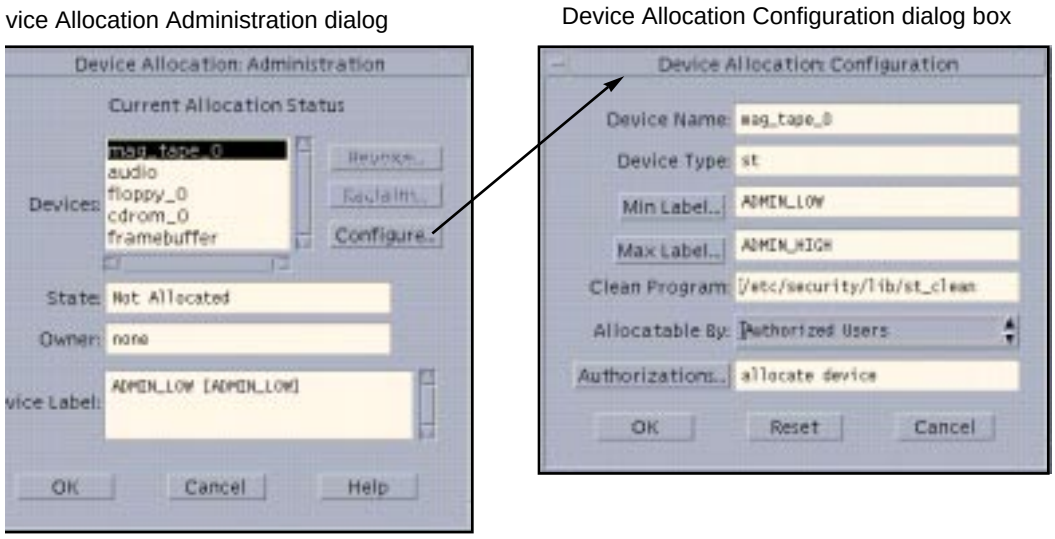


Figure 15–4 Device Allocation Administration and Configuration Dialog Boxes

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 Figure 15–4.

Device Name and Device Type

The Device Name and Device Type display for the device selected in the Administration dialog. These fields cannot be edited.

Min Label... and Max Label...

Clicking the Min Label... and Max Label... buttons brings up a label builder that the security administrator can use to specify either a minimum or maximum label. 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 Allocation and Setting Device Label Ranges” on page 458 for more about setting a device’s label range. This field is valid for allocatable and non-allocatable devices.

Clean Program

The Clean Program field allows the security administrator to enter the path of a device_clean(1MTSOL) 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 468.

Allocatable By

For allocatable devices, the Allocatable By scrolling list offers three choices:

- Authorized users
- All users
- No users

  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 *allocate device*. The security administrator can click the Authorizations button to change to another authorization, or to specify multiple authorizations. The following figure shows the Authorizations dialog box.

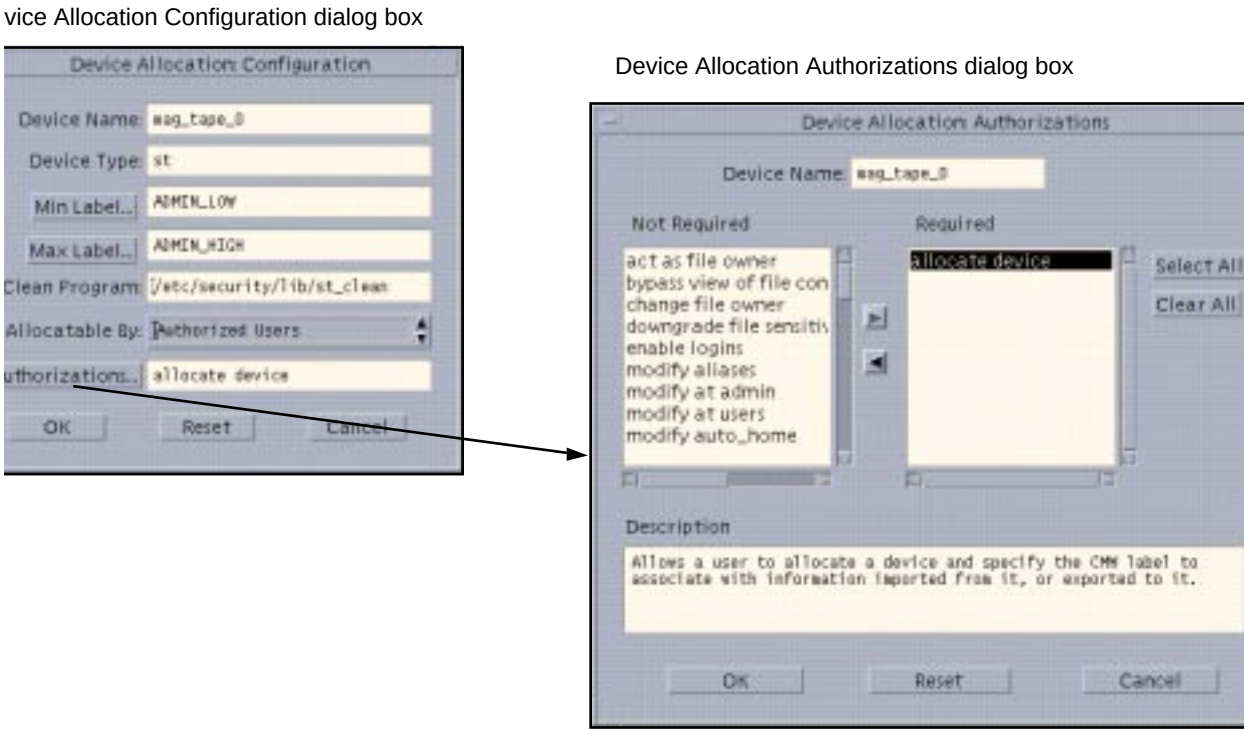


Figure 15–5 Clicking the *Authorizations* Button Displays the *Device Allocation: Authorizations* Box

Remote Device Management

The add_allocatable(1MTSOL), remove_allocatable(1MTSOL) commands, the Add Allocatable Device action, and the Device Allocation Manager makes changes to local versions of the device_allocate(4TSOL) and device_maps(4TSOL) files on the host on which they are run. To push the same changes onto multiple hosts across the network, the security administrator can remotely distribute changed versions of the device_allocate and device_maps files using rdist(1). However, if adding a new device, the security administrator still needs to log in remotely to each host and manually create the required ancillary file. Alternately, the security administrator can do a CDE remote login to each host where the change must be madïÿ|$d run the Device Allocation Manager.

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 must also have a certain set of DAC permissions, owner, and group in order for the device to be allocatable. See the following table for the DAC permissions, owner, group, and sensitivity label of the ancillary file when it is:

- Allocatable
- Allocated
- In the allocate error state

TABLE 15–3 Required Ancillary File Characteristics for Allocatable and Allocated Devices

	DAC permissions (mode)	**Owner**	**Group**	**SL**
Allocatable	0000	bin	bin	ADMIN_LOW
Allocated	0600	*user*	*user's group*	*user's process's SL*
Error State	0100	bin	bin	ADMIN_HIGH

Allocate Error State

As shown in Table 15–3, 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 sensitivity label of ADMIN_HIGH. One way that a device can be put into an allocate error state is by the device`_clean`(1MTSOL) 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 Reclaim button on the Device Allocation: Maintenance dialog box can be used by both of the default administrative roles to reclaim devices from the error state. See also "Reclaim" on page 463.

Device-Clean Scripts

A device-clean script is run any time a device is allocated or deallocated. Deallocation is usually done by the user who allocates the device. If necessary, the Revoke button on the Device Allocation: Maintenance dialog box can be used by either of the default administrative roles to forcibly deallocate a device. See “Revoke” on page 463 for more about forced deallocation.

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 470.

Device-Clean Script for Tape Devices

Table 15–4 shows the three supported tape devices. `st_clean` is the name of the *device-clean* script used for all.

TABLE 15–4 Supported Tape Devices

Tape Device Type
SCSI 1/4 inch tape
Xylogics 472 1/2 inch tape
Archive 1/4 inch tape

The `st_clean` script uses the `rewoffl` option to mt(1) to do the device cleanup. 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 CMW 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 the security administrator 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, if media is found in a device, the media 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 CMW label. If the eject(1)command fails, the device is placed 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 can prevent the automatic display of the File Manager by following the procedure in Step 1 on page 487. 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 can the following action in `rmmount.conf` to automatically bring up `workman`.

```
action cdrom action_workman.so /pathname/to/workman
```

For example, the following brings up `workman` when the program is at ADMIN_LOW 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` or `newsecfs` 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(1) 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 CMW label is not displayed by the `audio_clean` script.

Writing New Device-Clean Scripts

Some devices that a site can make 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 `st_clean` script (`/etc/security/lib/st_clean`). See the device_clean(1MTSOL) man page.

- 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:

`st_clean -[I|F|S] -[A|D]` *`device_name cmw_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`.

device_name is a string with the name of the device

cmw_label is a hexadecimal representation of the CMW label

Handling of Allocated Devices at Boot

At boot time, allocated devices are re-allocated and remounted. Entering `boot` with the `-r` option forces deallocation of the devices.

Considerations When Importing and Exporting Information

To Export Information

A media device (for a tape or floppy) must be allocated at the sensitivity label of the information that is to be stored on the device.

To Import Information

When a device that contains a tape or floppy used for storage of information is allocated for reading, the user must allocate the tape drive at the sensitivity label shown on the physical label on the tape.

Using `tar` *with Device Allocation*

The only normal user command modified to store and retrieve Trusted Solaris security attributes is tar(1TSOL). The tar command has been extended to be able to create, update, list the table of contents of, and extract a tarfile that contains extended Trusted Solaris security attributes. Without privileges, the `tar` command works within the Trusted Solaris security policy. When invoked by an ordinary user without privileges, `tar` works at a single sensitivity label and can be used only to create a tarfile at the sensitivity label of the current workspace. `tar` does not work unless the user first allocates the media device. The following figure shows the error message displayed by `tar` when the device is not allocated first.

```
trusted4% tar -cvT *

tar: /dev/rmt/0: Permission denied
```

`tar` has new options used to store and extract Trusted Solaris security attributes.

The `tar` command traverses any MLD it encounters. SLDs dominated by the `tar` process's sensitivity label are walked. With the appropriate override privileges all SLDs can be walked.

Ancillary Files for `tar` *Files*

When files are archived with the Trusted Solaris extended security attributes, each file is preceeded by its own ancillary file that holds the extended security attributes. The ancillary file has the same filename as its corresponding archived file, suffixed by the string "(A)". The following figure shows two files being archived using `tar` with the `T` option to preserve security attributes and the `v` option for verbose output. In the figure, both files shown are preceded by their ancillary files. For example, `Dtapps.bw.xbm` file is preceeded by the `Dtapps.bw.xbm(A)` ancillary file.

```
a Dtapps.bw.xbm(A) 1 tape blocks

a Dtapps.bw.xbm 2 tape blocks

a FrontPanel.Workspace.admin.menu.rs(A) 1 tape blocks

a FrontPanel.Workspace.admin.menu.rs 593 tape blocks
```

New Trusted Solaris `tar` *Options*

The new `T` option is used to store and extract security attributes for each file. The new `d` option works with the `T` and `x` options to retrieve the security attributes from Trusted Solaris 1.x tarfiles.

`T`

When the `T` option is used with the `c`, `r`, or `u` options, `tar` stores security attributes for each file (including MLD and SLD information for directories). When the `T` option is used with the `x` option, `tar` extracts security attributes along with the files.

When `T` is used with the `t` option, the tarfile content is displayed with a line for each ancillary file and a line for each archived file.

When used with the x option to extract a tarfile, the tar program restores each archived file using the MLD and SLD information, and the extended security attributes.

`d`

The `d` option should be used when a tarfile is in Trusted Solaris 1.x format. The `d` option is only valid with the options `t`, `T`, and `x`. When `d` is used with `t` to display a tarfile's contents, the `tar` program processes the input tarfile according to the Trusted Solaris 1.x format. If `d` is used with `t` and `T`, then the contents of the Trusted Solaris 1.x tarfile are displayed with a line for each ancillary file and a line for each archived file.

When `d` is used with `x` to extract a tarfile, the `tar` program processes the input tarfile according to the Trusted Solaris 1.x format. If `d` is used with `x` and `T`, the appropriate MLD and SLD information and other extended security attributes that are valid on the Trusted Solaris 2.5/2.5.1 system, are applied when each archived file is restored.

When `d` is used with `c`, ACLs are created in the tarfile along with other information.

Note - Errors will occur when a tarfile with ACLs is extracted by previous versions of `tar`.

The procedure under “To Allocate a Tape Device and Use tar to Save Security Attributes on Exported Information” on page 474” uses `tar` with the `cvT` options to create a tarfile that saves extended security attributes. See also the tar(1TSOL) man page for more information.

Extracting Files Created on Trusted Solaris 1.x or 2.x Systems

When either Trusted Solaris 1.x or Trusted Solaris 2.5 tarfiles are being extracted, the `label_encodings`(4TSOL) file in effect when the tarfile was created must be compatible with the currently installed `label_encodings` file. When a Trusted Solaris 1.x tarfile is restored onto a Trusted Solaris 2.5 system, the label SYSTEM_HIGH is mapped to the label ADMIN_HIGH, and the label SYSTEM_LOW is mapped to label ADMIN_LOW. The privileges and file audit mask from Trusted Solaris 1.x are not used on the restored files because their formats are not compatible with Trusted Solaris 2.x equivalent security attributes.

Device-related Commands and Databases

See the man pages for the following commands and databases:

TABLE 15–5 Device-related Commands and Databases

Command or File Name	Description
allocate(1MTSOL)	Device allocation command line interface
add_allocatable(1MTSOL)	Add a device to device_allocate(4TSOL), device_maps(4TSOL), and create an ancillary file in `/etc/security/dev`
deallocate(1MTSOL)	Device deallocation command line interface
device_clean(1MTSOL)	Device cleaning programs
dminfo(1MTSOL)	Report on specified device’s entry in the device_maps file.

TABLE 15–5 Device-related Commands and Databases *(continued)*

Command or File Name	Description
list_devices(1MTSOL)	List devices specified in the `device_maps` file
remove_allocatable(1MTSOL)	Remove a device from `device_allocate`, `device_maps` and delete its ancillary file from `/etc/security/dev`
device_allocate(4TSOL)	Database for managing allocatable and some non-allocatable devices
device_maps(4TSOL)	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 CMW label (INFORMATION LABEL[SENSITIVITY 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 ADMIN_LOW[C}


   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 in the Device Deallocation for mag_tape0 window.

```
Please eject the tape in mag_tape_0
```

5. **Make sure to protect the exported information at the security level indicated in the CMW label 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 Login and Assume an Administrative Role" on page 15, if needed.

2. **Determine the *driver_name* and *minor_name* and the device special filenames 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/c0t6d0s

   2 /dev/dsk/c0t6d0s3 /dev/dsk/c0t6d0s4 /dev/dsk/c0t6d0s5
   /dev/dsk/c0t6d0s6 /dev/

   dsk/c0t6d0s7 /dev/rdsk/c0t6d0s0 /dev/rdsk/c0t6d0s1
   /dev/rdsk/c0t6d0s2 /dev/rdsk

   /c0t6d0s3 /dev/rdsk/c0t6d0s4 /dev/rdsk/c0t6d0s5
   /dev/rdsk/c0t6d0s6 /dev/rdsk/c0

   t6d0s7:\
```

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 29, if needed.

4. When the default policy for devices is not consistent with your site’s security policy, create a specific 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 Table 15–6. For how to specify alternate policy settings, see the device_policy(4TSOL) man page.

TABLE 15–6 Default Device Policy

Policy Types	Description	Default Policy
il_float_policy	Whether to float the device's IL	For writes, float the device's IL to the reading process' IL. For reads, float the process' IL.
data_mac_policy	What SL the process must have to access the device.	For reads and writes, the process' SL must equal the device's SL.
attr_mac_policy	How to handle access to the device's attributes [by acl(2), chmod(2TSOL), chown(2TSOL), and stat(2TSOL)]	For read access to the device's attributes, the process' SL must dominate the device's SL. For write access to the device's attributes, the process' SL must equal the device's SL.
open_priv	Privilege required to open the device	No privileges are required.
str_type type	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 Access the Device Allocation Administration Dialog Box

1. **Assume the security administrator role and go to an ADMIN_LOW workspace.**
   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **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. The Device Allocation Manager displays, as shown in the following figure.**

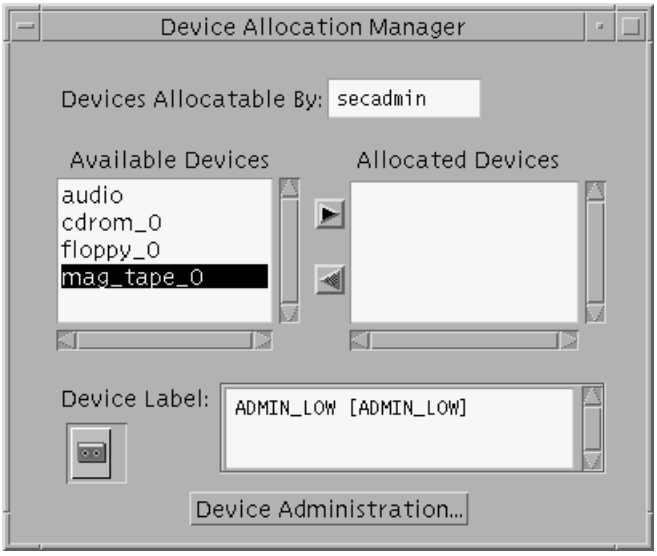


Figure 15–6 Device Allocation Manager

3. Click the Device Administration button to bring up the Device Allocation Administration dialog box shown in Figure 15–7.

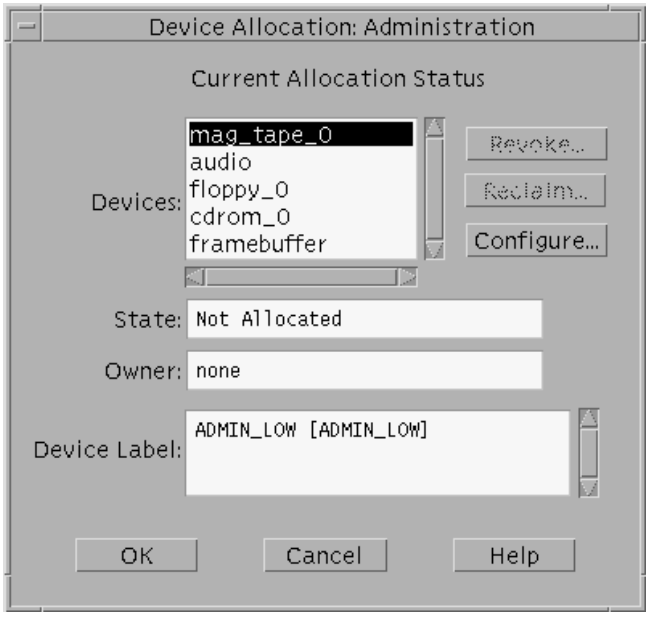


Figure 15–7 Device Allocation: Administration Dialog Box

4. **Check the status of a device by highlighting the name of the device and looking at the State: field.**

5. **If the device is in the "Allocate Error State," go to "To Correct an Allocate Error State" on page 479 to reclaim it.**

6. **If a device is State is "Allocated," do one of the following:**
   a. **Contact the Owner to deallocate the device.**
   b. **Go to "To Forcibly Deallocate a Device" on page 480 to revoke the allocation.**

7. **To configure the device, go to "To Add a New Allocatable or Non-allocatable Device" on page 480.**

▼ To Correct an Allocate Error State

1. **Assume the administrator role and go to an ADMIN_LOW workspace.**
   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **Access the Device Allocation: Administration dialog box.**

   If needed, see "To Access the Device Allocation Administration Dialog Box" on page 477.

3. **Highlight the name of the device.**

4. **If the State field is "Allocate Error State," click the Reclaim button to correct the error state.**

5. **Click OK to save the changes and close the dialog box.**

▼ To Forcibly Deallocate a Device

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

   See "To Login and Assume an Administrative Role" on page 15, if needed.

1. **Access the Device Allocation: Administration dialog box.**

   If needed, see "To Access the Device Allocation Administration Dialog Box" on page 477.

2. **Highlight the name of the device.**

3. **If the State field is "Allocated," click the Revoke button to force deallocation of the device.**

4. **Click OK to save the changes and close the dialog box.**

▼ To Add a New Allocatable or Non-allocatable 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 Xylogics or an Archive tape drive can use the default `st_clean` script as is, or it 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 487.

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

See “To Login and Assume an Administrative Role” on page 15, if needed.

3. **Bring up the Add Allocatable Device action.**

   “To Launch Administrative Actions” on page 29, if needed.

4. **Use the Add Allocatable Device action to create or update an entry for a device in the device allocation databases and create an ancillary file.**

   a. **Enter the name of the device.**

   b. **Enter the device type.**

   c. **Enter the pathname for all the device special files associated with the device separated by spaces.**

   d. **Save and quit the action.**

5. **To change the default settings for the label range, the pathname of the device clean script, whether the device is allocatable, or which authorizations are required for allocation, use the Device Allocation: Administration dialog box.**

   If needed, see “To Access the Device Allocation Administration Dialog Box” on page 477.

   The following table shows the default values in effect when a device is added using the Add Allocatable Device action or when no values are specified for a device created using the add_allocatable(1MTSOL) command.

TABLE 15–7 Default Values for Devices

Value	Default
minimum SL	ADMIN_LOW
maximum SL	ADMIN_HIGH
clean script	`/bin/true`
allocatable by	All Users
authorizations	None

▼ To Configure an Existing Device

Note - To be managed by the Device Allocation Manager, a device needs an entry in the device_maps(4TSOL), device_allocate(4TSOL) files and an ancillary file in `/etc/security/dev`. If needed, follow the procedure in "To Add a New Allocatable or Non-allocatable Device" on page 480.

---

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

   See "To Login and Assume an Administrative Role" on page 15, if needed.

2. **Access the Device Allocation: Administration dialog box.**

   If needed, see "To Access the Device Allocation Administration Dialog Box" on page 477.

3. **Select the name of the device you want to configure, and click the Configure... button.**

   The Device Allocation: Configuration dialog box. displays as shown in Figure 15–8.

4. **Change the minimum sensitivity 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.**

5. **Change the maximum sensitivity 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.**

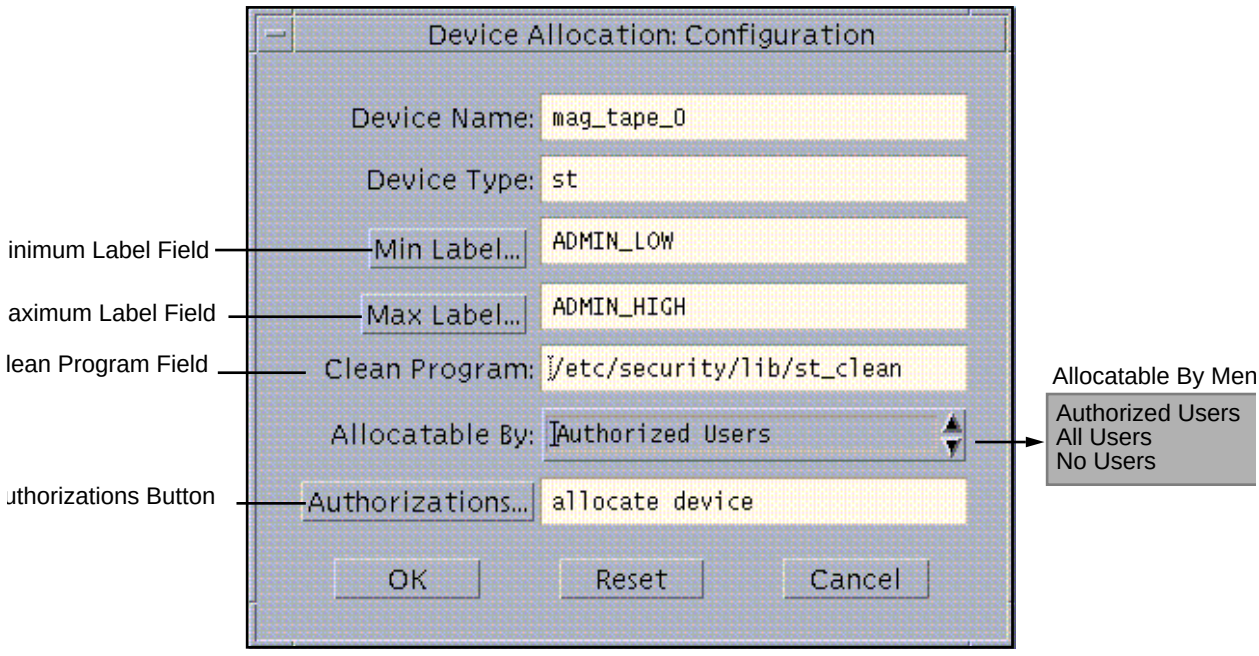


Figure 15–8 Device Allocation: Configuration Dialog Box

6. **Change the name of the device_clean(1MTSOL) script, if desired, by entering a new pathname in the Clean Program field.**

7. **Change the Allocatable By field, if desired, by using the right mouse button to drag down the menu and select one of the following three choices:**

```
Authorized Users
   All Users
   No Users
```

Caution - 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 button becomes active, and the default displayed in the Authorizations field is *allocate device.*

8. **Add a new device allocation authorization, if desired.**

   To add a new authorization, see Chapter 2," under "Extending Extendable Security Mechanisms" on page 44. Once a new authorization is added to the

system, the authorization displays in the Not Required list the next time the Device Allocation: Authorizations dialog box is launched. The Authorizations button is only active when the device is specified as Allocatable By Authorized Users. The default in the authorizations field is *allocate device.*

9. **Change the authorizations from the default of *allocate device,* if desired.**
   a. **Click the Authorizations button.**
      The Device Allocation: Authorizations dialog box displays.
   b. **Remove the default allocate device authorization, if desired, by selecting it in the Required list and using the left arrow to move it to the Not Required List.**
   c. **Add an authorization or replace a removed authorization, if desired, by selecting an authorization in the Not Required list and using the right arrow to move it to the Required list.**
   d. **Click OK to save the changes and close the Authorizations dialog box.**

10. **Exit the Device Allocation Manager.**
    a. **Click the OK button on the Configuration dialog box to save the changes and close the dialog box.**
    b. **Click the OK button on the Administration dialog box to close it.**
    c. **Click the arrow in the upper left corner of the Device Allocation Manager to close it.**

11. **If desired, configure a non-default policy for the device.**
    See “To Set Device Policy on a New Device or Modify Policy on an Existing Device” on page 475,” for how to change the default policies.

▼ To Assign Device-related Authorization(s) to an Account

1. **Assume the security administrator role and bring up the User Manager.**
   See Chapter 1,“ “To Launch Solstice Administration Tools” on page 27, and Chapter 5,” in this manual if needed.

2. **Click the Profiles button to open the profiles dialog, and make sure that the desired device allocation authorization or other device-related authorization(s) are contained in one of the user's execution profiles.**

   a. **Move a profile containing the device allocation authorization to the Available list, if desired.**

      If the defaults have not been modified, including one of the profiles shown in the following table in the user's list of profiles gives that user the allocate device authorization.

TABLE 15–8 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

   b. **Move a profile containing the revoke or reclaim devices authorization to the Available list, if desired.**

      Note - The revoke or reclaim devices authorization is in administrative profiles, which should be given only to administrative role accounts.

      The following table shows the revoke or reclaim devices authorization and the default profiles that contain it.

TABLE 15–9 Device Deallocation and Reclamation Authorization, Default Profiles that Include It, and

TABLE 15–9 Device Deallocation and Reclamation Authorization, Default Profiles that Include It, and Default Roles Assigned It *(continued)*

Default Roles Assigned It

Authorization Purpose	Authorization Name	Default Profiles	Default Role Assigned Profile
Forcing deallocation of an allocated device, or correcting a device's allocate error state	revoke or reclaim device	Device Management	admin

c. Move a profile containing the configure device authorization to the Available list, if desired.

Note - The configure device authorization is in administrative profiles, which should be given only to administrative role accounts.

The following table shows the configure device authorization and the default profiles that contain it.

TABLE 15–10 Device Configuration Authorization, Default Profiles that Include It, and Default Roles Assigned It

Authorization Purpose	Authorization Name	Default Profiles	Default Role Assigned Profile
Configuring device's attributes: device_clean script, label range, required authorization(s)	configure device	Device Security Printer Security	secadmin secadmin

Note - If none of the default execution profiles are appropriate for the account being reconfigured, the security administrator 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 Chapter 8," in this manual.

▼ To Prevent Automatic Display of File Manager After Device Allocation

1. **Assume the administrator role and go to an ADMIN_LOW workspace.**
   See "To Login and Assume an Administrative Role" on page 15, 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 29, 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.

See Figure 15–8.

e. Change the name of the device_clean(1MTSOL) 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.

CHAPTER 16

Adding Software

This chapter covers these main topics:

This chapter provides these procedures:

Review of Terms and Concepts

Security administrators can add the following types of software:

- Sun unbundled products and third-party applications that neither understand nor enforce Trusted Solaris security policy
- New programs, created by in-house developers 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)
- Shell scripts (created or approved by the security administrator)

Security administrators may also add to or modify the commands that run during start-up 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 provides guidance on how the security administrator does the following:

- Brings in new software
- If the software needs privileges in order to run, finds out what privileges the software needs

- Decides whether giving any needed privileges to the new software is consistent with the site's security policy, and
- Makes privileges available to the software
- Makes privileges available to commands in run control scripts

  See the *Trusted Solaris Developer's Guide* for more about how programs use privileges.

  As configured in the default system, the security administrator role is the only account that is able to do the following:

- Import and export software at multiple sensitivity labels
- Install software at ADMIN_LOW in the public directories (such as `/etc` and `/usr/bin`) that allow use of the software by multiple users at all sensitivity labels.
- Assign privileges to program files, although this ability may be given to other accounts by assigning the needed authorization.
- Use the Profile Manager to assign privileges that are in effect when a command or action is executed in a trusted process from which it can inherit privileges.

Note - Because applications, whether they are externally or internally obtained, and shell scripts are added to a site's execution profiles as commands, the term *command* is used frequently in this chapter when referring to applications, site-developed executable programs, and shell scripts.

See "Privileges" on page 492," and the following sections, which define what it means for a command to have privileges and for a command or action to inherit privileges.

Assuming that a site has procedures to screen imported software for viruses and worms, the security administrator, 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:

- It does not need to run with privilege
- It does not need to run with an effective UID or GID that differs from the real UID or GID of the invoking user
- It does not need to run at multiple labels

Controls for Software Creation and Use

The security administrator controls beforehand *who can create* programs, actions, and shell scripts by controlling what commands and actions are made available to individual accounts through the profile mechanism. For example, if a user or role account is not allowed to use the Create Action action or some other tool that would allow the editing of files that create an action, that account simply cannot create an

action. And, even if an account can create an action, the account would not be able to use any new action that he or she might create, without the security administrator's cooperation, unless the account already has the All profile, which turns off all checks for the use of commands or actions. To allow anyone without the All profile to use a new action, the security administrator would have to add the action to site's profiles. The All profile also turns off all checks for commands, and would enable an account to run any standard shell and enter any existing or new command or script.

Programs and shell scripts are controlled in a slightly different manner than actions. If an account is allowed to use a terminal or an application that allows him or her to create a new program or shell script, the resulting new program or shell script can be executed by the creator or by anyone who has access to the software and who has a standard UNIX shell. For accounts (such as secadmin and admin accounts) that have the profile shell and no other shell assigned, the security administrator would need to add the name of the new command to one of the account's profiles in order for the account to use it, unless the account also has the All profile.

Controls for Importing Software

Similar mechanisms are in place for controlling the importing of programs, actions and scripts as there are for creating them. The security administrator controls *who can bring in software* by granting or denying the device allocation authorization to an individual. If an account has the device allocation authorization, the account is then restricted to importing or exporting data at a single sensitivity label within that user's clearance.

Privileges

Having privilege means having the capability to override some aspect of security policy.

In standard UNIX systems, only commands running with the user ID of 0 can run with all the powers of the superuser, while commands running with any other user ID have none. In the Trusted Solaris system, however, a command with any user ID may be configured to use some but not all of powers that are reserved for the superuser in standard UNIX system. The ability of the UNIX super-user to bypass access restrictions, to execute restricted commands, and to use some command options not available to other users has been replaced with privileges. The superuser in standard Solaris operating environment and other UNIX operating systems has all privileges in the system, including the ability to bypass all DAC restrictions. The privileges mechanism in Trusted Solaris breaks the power of the superuser into many small pieces and provides additional power that may be needed to bypass MAC restrictions. Providing discrete privileges makes it possible to assign to programs only those privileges they need to do their work and no more.

Required Privileges

When a command or one of its options or an action needs a privilege in order to succeed, that privilege is a *required* privilege; if a required privilege is not available to a command, option or action, it will not work at all. Required privileges are indicated on the man page with the words “must have” as shown in this sentence: “The ifconfig(1MTSOL) command must have the sys_net_config privilege to modify network interfaces.”

Override Privileges

When a command or action is designed to work within security policy, and then it fails when certain DAC or MAC checks are not passed, an *override* privilege may be assigned at the security administrator’s discretion. On man pages, the names of privileges that may be used to override access restrictions are given in the ERRORS section.

The DAC override privileges are file_dac_read and file_dac_write. The MAC override privileges are file_mac_read and file_mac_write. If a user does not have DAC or MAC access to a file, the security administrator may assign one or both of these privileges to a command or action, depending on whether MAC or DAC applies and on whether read or write access or both are desired.

Alternatives to Assigning Privilege

Besides being able to assign an override privilege to a command or action to make it work, the security administrator has other options:

- Assign an effective UID
- Assign an effective GID
- Assign a sensitivity label at which the command executes

  For example, to avoid the use of privilege the security administrator may configure a command or action to execute with another user’s ID or an alternate group ID that allows access to the file or directory based on its permissions or its ACL. When software executes with another UID or GID than the *real UID or GID* of the person who invoked it, it is said to be running with an *effective UID or GID*. For another example, the command or action may be specified to run at a sensitivity label that removes the need to give it a MAC override privilege.

Principle of Least Privilege

The *principle of least privilege* requires that each command or action in the system be granted the most restricted set of privileges that it needs to do the required task for

the person who is using it. The principle of least privilege is supported by making privileges available to programs on an as-needed basis, and by using privilege bracketing, which is used within programs to turn the use of the program's privileges on and off so that privileges are only in effect while they are being used for a specific purpose during the run time of the program. See the *Trusted Solaris Developer's Guide* for more about privilege bracketing.

File Privilege Sets

Executable program files can have two *file privilege sets* associated with them: *forced* and *allowed*. The *allowed* privilege set determines which privileges *may* be made available to a program, while any privileges in a program file's *forced* privilege set are always made available to the program, as long as the privileges are also in the allowed set, no matter what shell the program is executed in or who invokes it.

How Two Standard Programs Use Privilege in Trusted Solaris

The ways that ls(1) and mount(1MTSOL) use privileges illustrates some of the principles talked about so far in this chapter.

ls(1B) is an example of a well-known Trusted Solaris program that does not need privileges in order to work, as long as it is used to list files that DAC and MAC restrictions allow it to see. Therefore, `ls` has no *required* privileges. However, the security administrator could assign *override* privileges if he or she wants to allow `ls` to bypass DAC or MAC security policy for some reason.

When executed by a normal user for the purpose of seeing which file systems are mounted on that user's machine, mount(1MTSOL) needs no privileges. When used to mount file systems, `mount` always needs the privilege sys_mount (so sys_mount is a *required* privilege for `mount`). Depending on whether it is being used to mount file systems remotely or to specify mount-time security attributes, `mount` also may need several other *override* privileges. In the default configuration, the executable program file for `mount` is specified with all allowed privileges and no forced privileges. The System Management profile (which is assigned to the system administrator role) has mount defined with all the privileges `mount` needs to inherit when mounting file systems. See the mount(1MTSOL) man page for more details of mount's privilege requirements.

Actions

Actions are a feature from the CDE window system that are also used extensively in the Trusted Solaris window system, with a number of restrictions on their use and

creation that are necessary to make them compatible with the Trusted Solaris security policy. The restrictions on their use and creation are described in “When Adding Actions ” on page 513.

Actions are instructions that are written to automate tasks such as opening files for editing or running applications. Actions are defined much like application macros or programming functions. Each action is defined with a name that is used to run the action. Actions can be attached to icons, Front Panel controls, and menu items.

Effects of the Execution Profiles on the Use of Commands and Actions

The security administrator may *enable* an account to bypass system security policy by specifying one or more of the following in the account’s execution profiles:

- Authorizations
- Privileges assigned to specific commands or actions
- Effective UIDs or GIDs assigned to specific commands or actions

  The privileges specified in execution profiles are made available to commands and actions by inheritance. Commands can inherit privileges and can run with an effective UID or GID or a sensitivity label specified in one of the invoking account’s profiles only when they are invoked in a *profile shell* or in a *system shell*. Actions can inherit privileges and can run with an effective UID or GID or a specific sensitivity label only when they are launched from one of the trusted processes in the window system (which are described in “The Profile Shell, the System Shell, and Trusted Processes” on page 496).

  A user account may be given the All profile that bypasses all checks on the use of actions or commands, and then the account can use *all actions or commands without inheritable privileges*—giving that account the freedom of access available to a standard UNIX user.

  If an account has a standard UNIX shell, either as the default shell or listed in one of its profiles, the user can execute *all commands* in that shell (without inheritable privileges) but may not be able to launch all actions, depending on how actions are configured in the account’s profiles. A user or role account is *restricted* to use only the set of actions that are explicitly specified in that account’s execution profiles (unless one of the profiles is the All profile). Commands are handled differently; only if an account is given a profile shell as the default shell is the account is restricted to use only those commands that are listed in the account’s profiles

  Each account may have multiple profiles. The order in which profiles are assigned is important. The profile shell and trusted processes in the window system search the profiles in the order specified for the account in the User Manager. If more than one profile is assigned to an account, the order is significant because the first time the profile shell or trusted process finds the command or action in any

profile, the command or action is given whatever security attributes have been assigned to it in the first profile where it appears.

For example, Table 16–1 shows user account roseanne has All, ProfileA, ProfileB, and ProfileC, in that order, in its set of profiles.

TABLE 16–1 Example of an Incorrect Ordering of Profiles in an Account's List

Account name	Default Shell	Profiles	Command
roseanne	pfsh	All	all commands with inheritable=none
		ProfileA	`command1` with inheritable=1,2,3
		ProfileB	`command1` with inheritable=2,3,5
		ProfileC	`command1` with inheritable=7,9, 12

If roseanne executes `command1`, the profile mechanism finds the All profile first, which allows the use of any command without special attributes, and stops there, allowing roseanne to execute the command without giving it privileges or other special attributes. If the task that roseanne needs to do requires that `command1` has privileges 7, 9, and 12 in order to succeed, the security administrator should move ProfileC to the top of roseanne's list of profiles.

If the All profile is assigned to an account, it should be at the bottom of an account's list of profiles, where it can act as a type of fallback mechanism for commands, allowing the account to use any command without privileges, effective UID or GID or specified sensitivity label, if the command is not otherwise explicitly specified in one of the account's other profiles.

The Profile Shell, the System Shell, and Trusted Processes

The profile shell, pfsh(1MTSOL), is the default shell for the security administrator and the system administrator roles, and the `pfsh` may be assigned, at the discretion of the security administrator, as the default shell for any other user or role account. Any account working with the profile shell is restricted to use only *commands* that are *specified in the account's execution profiles.*

The system shell, sysh(1MTSOL), is a shell that is used to control the use of privileges by commands run from *run control* (`rc`) scripts. `sysh` allows any command to execute but consults profiles for any privileges, effective user ID,

effective group ID, and sensitivity label with which the command is to be run. See " Starting Commands During Boot" on page 519 for more about `sysh`.

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

Trusted processes in the window system are available to everyone, but accounts are restricted so they can access from the window system only the *actions* that are *specified in the account's execution profiles*. For example, the administrative actions that are in the System_Admin folder in the Application Manager, can only be used if they are in one of the account's profiles. For example, because Edit Encodings is in the Object Label profile and the Set Mount Points action is not in any of the profiles assigned to the security administrator role, the security administrator can use the Edit Encodings action, but he or she 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 comes up 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 cases, if the action is in the user's profiles, it is run with any specified privileges or an effective UID or GID.

Processes, Programs, and Their Privileges

A process is derived from a file containing an executable program. At any instant, a process is always executing a program. A process is created using the fork(2TSOL) system call, which creates a duplicate of the calling process. The new process inherits all the parent's attributes, including the currently executing program. A process can change the program it is executing with the exec(2TSOL) system call, which replaces the entire process address space with new versions derived from the program file named in the `execve` call.

For example, the process running the profile shell, `pfsh`, uses `execve` to execute the `mount` command.

Process Privilege Sets

Four privilege sets are associated with each process:

- *Permitted Privileges*

  The privileges in a process's *permitted privilege set* are those the process is *allowed* to use.

- Effective Privileges

  The privileges in a process's *effective privilege set* are those a process is *currently using*. System calls that check for privilege check a process's set of effective privileges.

- Inheritable Privileges

  The privileges in a process's *inheritable privilege set* are those a process *may pass to another process* when another program is invoked with an exec(2TSOL).

- Saved Privileges

  The privileges in a process's *saved privilege set* contains the set of privileges *actually inherited* across an `execve`. The contents of the saved privilege set become invalid if a process changes its effective user ID.

  When a process executes a program through the execve(2TSOL) system call, the permitted (P) and effective (E) privilege sets are reset equal to the same value, which is the intersection of the process' previously existing inheritable (I) privileges and the program file's allowed (A) privileges intersected with the program file's forced (F) privileges:

```
P=E=(I[process] union F[program] restricted by A[program])
```

The saved privilege set is set initially to the intersection of the existing inheritable privilege set and the file's allowed privileges:

```
S=(I[process] intersected by A)
```

Keeping the saved privilege allows the process to determine which privileges it had when the currently executing program was invoked.

When a new program is invoked, the inheritable privilege set is initially set to be the same as the inheritable privileges of the process that invoked the current program:

```
I[new]=I[old]
```

See "Why Inheritable Privileges Are Important " on page 502 for the benefits of setting the inheritable privileges without reference to the forced or allowed privileges on an executing program.

If the effective UID is set by setuid(2TSOL), to be different from the original, the effective set is copied to the saved set and the effective set is cleared:

```
S=E; E=0
```

If the process changes its effective user ID back to the original, the saved privilege set is copied to the effective set, thus restoring its privileged state:

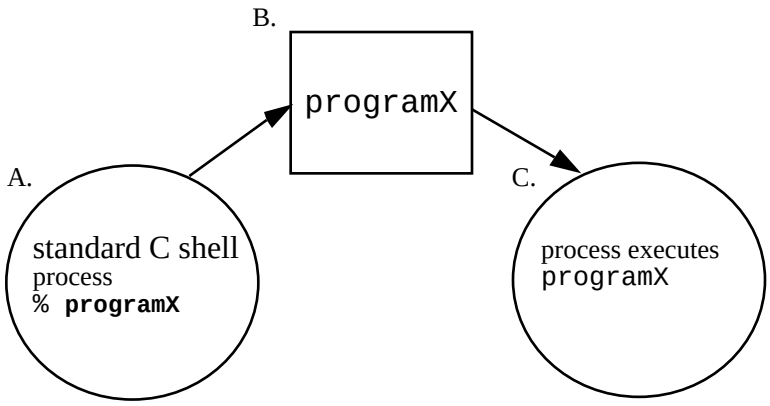
```
E=S
```

Examples of How Processes Acquire Privileges

The following examples show how a process acquires privileges in a standard UNIX shell compared to how a process acquires privileges in a profile shell when a program is executed. The first example shows that when a program executes in a standard UNIX shell, the program's process can use only privileges that are in the program's forced and allowed privilege sets. The second example shows that when a program executes in a profile shell, the program's process can acquire privileges from the shell's inheritable set, which it can use if the privileges are also in the allowed privilege set of the program itself. In the examples, for brevity, the privileges are assigned numbers instead of names.

In a Standard Shell

The circle on the left side of Figure 16–1 stands for a standard UNIX shell's process in which the user enters the command `programX`. The inheritable privilege set shown for the shell's process is null because standard shells' processes have no privileges.



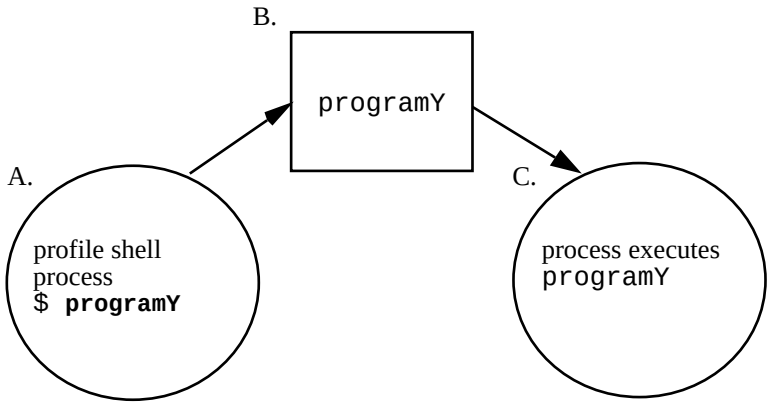
A.
standard UNIX shell process with inheritable=none,
B.
programX file's privilege sets: forced=1,3,5; allowed=1,3,5,11,12,19
C.
process running programX now has permitted=1,3,5; effective=1,3,5;
inheritable=none

Figure 16–1 Process Acquiring Forced Privileges when Run in a Normal User's Shell

In Figure 16–1, the square in the middle stands for the programX, which has forced and allowed privileges as shown. The forced privileges are 1, 3, and 5. The allowed privileges are 1, 3, 5, 11, 12, and 19. The circle on the right stands for the process executing programX in which the permitted and effective privileges sets are given the forced privileges associated with the program file. Figure 16–1 illustrates that when the parent process is a standard shell, and therefore it has no inheritable privileges, the privilege sets associated with a new process contain only the forced privileges acquired from the program file, and the inheritable set of the new process remains empty.

In a Profile Shell

In Figure 16–2, a user or a role account executes programY in a profile shell.



A.
In one of the invoking account's profiles, `programY` has inheritable=10,12,19, so `pfsh` sets its own inheritable=10,12,19 and executes `programY`.
B.
`programY` file's privilege sets: forced=none, allowed=10,12,19
C.
process running `programY` now has inheritable=10,12,19; effective=10,12,19; permitted=10,12,19

Figure 16–2 Process Inheriting Privileges from the Profile Shell

In Figure 16–2, the profile shell checks which user or role is executing the command, checks to make sure the command is in one of the account's profiles, checks to see if any privileges have been specified for the command in any of the account's profiles, and puts only the specified privileges into its own inheritable set. As shown in the illustration, the executable `programY` itself has no forced privileges, and its allowed set includes 10, 12, and 19. Because `programY` has the three privileges 10, 12 and 19 in its allowed set, the process running `programY` can inherit privileges 10, 12 and 19 from the parent process, and privileges 10, 12, and 19 are put into the process's effective and permitted sets.

If there are no privileges in a program file's allowed set, then the permitted and effective sets for the process running the program always will be empty.

How a Process Executing the `mount` Command Acquires Privileges

This section shows how the `mount` command used in the example in "How Two Standard Programs Use Privilege in Trusted Solaris" on page 494 is executed in a profile shell process and acquires privileges. When `mount` is executed in the profile shell, the process running `pfsh` does the following:

- It checks that the invoking account has the mount command in one of its profiles

- It puts the sys_mount privilege and the other privileges that are specified for mount in the invoking account's profiles into its own (`pfsh`'s own) inheritable set
- It runs `execve`, which creates a process running the `mount` command

  In `mount`'s process, the *inheritable* privileges are set to be the same as the inheritable privileges of the profile shell's process that invoked `mount`. Then the forced and allowed privileges on `mount`'s program file are used in conjunction with the inheritable set in determining the process's permitted, effective, and saved sets.

  To determine the *permitted* and *effective* set, a process's inheritable privileges are first combined with any additional privileges from the forced set on the program file—which does not apply here since `mount` has no forced privileges. The combination of both the inheritable and forced privileges is limited to the allowed privileges specified for the program file, and because the `mount` program file has all allowed privileges, `mount` can use all of its inheritable privileges, and, as a result, the permitted and effective sets of the process running mount are set equal to the inheritable privilege set.

  `mount`'s *saved* privilege set is set to be the inheritable set and the forced and allowed privileges on mount's program file.

Why Inheritable Privileges Are Important

Inheritable privileges are discussed in detail under "Process Privilege Sets " on page 498. Inheritable privileges are important for security administrators 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
- Trusted processes in the window system to pass privileges to actions

  As described in "Process Privilege Sets " on page 498, 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[new]=I[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[new]=I[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.

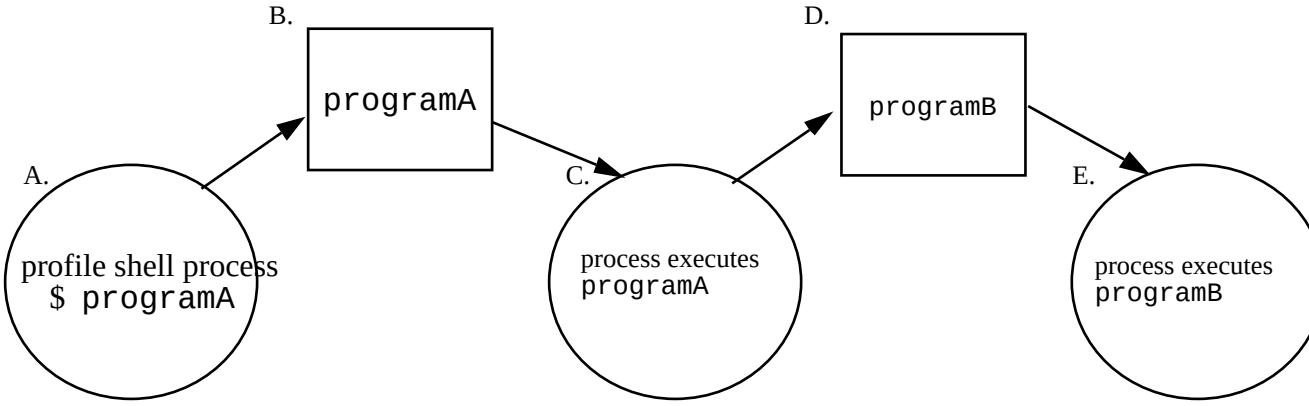
  See "When a Program File Has No Allowed Privileges" on page 503" for details.

- 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 503 for details.

When a Program File Has No Allowed Privileges

When a program file has does not have allowed privileges, the inheritable set of the process executing the program is not reduced to match the allowed privileges on the program. 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. See Figure 16–3, Figure 16–3.”



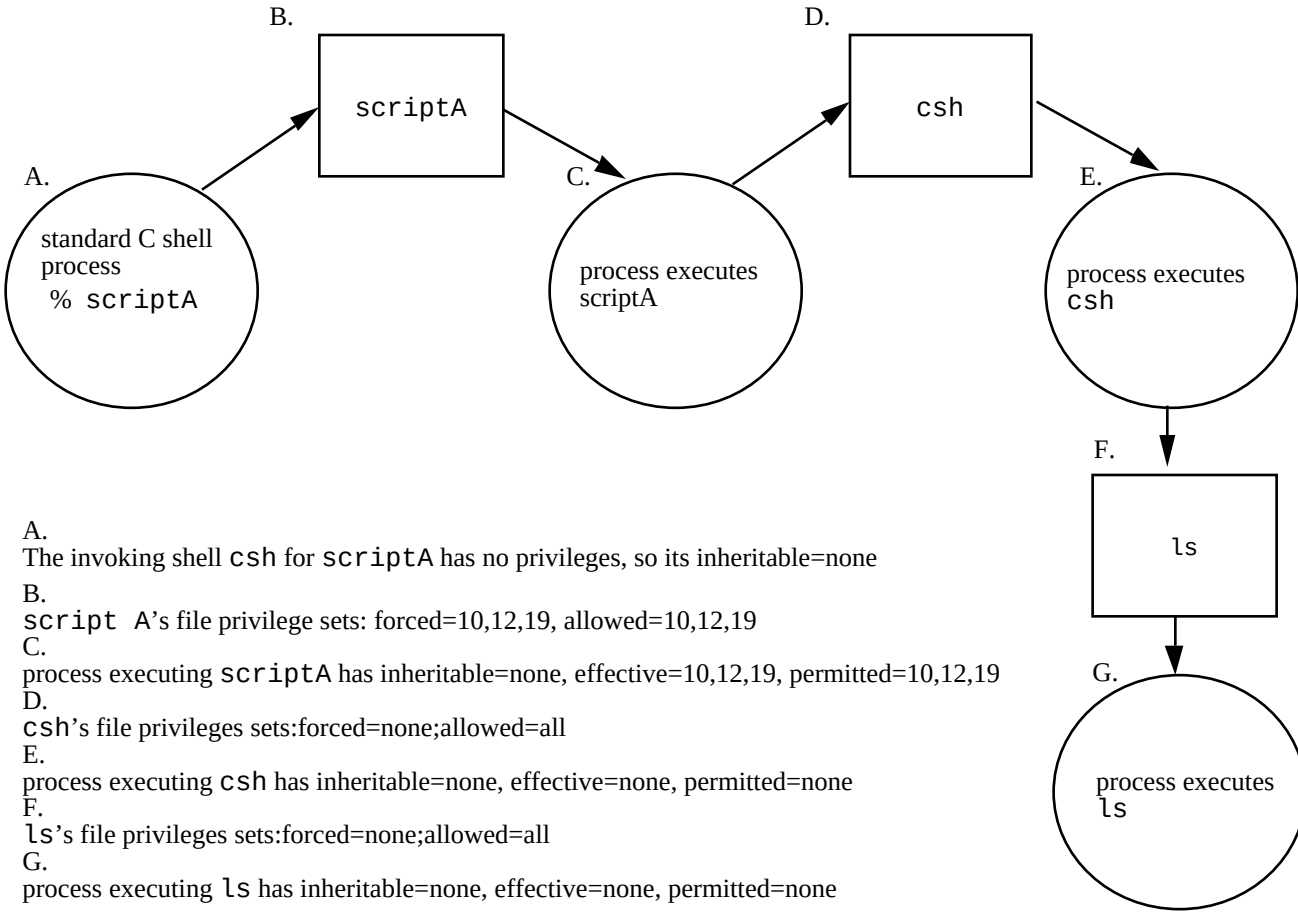
A.
In one of the invoking account’s profiles, `programA` has inheritable=10,12,19, so `pfsh` sets its own inheritable=10,12,19 and executes `programA`.
B.
`programA` file’s privilege sets: forced=none, allowed=none
C.
process executing `programA` has inheritable=10,12,19; effective=none
D.
`programB` file’s privileges sets: forced=none;allowed=10,12,19
E.
process executing `programB` has inheritable=10,12,19;effective=10,12,19

Figure 16–3 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 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 Figure 16–4, Figure 16–4."



A.
The invoking shell `csh` for `scriptA` has no privileges, so its inheritable=none
B.
`script A`'s file privilege sets: forced=10,12,19, allowed=10,12,19
C.
process executing `scriptA` has inheritable=none, effective=10,12,19, permitted=10,12,19
D.
`csh`'s file privileges sets:forced=none;allowed=all
E.
process executing `csh` has inheritable=none, effective=none, permitted=none
F.
`ls`'s file privileges sets:forced=none;allowed=all
G.
process executing `ls` has inheritable=none, effective=none, permitted=none

Figure 16–4 How Forced Privilege Shell Scripts Are Prevented from Passing Forced Privileges to Their Commands

How Privileges Are Assigned to Commands and Actions

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 only if the security administrator is convinced that the command or action itself or the person invoking the command or action will use the privilege in a trustworthy manner.

The security administrator 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 from a trusted process in the window system or by an action when it is launched from a trusted process in the window system.

Giving Forced Privileges to a Command

The security administrator can assign forced privileges to an executable file for a command by using the File Manager—Privileges dialog box or by entering the setfpriv(1TSOL) command in a profile shell, as described under "To Give Forced Privileges to a Command" on page 533.

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 the profile shell as its default shell and by not assigning the command or any terminal emulator in any of that account's profiles.

To change the privileges on an executable file, the process's sensitivity 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 sensitivity label (write-equal) or by a security administrator (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(1TSOL) 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 can specify inheritable privileges for a command or an action in an execution profile (using the Profile Manager) and assign the profile to a user or role account using the User Manager. See Chapter 5," and Chapter 8" for how to use the User Manager and Profile Manager.

Note - For privileges to be made available by inheritance, the privileges must be available in the command's allowed privilege set.

Why Privileged Programs Need to Use Trusted Shared Libraries

Most applications use shared library routines. The Trusted Solaris security administrator needs to make sure that shared libraries used by any application that requires privilege(s) are in a trusted directory. The dynamic linking of privileged applications to shared libraries is restricted—to ensure that privileged applications can never use untrusted libraries. A privileged application that tries to link to an untrusted library fails with an error.

Default Trusted Shared Library Directories

The default shared libraries are stored in the standard directories used by privileged programs in the base Solaris operating environment, which are listed in the following table.

TABLE 16–2 Default Directories for Shared Libraries

Trusted C Function Libraries	Trusted Extensions to X Server
`/usr/lib`	`/usr/openwin/lib`
`/etc/lib`	`/usr/dt/lib`

The directories listed in Table 16–2 are protected by mandatory and discretionary access controls (MAC and DAC) to keep anyone but administrators from writing into them or modifying existing library files.

Shared Libraries Used by Third Party or Site-Created Applications

If at all possible, when a third party or site-created application is given privilege(s) by a site's security administrator, any shared libraries on which the trusted application relies should be moved into one of the default shared library directories shown in Table 16–2. The other alternative is to define a shared library's directory as trusted, but this alternative should not be used unless the libraries *cannot* be moved into one of the existing trusted directories. To identify a library directory as trusted, the security administrator lists the directory pathname for a privileged application's library in a `rtld` file in `/etc/security/tsol`.

When running a privileged program, set-user-ID, or set-group-ID program, the runtime linker only searches for libraries in a pathname specified within the executable as a result of runpath being specified when the executable was constructed, or in `/usr/lib`, `/etc/lib`, `/usr/dt/lib`, `/usr/openwin/lib`, and if `/etc/security/tsol/rtld` exists, in the colon-separated list of directories specified in the `rtld` file.

See the ld(1TSOL) man page for information on the link editor for object files and on the `rtld` file.

Note - Any application that is given privileges becomes trusted. The security administrator must make sure that a program that needs privileges is actually worthy of trust. Because any application's libraries listed in `rtld` become trusted, they require the same level of protection as the default libraries so that they cannot be modified inappropriately. The security administrator should ensure that the MAC and DAC permissions on any directory listed in `rtld` are the same as they are on the default trusted libraries.

Security Administrator'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 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 by in-house developers 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)
- Shell scripts (created or approved by the security administrator)

  Some programs run at a single level with no privileges required, so the security administrator can simply install them at ADMIN_LOW in a public directory and assign them as desired as commands in the execution 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.

Issues Around the Adding of Privileges to Any Software

To recap a basic principle of privilege management, software needs privileges or other security attributes only when it needs to circumvent some aspect of the system security policy in order to do its work.

To find out what privileges a program needs, the security administrator can turn on the `tsol_privs_debug` switch in system(4) and use runpd(1MTSOL) as described in "To Find Out Which Privileges an Application Needs" on page 530 to find out what privileges are being requested, but that is only the beginning.

If a program needs to override DAC or MAC restrictions on accessing a file, the security administrator might decide to assign an effective UID or GID to make the privilege unnecessary, as described in "Alternatives to Assigning Privilege" on page 493.

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. The problem is that if you allow software to bypass security policy, you can make software *trusted* even though it might not be *trustworthy*. The security administrator must 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*.

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 can run at all 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 were added unless other code changes were 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 510.

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 “How Privileges Are Assigned to Commands and Actions” on page 504.

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 knowledgable 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 “Why Privileged Programs Need to Use Trusted Shared Libraries” on page 506.

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 that executes with setuid `root`. This kind of program may be assigned an effective UID of root in a profile.

Most applications are written in environments that do not have the security mechanisms needed by evaluated systems at B1 and above such as MAC and information labeling. For this reason, it is necessary that the person who assesses the program thoroughly understands security and thoroughly understands what the new program is trying to accomplish.

Be especially careful when allowing any program to violate Trusted Solaris policies such as MAC and information labeling, which have no analogs in standard UNIX systems. 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 you must assess 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 have the built-in ability to do the same kind 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 enforced 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 code to do the necessary MAC checks.

Some software may need privileges for reasons that are not obvious. 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, the offending policy overrides may be removed without impact on 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 Be Installed as Root

Often the software that installs a particular application or package requires a *real* UID of root in order to succeed. Because the profile mechanism only allows the security administrator to assign an *effective* UID to an application, assigning a UID of root to the installation program in a profile would not fill the requirement. The security administrator can enable this type of application to be installed by doing the steps in “To Set Up an Application to Run with a Real UID of Root” on page 527.”

When Applications Need to Run As Root

When an application has been written to run as root, the security administrator 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

  See “To Set Up An Application to Run with An Effective UID of Root” on page 528.

- 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 530.

- Assign the root role (temporarily or permanently) to the desired account(s) and assign the command to one of the account’s profiles.

  See“To Set Up an Application to Run with a Real UID of Root” on page 527.”

Example: The Use of Allowed Privileges with the `mount` Command

Code Example 16–1 shows the privileges, in hexadecimal, assigned in the `tsolprof` database to the `/etc/mount` program in the XXX profile, which is assigned to the system administrator role. The entry means that when the `/etc/mount` command is entered by the system administrator, the profile shell puts the specified privileges into the shell’s inheritable set, to be inherited by the process executing `mount`.

CODE EXAMPLE 16–1 Privileges for `mount` in `tsolprof`

```
TO BE ADDED
```

Anyone using the profile shell can enter the `clist` command with the `-p` option to display (Code Example 16–2) in ASCII all the privileges that have been assigned to commands in any of the current account's profiles. Code Example 16–2 shows the privileges listed for the mount command the first time it appears in the current account's profiles.

CODE EXAMPLE 16–2 Privileges for `mount` (as displayed by `clist -p`)

```
admin# clist -p
      .
      .
      .
/usr/etc/mount:file_chown,file_dac_execute,file_dac_read,file_d
ac_search,file_dac_write,file_downgrade_il,file_downgrade_sl,fi
le_mac_read,file_mac_search,file_mac_write,file_nofloat,file_ow
ner,file_setdac,file_setid,file_setpriv,file_upgrade_il,file_up
grade_sl,net_mac_override,net_privaddr,net_setil,proc_nofloat,p
roc_setil,proc_setsl,sys_mount,sys_nfs,sys_trans_label
```

As was true for `programY` in Figure 16–2, the profile shell's process puts the specified privileges from the profile into the shell's inheritable set, and the process executing the mount command inherits them. The privileges may be inherited only because they are in the allowed set for the `mount` program.

When Adding a New Trusted Program

Even though program's developer can determine which privileges to put into the program's source code, if the security administrator does not assign a privilege that the program needs, then the program cannot use the privilege. Because both the security administrator and the developer must work together to add privileges to trusted programs, the use of privilege by new programs can be said to be another Trusted Solaris administrative task that is under two-person control. 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 to place shared libraries linked to the program in a trusted directory as described in “Why Privileged Programs Need to Use Trusted Shared Libraries” on page 506 and use the trusted directory location when compiling the program in in “To Allow Trusted Programs to Link to Trusted Libraries” on page 533.

   See the *Trusted Solaris Developer's Guide* for additional instructions on how to use privileges in programs.

Security Administrator‘s Responsibilities

As the security administrator, you 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 530.
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 the base. Adding actions is described in the *CDE Advanced User's Guide and System Administrator's Guide.*

In Trusted Solaris, *use* of actions is controlled by the execution profile mechanism and by MAC. Actions may be assigned inheritable privileges in any execution profile, and they can run with privileges if they are invoked within one of the window system's trusted processes that can pass them privileges from its inheritable set. In the Trusted Solaris system, a number of actions have been assigned privileges in execution profiles that are assigned to certain roles by default. Table 16–3 summarizes the main differences encountered in creating and using actions in the Trusted Solaris system.

TABLE 16–3 Differences in Creating and Using Actions Under Trusted Solaris Restraints

Base CDE	Trusted Solaris
New actions may be created by anyone within the originator's home directory, and a new action is automatically usable by its creator.	An action is only usable by a user or role if the action is one of the account's execution profiles.
	If the Create Action action or commands or actions that permit the editing of files are in an account's profile, the user or role *can* create a new action in the account's home directory, but the account may not be able to use the new action.
	There are two ways a user can use any new action: if the security administrator adds the name of the new action to one of the account's execution 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.
	Only if the account is allowed to use the action by its execution profiles, will the account be able to launch the action from its home directory through the File Manager.
A new action's files may be copied to public directories under `/etc/dt` to make the new actions available to others.	MAC restrictions restrict the copying of new actions to public directories. The public directories where actions are kept are not writable by normal users or non-administrative role accounts, so new actions cannot be moved to these public locations.
Actions can be dragged and dropped to the Front Panel.	The Front Panel is part of the trusted path. The window manager recognizes only the administratively-added actions that are located in `/usr/dt` and `/etc/dt` 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.
The only way that actions can do privileged operations is if they are run by root.	If actions are specified to have privileges in one of the invoking account's execution 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.

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 access a terminal or shell tool in which to run the shell script.

Making privileges available to commands run in shell scripts may be done only by security administrators. To begin with, 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

  .

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, in Trusted Solaris 2.5, 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 to being modified without detection. Before making shell scripts available that use inheritable privileges, the security administrator should keep in mind that the same protection against tampering that is available for programs is not available to shell scripts.

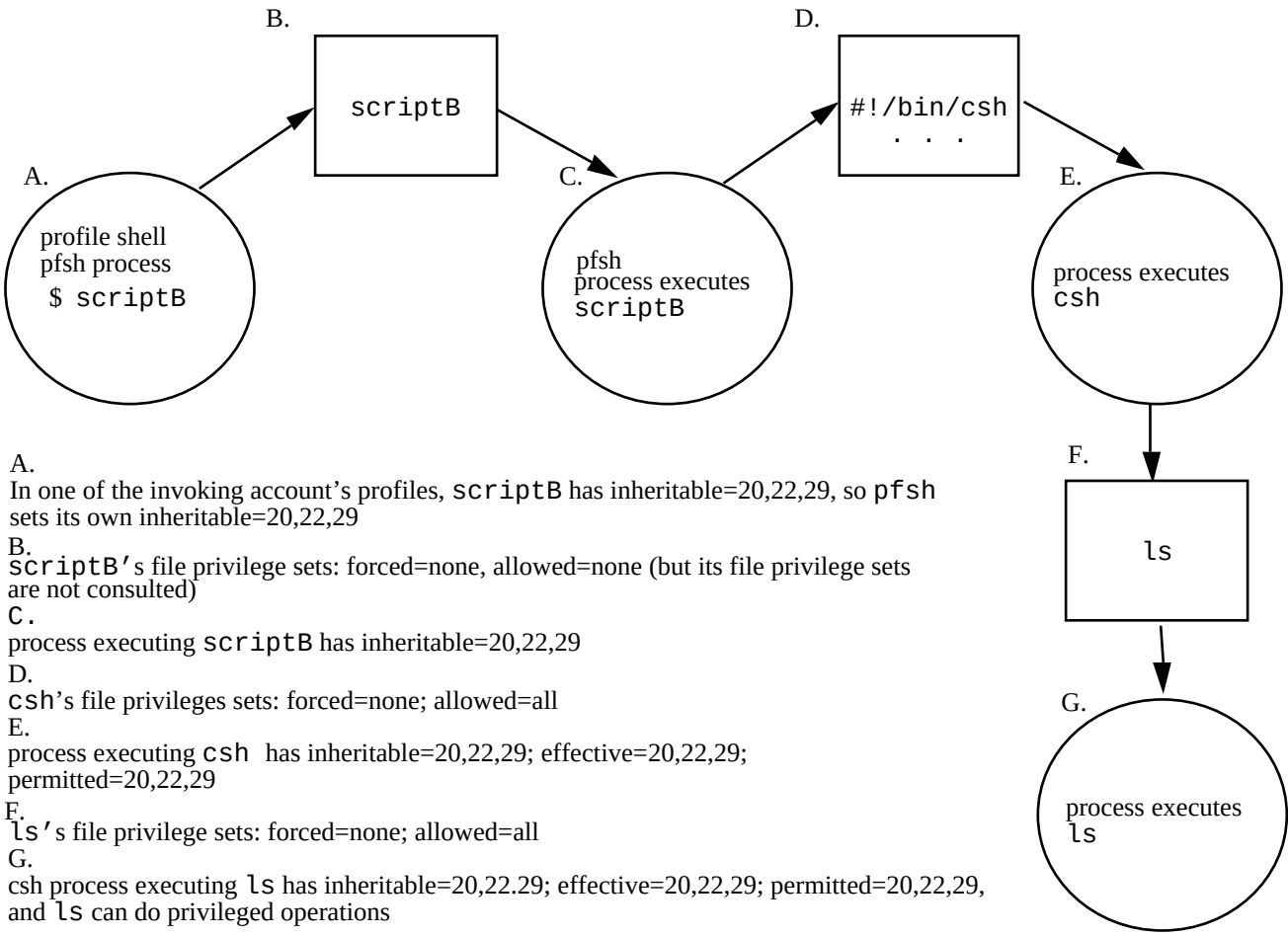
Summary of Shell Script Behavior in Trusted Solaris Systems

- 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 and can used by anyone who has access to the software and who is able to access a terminal or shell tool in which to run 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 its programs 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 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 Figure 16–5 Figure 16–5 on Figure 16–5.



A.
In one of the invoking account's profiles, `scriptB` has inheritable=20,22,29, so `pfsh` sets its own inheritable=20,22,29

B.
`scriptB`'s file privilege sets: forced=none, allowed=none (but its file privilege sets are not consulted)

C.
process executing `scriptB` has inheritable=20,22,29

D.
`csh`'s file privileges sets: forced=none; allowed=all

E.
process executing `csh` has inheritable=20,22,29; effective=20,22,29; permitted=20,22,29

F.
`ls`'s file privilege sets: forced=none; allowed=all

G.
csh process executing `ls` has inheritable=20,22.29; effective=20,22,29; permitted=20,22,29, and `ls` can do privileged operations

Figure 16–5 How Normal Shell Scripts Invoked in `pfsh` Can Pass Inheritable Privileges to Their Commands

- Shell scripts that use the profile shell (that is script that begin with the line `#!/bin/pfsh`) 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 role must have the name of any script using `#!/bin/pfsh` *explictly* listed in the Custom *role_name* Profile.
- A role must run any script using `#!/bin/pfsh` in the profile shell.

The differences exists because controls for administrative roles are stricter:

- Roles cannot execute the profile shell (or bring up a GUI) without the trusted path.
- A script must be explictly listed in a profile for the trusted path to be available.
- 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 explictly listed.
- As is true for normal users, any commands in the profile-shell script have to be in one of the user's profiles, for roles that should be the Custom <role> profile. Commands that need privilege would have to be listed with the privileges they need in the profile.

See "To Write a Profile Shell Script that Runs Privileged Commands " on page 534.

How Edited Program File 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 can save the list of privileges on such a file before editing it and restore them afterwards, as described in "To Restore Privileges Lost when a File is Edited" on page 539.

Starting Commands During Boot

As in the base Solaris system, the commands that run during boot of the Trusted Solaris system may be added to or otherwise modified by administrative action. The base behavior is described in "Run Control Scripts" in the Solaris *System Administration Guide, Volume 1* and on the init.d(4) man page. See the README in each `/etc/rc`*n*`.d` directory for some guidelines about the numbering of the scripts that start system services. See "Background" on page 519 below for a brief review.

The remainder of this section describes what the security administrator needs to do to provide Trusted Solaris extended security attributes needed by services being started during boot. A service may need to inherit privileges, start with a certain sensitivity label or clearance other than ADMIN_LOW, or have a non-root UID or alternate GID assigned. See "Default Trusted Solaris Boot Scripts" on page 520, "Locally-added Trusted Solaris Boot Scripts" on page 520, and see "To Specify Commands to Run with Extended Security Attributes During Boot" on page 538 for the procedure.

Background

During boot, the `/sbin/rc`*n* scripts execute the scripts in the corresponding `/etc/rc`*n*`.d` directories. The number *n* in the run control scripts' names and directories' names stands for the run level.

The scripts in the `/etc/rcn.d` directories are hard links to scripts actually located in `/etc/init.d`. For example, three `sendmail` scripts with different names in the `/etc/rc0.d`, `rc1.d`, and `rc2.d` directories are actually hard links to the `/etc/init.d/sendmail` script, as shown in Code Example 16–3.

CODE EXAMPLE 16–3 `Where /etc/initd.d/sendmail` is Linked to `/etc/rc`*n*`.d Directories`

```
/etc/rc0.d/
 K57sendmail

/etc/rc1.d/

 K57sendmail

/etc/rc2.d/
 S88sendmail
```

A script in the `/etc/rcn.d` directories starts with a S prefix when the script needs to be run with the `start` option, and it starts with a K prefix when it needs to be run with the `stop` option. In run levels 0 and 1, `sendmail` is stopped (as indicated by the K prefix on the filename), and in run level 2, sendmail is started (as indicated with the S prefix on the filename).

After installing a new script into /etc/init.d, the security administrator does the following:

- Identifies where in the boot sequence the script needs to be started and/or stopped
- In one or more /etc/rc*n*.d directories, makes a hard link to the script in /etc/init.d
  - Uses the proper prefix in the target file's name for either starting or stopping
  - Uses the proper numbers in the target file's name to help determine the order in which the script is executed during the run level

    When in the boot sequence a new script should be run determines whether the security administrator puts the command(s) run by the script into the local file or a NIS+ profiles database by selecting none or NIS+ as the naming service. See "To Specify Commands to Run with Extended Security Attributes During Boot" on page 538 for more details.

Default Trusted Solaris Boot Scripts

In the Trusted Solaris system, the /sbin/rc*n* scripts have been modified to use the system shell, sysh(1MTSOL), instead of the Bourne shell, sh(1) when the service being started requires privileges or other extended security attributes. In the default Trusted Solaris system, a *boot* execution profile has been set up to specify extended security attributes for commands started during boot. In the /sbin/rc*n* scripts, /bin/sysh is used without a profile name argument because it looks at the boot profile by default.

Warning - Do not modify the boot profile or the /sbin/rc*n* scripts.

Locally-added Trusted Solaris Boot Scripts

At sites that need to add to the commands running during boot, the security administrator creates system shell scripts that start with #!/sbin/sysh and that specify a locally-created boot-time execution profile (using the setprof command) that assigns to the commands the required security attributes. As shown in the following example, a system shell boot script has #!/sbin/sysh as the first line and setprof local_boot_profile as the second line.

```
#!/sbin/sysh

setprof local_boot_profile
```

The system shell, sysh, consults the local_boot_profile to determine which extended security attributes have been assigned to the command being started in the script.

For example, if a command needs to be started with a sensitivity label or clearance other than ADMIN_LOW, the profile needs to set the minimum SL and the maximum SL or clearance for the command, and if the command needs a UID of root or an alternate GID, the profile needs to specify the required UID or GID.

The security administrator should follow the steps in “To Specify Commands to Run with Extended Security Attributes During Boot” on page 538 to create new profile that has the name of the command run by the script and should use the `sysh` shell in the script, telling the `sysh` to consult the new profile by using `sysh`’s `setprof` option.

Using Scripts in the `/etc/init.d` Directory to Start and Stop Services

In Solaris, the superuser can start and stop any command (also referred to a service) in the `/etc/init.d` directory by entering the name of the command followed by either `start` or `stop`. For example, the superuser can enter:

CODE EXAMPLE 16–4 Starting and Stopping `sendmail` Using the `start` and `stop` Options with the `/etc/init.d/sendmail` Script

```
# /etc/init.d/sendmail start
```

and

```
# /etc/init.d/sendmail stop
```

Stopping `sendmail` and other commands in a Trusted Solaris system as shown in Code Example 16–4 does not require privileges, but the command must be run by the administrator role in an administrative role workspace with the trusted path attribute and the script name must be in one of the account’s execution profiles.

Most commands run during Trusted Solaris boot usually need to inherit privileges when starting, so entering the `/etc/init.d` pathname of a script to *start* a command succeeds only if the script is invoked in a shell that has the needed privileges and other attributes. And the name of the `script` would have to be in one of the account’s profiles with the needed privileges.

See each command’s man page for the privileges that a command needs. For example, the sendmail(1MTSOL) command’s man page states that it needs the net_mac_read, net_privaddr, proc_nofloat, proc_setil, file_mac_read, and file_mac_search privileges to run the options that it needs during boot. The man page states that it needs to run at ADMIN_HIGH with an effective UID of 0, so the command in the above example to start `sendmail` would work only if run with the specified attributes.

Installing the Trusted Solaris AnswerBook

The Trusted Solaris 2.5 AnswerBook can be installed and viewed in both the Solaris and the Trusted Solaris environments. Instructions for both types of installations are documented in “To Install the Packages on the Trusted Solaris AnswerBook CD” on page 540.If desired, see also the Solaris 2.5 *System Administration Guide* Volume 1, which contains more procedures of interest when installing and administering AnswerBooks, including how to set up an AnswerBook server.

The procedures in the base Solaris documents for using and administering AnswerBooks generally apply in the Trusted Solaris environment, with some differences described in the following subsections.

Installation: `swmtool`(1M) Run by the admin Role

Installation of the AnswerBooks in the Trusted Solaris environment should be performed by the administrator (admin) role. The `swmtool`(1M) command is recommended (because it is easier to use), but `pkgadd`(1M) also works. In the default system, both the `swmtool` and `pkgadd` commands are in the Software Installation profile assigned to the admin role. A site that wants to assign the `swmtool` or `pkgadd` command to another role should specify these commands to run with a UID of 0 and with the same privileges defined for them in the Software Installation profile.

Possible Modifications to Execution Profiles or Changes to Accounts

The Trusted Solaris execution profile mechanism and security policy result in the following access limitations:

- The AnswerBook action can be launched from the Help subpanel on the Front Panel only if the account has the AnswerBook action in an execution profile.
- To run the `answerbook`(1) command, an account must have a terminal emulator, such as `dtterm` or `cmdtool`, from which to invoke the command, along with the `answerbook` command itself in an execution profile.
- In the default configuration, only those accounts with the All Commands profile can use the `answerbook` command, and only those accounts that have the All Actions profile can use the AnswerBook action in the front panel.

- To enable an account to view the AnswerBook, the security administrator can assign either of the existing All Commands or All Actions profiles to the account. Alternately, the security administrator can modify one or more execution profiles as described under, “To Add the AnswerBook Command or Action to a Profile” on page 544, and then can assign the modified profiles to accounts as desired, using the User Manager.

Viewing the Trusted Solaris AnswerBook

The Trusted Solaris AnswerBook is viewed either by using the `answerbook`(1) command, `/usr/openwin/bin/answerbook`, or by using the AnswerBook action that calls the command. Versions 2.6 and later of the Solaris environment come with an unbundled package for a follow-on product called AnswerBook 2, which allows viewing of Sun AnswerBooks from a browser such as netscape(1) or hotjava(1). The format of the Trusted Solaris 2.5.1 AnswerBook is not completely compatible with the AnswerBook 2 viewer, and the use of AnswerBook 2 for viewing the Trusted Solaris 2.5.1 AnswerBooks on this CD is discouraged. Visitors to Sun’s external documentation web site `http://docs.sun.com/ab2` can view the Trusted Solaris 2.5.1 AnswerBook but the format and content may not always be correct or complete.

CD Contents

Manuals in the Trusted Solaris 2.5.1 AnswerBook

The Trusted Solaris 2.5.1 AnswerBook contains these twenty-one books:

Trusted Solaris Documentation Roadmap
Trusted Solaris 2.5 Release Notes
Trusted Solaris 2.5 Transition Guide
Trusted Solaris Installation and Configuration
Trusted Solaris Global Index
Trusted Solaris User’s Guide
Trusted Solaris Administration Overview
Trusted Solaris Administrator’s Procedures
Trusted Solaris Label Administration

Compartmented Mode Workstation Labeling: Encodings Format

Trusted Solaris Audit Administration

Trusted Solaris Developer's Guide

Trusted Solaris Reference Manual divided by main section numbers, consisting of the following 9 books:

Man Pages (1BTSOL)), (1TSOL): User Commands

Man Pages (1MTSOL): Administrative Role Commands

Man Pages (2TSOL): System Calls

Man Pages (3CTSOL), (3NTSOL), (3RTSOL), (3TSOL), (3X11TSOL): Library Routines

Man Pages (4TSOL): File Formats

Man Pages (5TSOL): Macros

Man Pages (7DTSOL), (7TSOL): Special Files

Man Pages (9TSOL): Device Driver Interfaces

Man Pages (9FTSOL): Kernel Functions for Device Drivers

Manuals in the Solaris 2.5 User AnswerBook

The Solaris 2.5 User AnswerBook contains these three books:

Solaris User's Guide

Solaris Advanced User's Guide

Solaris 2.5 Introduction

Manuals in the Solaris 2.5 Administrator AnswerBook

The Solaris 2.5 Administrator AnswerBook contains these twenty books:

Binary Compatibility Guide

Direct Xlib User's Guide

Mail Administration Guide

NFS Administration Guide

NIS+ Transition Guide

NIS+ and DNS Setup and Configuration Guide

NIS+ and FNS Administration Guide

OpenBoot 2.x Command Reference Manual

OpenBoot 2.x Quick Reference Card

OpenBoot 3.x Command Reference Manual

OpenBoot 3.x Quick Reference Card

SPARC: Installing Solaris Software

Solaris 1.x to 2.x Transition Guide

Solaris Common Messages and Troubleshooting Guide

SunSHIELD Basic Security Module Guide

System Administration Guide, Volume II

System Administration Guide, Volume I

TCP/IP and Data Communications Administration Guide

Undocumented Messages

x86: Installing Solaris Software

Manuals in the CDE AnswerBook

List of Books associated with the FCS release of AnswerBook 69.2 -

The Common Desktop Environment AnswerBook contains the following fifteen books

Solaris Common Desktop Environment: Motif Transition Guide

Solaris Common Desktop Environment: Installation and System Administration Guide

Introduction to Solaris Common Desktop Environment

Solaris Common Desktop Environment: User's Transition Guide

Common Desktop Environment: Product Glossary

Common Desktop Environment: Desktop Korn Shell User's Guide

Common Desktop Environment: ToolTalk Messaging Overview

Common Desktop Environment: Style Guide and Certification Checklist

Common Desktop Environment: Internationalization Programmer's Guide

Common Desktop Environment: Application Builder User's Guide

Common Desktop Environment: Help System Author's and Programmer's Guide

Common Desktop Environment: Programmer's Guide

Common Desktop Environment: Programmer's Overview

Common Desktop Environment: Advanced User's and System Administrator's Guide

Common Desktop Environment: User's Guide

Manuals in the Solstice AdminSuite AnswerBook

The Solstice AdminSuite AnswerBook contains these two books:

Solstice AdminSuite Print Administration Guide

Solstice AdminSuite User's Guide

Procedures for Adding Software

▼ To Mount a CD-ROM for Adding a Package

1. **Assume the system administrator role, and go to an ADMIN_LOW workspace.**
   See "To Login and Assume an Administrative Role" on page 15 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[ADMIN_LOW] is specified as the sensitivity label and click the OK button.**

   A prompt then appears with the specified label, as shown below:

```
Insert disk into /dev/dsk/c0t6d0s0.

   Make sure disk is labeled:

       ADMIN_LOW[ADMIN_LOW]

   Press RETURN when cdrom_N is ready or ^C to cancel...
```

5. **Insert the CD-ROM into the drive and press Return.**

   A prompt displays, as shown below:

```
Do you want cdrom_N mounted: (y/n)?
```

6. **Answer `y`.**

   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 Set Up an Application to Run with a Real UID of Root

1. **Use the User Manager to assign the root role to the user account or administrative role account that is responsible for doing the installation.**

   See Chapter 5 if needed.

2. **Use the Profile Manager to assign the name of the application program to one of the profiles assigned to the root role.**

   See Chapter 8 if needed.

3. **If desired, use the User Manager to remove the root role from the account when the task is done.**

▼ To Set Up An Application to Run with An Effective UID of Root

1. **Use the Profile Manager to assign the command name of the application program to an appropriate profile and give the command a UID of root.**

   See Chapter 8 if needed.

2. **Use the User Manager to assign the profile with the added command to any user account or administrative role account as desired.**

   See Chapter 5 if needed.

Adding Privileged Shell Scripts

Giving privileges to shell scripts is not safe. Forced privileges do not work: Commands invoked in shell scripts cannot use forced privileges . To give privileges to shell scripts, you would have to add the commands with the needed privileges into a profile along with the shell script and use t the profile shell to run the shell script.

Adding New System Shell Scripts to Boot

The System Shell Script is used for booting. it is described in the Profile Shell Design Spec. New system shell scripts may be added to the rc.files and to the boot profile or a new boot profile may be added.

NOTES:

A system shell script is for situations in which you want different commands within the script to have different privileges. A command's privileges can also be changed as a system shell script progresses. The system shell must run with the trusted path attribute

The name of the system shell is `systsh` and the first line of a script that invokes the system shell is:

```
#!/bin/systsh
```

Before using `systsh`, you should set up the `systsh` database: `/etc/security/systsh.cmds`. The format of each line in the file is:

```
<context>=<command>:<hex priv list>
```

The elements of the file are:

- context - Provides a way of grouping commands. You can choose a context name that is descriptive of the tasks for the commands that share a context. A context

name can appear in many lines with a different command element in each line. Each command that is in a matching context line is part of that context grouping.

- command - A standard command, such as `rm`(1). A command can appear once for each grouping by context. A command can appear in many different lines, as long as each line has a different context element.
- hex priv list - A hex listing of privileges that the command has for the given context.

  Code Example 16–5 is an example of `systsh.cmds` entries. The first two lines are for different contexts and give different privileges to the same command. When a script invokes the context `BEGIN_STUF`, the `rm` command gets the first set of privileges. When the `END_STUF` context is invoked, the `rm` command functions with the second set of privileges. The third line shows the `cp`(1) command in the same context as the first `rm` command. The two commands are now grouped by context and function with the `BEGIN_STUF` privileges whenever that context is invoked.

CODE EXAMPLE 16–5 Sample `systsh.cmds` Entries

```
BEGIN_STUF=/usr/bin/rm:01400000 0 0 0
END_STUF=/usr/bin/rm:00100000 0 0 0
BEGIN_STUF=/usr/bin/cp:01300000 0 0 0
```

You can set or change the context from within the script with the `setsysctx` built-in. The following example sets the context for `BEGIN_STUF`.

```
#!/bin/systsh
setsysctx BEGIN_STUF
```

A system shell script runs with the default context of `boot`, so you must either immediately set the context within the script, or set the context when invoking the script from a command line. Here is a sample command line that invokes the `BEGIN_STUF` context for a script named `SETUP`.

```
isso# /bin/systsh -C BEGIN_STUF -c SETUP
```

Any command can run in a system shell, but only those in the `systsh.cmds` data base run with their given privileges. Commands that are not in the `systsh.cmds` file run without privileges.

▼ To Use `runpd`(1MTSOL) to Determine Which Privileges a Program Needs

1. **Assume the security administrator role and bring up a profile shell at `ADMIN_LOW`.**

   It's safest to test the program's use of privilege on a isolated development system.

   a. **If it is not already on the system, import the application.**

b. Use the `setlabel`(1T) command, if needed, to ensure that the application has a CMW label of ADMIN_LOW[ADMIN_LOW].

c. Use the `chmod`(1V) command to ensure that the program is executable by everyone.

If the application runs the first time or after you have added one or more privileges, go to Step 4 on page 530. If the application fails, go to Step 2 on page 530.

2. As security administrator, decide whether to give the privilege to the program.

If you trust the program to run safely in your environment with the privilege the program needs, go to Step 3 on page 530.

Record all the privileges the program needs to do the task for the destined normal user, administrative or non-administrative role. Decide whether to give the privileges to the program as forced or inheritable privilegs.

If you decide not to make the program available to any users, stop here.

3. As security administrator, use the File Manager to assign the needed privilege to the forced set for the command.

4. Import the application to the Trusted Solaris distributed system at ADMIN_LOW, and put the application in a publicly accessible place.

▼ To Find Out Which Privileges an Application Needs

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

See "To Login and Assume an Administrative Role" on page 15

2. Change the `tsol_privs_debug` setting to 1 in the `/etc/system` file.

a. Use the Admin Edit action to open the `/etc/system` file for editing.

See "To Use the Admin Editor Action to Edit a File" on page 29, if needed.

b. Search for `tsol_privs_debug=` and set the value to `1`.

```
set tsolsys:tsol_privs_debug=1
```

c. Save the changes to the file and close it.

```
:wq
```

3. **Use the Admin Editor action to modify the** `/etc/syslog.conf` **file.**

   a. **Find the lines shown in Code Example 16–6.**

CODE EXAMPLE 16–6 Commented Privilege Debugging Line in `/etc/syslog.conf`

```
# Un-comment the following line to enable privilege debugging with

# the runpd command.  See runpd(1MTSOL).

#kern.debug;local0.debug                                /var/log/privdebug.log
```

   b. **Remove the comment (#) at the beginning of the line that begins** `kern.debug; local0.debug`**.**

```
kern.debug;local0.debug                                 /var/log/privdebug.log
```

   c. **Save the changes to the file and close it.**

```
:wq
```

4. **Shutdown the machine using the Shut Down option on the Trusted Path (TP) menu and reboot from the monitor prompt.**

```
> boot
```

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

6. **On the command line in a** `pfsh` **at the appropriate sensitivity label, enter the** `runpd` **command followed by the full pathname of the command whose use of privilege you want to check.**

Note - To test which privileges a program needs when run as a normal user, you should run `runpd` with a user ID of someone who is not in a role. To do this, you could create an action including `runpd` and assign the EUID of a normal user to that action, and then put the action in a profile for the security administrator role.

Note - The sensitivity label at which to invoke `runpd` depends on who will be using the application you are testing. 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 Code Example 16–7, `runpd` displays the name of the privilege(s) that the program needs in order to succeed followed by the type of access attempted (create in the example) followed by the name of the resource (RAW_SOCKET in the example).

CODE EXAMPLE 16–7 `runpd` Displaying Privilege Needed for a Process to Succeed

```
$ ping redondo
   ping: socket: Not owner

   $ runpd /usr/sbin/ping redondo
   redondo is alive

   runpd: child terminated with a status of 0

   process /usr/sbin/ping pid 1096 lacking privilege net_rawaccess
   to perform create method upon resource RAW_SOCKET (Mar 29 11:39)

   $
```

7. See also the log file for the privilege debugging messages.

A typical privilege debugging log entry looks like the example shown in Code Example 16–8.

CODE EXAMPLE 16–8 Typical Privilege Debugging Entry in `/var/log/privedebug.log`

```
$ cat /var/log/privdebug.log
    Mar 29 12:18:43 crazy unix: DEBUG: /usr/sbin/ping pid 294 lacking
   privilege 36 to 5 79
```

The privilege numbers appear after the word “privilege.” In the example above, the command `ping` is lacking privilege numbers 36. 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.

8. To assign the needed privileges see “To Give Forced Privileges to a Command” on page 533 or and Chapter 8” for how to use the Profile Manager to assign inheritable privileges.

Note - For a command to be able to use either forced or allowed privileges, the privileges must be available in the command’s allowed privilege set.

---

9. To turn off privilege debugging, restore all the changes you made to the `/etc/system` file and the `/etc/syslog.conf` file in on through Step 3 on page 531 on Step 3 on page 531 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 `chown` command in your profiles and are the owner of the file, if you are in the security administrator role, or if you have the *change file owner* authorization, you may use chown(1TSOL) 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(1TSOL) 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 -p file_dac_read,file_dac_write test.priv.file
```

▼ To Allow Trusted Programs to Link to Trusted Libraries

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

   See "To Login and Assume an Administrative Role" on page 15

2. **If possible, move the shared libraries that a privileged application uses into one of the default trusted library directories in the following table.**

TABLE 16–4 Default Directories for Shared Libraries

Trusted C function libraries	Trusted extensions to X Server
/usr/lib	/usr/openwin/lib
/etc/lib	/usr/dt/lib

3. **If the shared libraries for an application that needs privileges cannot be moved to one of the trusted directories, add the directory name for the library to the** `rtld` **file.**
   a. **Use the Admin Editor action to create or open the** `/etc/security/tsol/rtld` **file for editing.**
      See "To Use the Admin Editor Action to Edit a File" on page 29, if needed.
   b. **Specify the pathname of the library directory.**
   c. **Save and quit the** `rtld` **file.**

```
:wq
```

▼ 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 needs to use the Profile Manager to update an appropriate profile to list each of the commands that run within the shell script and to assign the commands any privileges they need. If a new profile shell script needs to be used by a role, all the commands that need privilege must be added to the Custom *role_name* 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 `tfind` 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 530, if needed.

```
#!/bin/pfsh
# Copyright (c) 1993, 1997, 1998 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.
#
# 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 Login and Assume an Administrative Role” on page 15

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 Solstice Administration Tools" on page 27, if needed.

   To continue with the example, to enable root to run the example `cron` script with the needed privileges, the security administrator uses the Profile Manager to update the Custom Root Profile (which is assigned to root by default) 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 privilege.

Caution - Be aware that 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 also 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, because the profile shell executes a command or action with whatever security attributes are specified in the first profile in the account's list of profile. For example, if `tfind` is in the Custom Root Profile with privileges, and if the Custom root profile is the first profile in which `tfind` is found, when the root role executes `tfind` on the command line in a profile shell, `tfind` will inherit the privileges specified in the Custom Root Profile.

▼ 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. **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 530, 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 for the sake of example. A normal shell script would accept the privileges and the filename as arguments and do error checking. Do not use this shell script unless you want to assign the named privileges to an executable file called `executable`, which would then 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 Login and Assume an Administrative Role” on page 15

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 Access the Profile Manager ” on page 233, if needed.

   To enable the script called `autosetpriv` to run with the file_setpriv privilege needed by the `setfpriv` command, the security administrator would use the Profile Manager to update the Custom Secadmin Profile (which is assigned to the secadmin 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 Specify Commands to Run with Extended Security Attributes During Boot

---

Note - Do not change the default boot profile to add commands that need to start with Trusted Solaris extended security attributes during boot. Instead, add a new profile and use the `setprof` command in a new `sysh`(1MTSOL) script, as described in this procedure.

---

1. **Create a new `sysh` script to start and stop the desired command(s).**

   See "Run Control Scripts" in the Solaris *System Administration Guide, Volume 1* for the basics and see the sysh(1MTSOL) man page and " Starting Commands During Boot" on page 519. The first line of the script should be as follows:

```
#!/bin/sysh
```

   Note - Anyone with a text editor can create the shell script.

2. **Use the `setprof` option in the new system shell script to identify the name of an execution profile.**

   The second line of the script should be as follows, with the name of the profile substituted for *new_profile_name*:

```
setprof new_profile_name
```

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

   See "To Login and Assume an Administrative Role" on page 15.

4. **Install the script in the `/etc/init.d` directory and make a hard link it to the desired `/etc/rc`*n*`.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/rc`*n*`.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]nnscriptname
```

5. **Create a new profile to assign Trusted Solaris security attributes to the command(s) in the new script.**

   a. **Bring up the Profile Manager and pick a naming service.**

   See " To Specify Commands in the Profile Manager" on page 241, if needed.

   b. **If the command needs to start before networking services are up, select None from the Naming Service menu on the Profile Manager, so that the profile is created locally in** `/etc/security/tsol/tsolprof`**.**

   c. **If the command needs to start after networking services are up, select NIS+ from the Naming Service menu on the Profile Manager, so that the profile is created in the** `tsolprof` **NIS+ map on the NIS+ master.**

   d. **Use the Profile Manager to create a new profile that lists the command and assigns privileges, UID, GID, sensitivity label or clearance, as needed.**

6. **Shut down the system using the Shut Down option from the TP Menu and enter boot at the monitor prompt to reboot.**

```
> boot
```

▼ To Restore Privileges Lost when a File is Edited

1. **As security administrator, use getfpriv(1TSOL) to list the privileges on the executable file before editing it and direct the output into a temporary file.**

```
$ getfpriv executable_file > tempfile
```

2. **After editing the file, use the File Manager to make the file executable again (if needed) and to restore the privileges listed in the temporary file.**

▼ To Install the Packages on the Trusted Solaris AnswerBook CD

1. **If installing in the Solaris operating environment, log in as root, and go to Step 3 on page 540.**

   If the AnswerBook is being installed on a Trusted Solaris host, skip this step and go to Step 2 on page 540.

2. **If installing in the Trusted Solaris operating environment, log in as a user who is able to assume the admin role, and assume the admin role.**

   See "To Log In and Assume an Administrative Role" in the *Trusted Solaris Administrator's Procedures* manual, if needed.

3. **Ensure that the host has enough disk space.**

   The total disk space required for all the AnswerBooks on the CD is 383 MB. See the following table for the disk space required for each individual AnswerBook.

TABLE 16–5 Trusted Solaris 2.5.1 AnswerBooks with Disk Space Requirements

AnswerBook Name	Package Name	MB
Solaris 2.5 System Administrator AnswerBook	SUNWaadm	51
Solaris 2.5 User AnswerBook	SUNWabe	25
Solaris 2.5.1 Hardware 11/97 on Sun Hardware	SUNWabhdw	29
Solaris Common Desktop Environment User AnswerBook	SUNWdta	64
Solaris Common Desktop Environment Developer AnswerBook	SUNWdtad	49
Trusted Solaris 2.5.1 AnswerBook	SUNWtab	162

   a. **If installing the Common Desktop Environment (CDE) AnswerBooks, make sure that adequate space exists in the `/usr` partition.**

      The CDE AnswerBooks are automatically installed in `/usr/dt/share/answerbooks`. The total needed for both CDE AnswerBooks is 103 MB.

      i. **Use the `df -k` command to determine the space available under `/usr`.**

```
$ cd /usr
   $ df -k
```

ii. If not enough space is available in `/usr`, create an `answerbooks` directory in a partition that has the required amount of space and create a link in `/usr/dt/share`.

```
$ cd /spare
   $ mkdir answerbooks
   $ cd /usr/dt/share
   $ ln -s /spare/answerbooks answerbooks
```

a. If installing the Solaris User, Solaris System Administrator, Solaris on Sun Hardware, or Trusted Solaris AnswerBooks, make sure adequate space exists in the `/opt` partition.

i. Use the `df -k` command to determine the space available in the partition where `/opt` resides.

Note - If `/opt` is not a standalone partition, make sure that the root partition (`/`) has the needed amount of space.

```
$ cd /opt
   $ df -k
```

See Table 16–5 for the size requirements of the AnswerBooks that are typically installed in `/opt`.

ii. If more space is needed, find a directory that has the required amount of space, and make a separate link from that directory into `/opt` for each package being installed.

```
$ cd /opt
   $ ln -s /spare/answerbooks/package_name package_name
```

The screen example assumes that the required amount of space is available for installing the packages into `/spare/answerbooks`. See Table 16–5 for the package names for each AnswerBook.

The following screen example shows links created in the `/opt` directory for SUNWaadm, SUNWabe, SUNWabhdw, and SUNWtab from `/spare/answerbooks`.

```
$ cd /opt
   $ ln -s /spare/answerbooks/SUNWaadm SUNWaadm
   $ ln -s /spare/answerbooks/SUNWabe SUNWabe
```

```
$ ln -s /spare/answerbooks/SUNWabhdw SUNWabhdw
$ ln -s /spare/answerbooks/SUNWtab SUNWtab
```

4. **In an ADMIN_LOW workspace, use the Device Allocation Manager to allocate the CD drive at ADMIN_LOW[ADMIN_LOW], and insert the Trusted Solaris AnswerBook CD-ROM when prompted.**

   See the "To Mount a CD-ROM for Adding a Package" on page 526, if needed.

5. **Bring up a terminal emulator and enter the** `swmtool` **command.**

```
$ swmtool&
```

   The Admintool: Software dialog box displays.

6. **Choose Add from the** `swmtool` **Edit menu.**

   The Admintool: Set Source Media dialog box displays.

7. **Specify the location of the directory where the CD is mounted.**

   a. **Choose CD With Volume Manager on the Software Location menu.**

   b. **In the Mount Point field, type in the full pathname.**

      The following pathname works in both the Solaris and the Trusted Solaris environments:

```
/cdrom/cdrom0
```

   c. **Click the OK button.**

8. **Click the button next to the names of the AnswerBooks displayed on the** `swmtool` **dialog box.**

9. **Answer the prompts that come up.**

   See the following table for recommendations:

TABLE 16–6 Suggested Answers to `swmtool` Prompts

Prompt	Recommended response	Notes
`Select an installation option:`	`2` (heavy)	The heavy option is required.
`Specify the parent of the AnswerBook home directory:`	`/opt`	Enter `/opt` when prompted. (The Common Desktop Environment AnswerBooks are automatically installed in parent directory: `/usr/dt/share/answerbooks`.)
`This package will be installed with scripts which will be executed with super-user permission during the process of installing this package. Do you want to continue with the installation of` <package_name>`? [y,n,?]`	`y`	

10. After installation of all specified AnswerBooks is complete, press return when prompted.

```
Installation of <SUNWpackage_name> was successful.

   Press <Return> to continue
```

11. Choose Exit from the File menu on the Admintool: Software dialog box to quit `swmtool`.

12. If installing on a Solaris host, eject the CD.

```
# eject
```

If installing on a Solaris host, stop here. If installing on a Trusted Solaris host, go to Step 13 on page 543.

13. If installing on a Trusted Solaris host, use the Device Allocation Manager to deallocate the device.

When deallocation completes, the CD is ejected and the following prompt displays.

```
Please remove the disk from cdrom_N.

   Press RETURN to quit window....
```

▼ To Add the AnswerBook Command or Action to a Profile

1. **Assume the secadmin role.**

   See “To Login and Assume an Administrative Role” on page 15, if needed.

2. **Bring up the Profile Manager.**

   See “ To Specify Commands in the Profile Manager” on page 241 and “To Specify Actions in an Execution Profile” on page 242, if needed.

3. **To add the AnswerBook action, modify an appropriate profile in the Profile Manager Actions View.**

   a. **In the Profile Manager: Open dialog box, select and double-click on the name of an appropriate profile from the scrolling list.**

   b. **From the View menu on the Profile Manager, choose the Actions option.**

   c. **Find the Information application group in the Excluded list, and expand it by selecting it and double-clicking it.**

   d. **Find the AnswerBook action in the Information application group, and move the AnswerBook action to the Included list by selecting it and double-clicking it.**

   e. **Save the changes to the profile by choosing the Save Profiles option from the Profiles menu.**

4. **To add the** `answerbook` **command, modify an appropriate profile in the Profile Manager Command View.**

   a. **In the Profile Manager: Open dialog box, select and double-click on the name of an appropriate profile from the scrolling list.**

   b. **From the View menu on the Profile Manager, choose the Commands option.**

   c. **In the Excluded list, expand the** `/usr/openwin/bin` **directory listing in the Excluded list by selecting it and double-clicking it.**

d. **Move the** `answerbook` **command to the Included list by selecting it and double-clicking it.**

e. **Save the changes to the profile by choosing the Save Profiles option from the Profiles menu.**

Note - To ensure that an account can use the `answerbook` command, if desired, make sure that a terminal emulator such as `dtterm`, `cmdtool`, or `shelltool`, is also in one of that account's profiles.

---

5. **Assign the modified profile or profiles to individual accounts as desired, using the User Manager.**

   See the *Trusted Solaris Administration Overview* manual for more information about using the Profile Manager and for how to modify accounts using the User Manager.

▼ To Bring Up the AnswerBook Viewer

1. **On a Solaris host, enter the** `answerbook`**(1) command in a terminal emulator, such as** `dtterm`**,** `cmdtool`**, or** `shelltool`**.**

2. **On a Trusted Solaris host, if you have the command in one of your execution profiles, enter the** `answerbook`**(1) command in a terminal emulator,** such **as** `dtterm`**(1) or** `shelltool`**(1)**

3. **On a Trusted Solaris host, if you have the action in one of your execution profiles, click on the Help icon in the Front Panel and double-click on the AnswerBook icon in the Help menu.**

4. **On the AnswerBook Navigator, add the newly-installed AnswerBooks to your library.**

   a. **Click the Modify Library button.**

   b. **Click on their names to select AnswerBooks to be added.**

   c. **Click the Apply button.**

5. **On the AnswerBook Navigator, double-click the name of the AnswerBook you want to view.**

CHAPTER 17

Host Administration Checklist

This chapter provides a checklist that you can photocopy and use as a worksheet when installing new hosts on the network. The first column identifies the dialog box and the second column presents the information that needs to be supplied. The third column is provided for your convenience so that you can write down your intended input prior to running the software.

TABLE 17–1 Host Administration Checklist

Dialog Box Title	Contents	Your Answer
Host Name		
Networked?	Yes | No	
IP address		
Ethernet address		
Primary network interface	Interfaces of workstation.	
Name service	NIS+ | None	None.
Trusted Solaris configuration	Multiple user Sensitivity Labels? Hide Upgraded Names? Enable ILs? Float ILs? Reset ILs upon EXEC?	

TABLE 17–1 Host Administration Checklist *(continued)*

Dialog Box Title	Contents	Your Answer
Subnet	Yes | No	
Subnet mask	255.255.255.0	
Time zone	Offset from GMT | Geographical	
Date and Time		
System type	Standalone | OS server | Diskless	
Software group	End user | Developer | Entire	
Customize?	Yes | No	
Disk(s) to use	Disks visible to the workstation.	
Preserve?	Disks to leave as they are.	
Auto-layout file systems?	Yes | No	
File systems to auto-layout	`/, /usr, /var, /opt, swap`	
Begin installation	Yes | No	
Reboot	Yes | No	
Root password		

APPENDIX **A**

Profile Summary Tables

This appendix provides three tables that describe the default set of execution profiles. These tables do not reflect any changes that may be made at your site; you should use the Profile Manager if there is any question about the accuracy of these tables.

Execution Profile Content Summary

Table A–1 lists each execution profile and the commands, actions, and authorizations assigned to it in the default configuration. The table also indicates in parentheses () to which roles (if any) the profile is assigned by default. If you need the security attributes for specific commands, see Table A–3. If you need the security attributes for specific actions, see Table A–4. You can also use the Profile Manager to view the current contents of a specific profile.

TABLE A–1 Execution Profile Contents

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
All (Root)	Provides access to all executables (commands and actions) but without privileges.	
all	all	
All Actions	Provides access to all actions but without privileges.	
	all	
All Authorizations (Root)	Provides all authorizations. For testing.	
		all
All Commands	Provides access to all commands but without privileges.	
all		
Audit Control (Security Administrator)	For managing the audit subsystem but without ability to read files.	
audit, auditconfig, auditd, auditstat, format, mkdir, mkfs, mount, mountall, newfs, newsecfs, rm, rmdir, share, shareall, tsol_audit_badpromlogins, tunefs, umount, umountall, unshare, unshareall, writeaudit	AuditClass, AuditControl, AuditEvent, AuditStartup, AuditUser	act as file owner, set user audit flags, set/get file audit flags
Audit Review (System Administrator)	For reading the audit trail.	
auditreduce, awk, cat, grep, praudit, sed, tail		
Basic Actions (Security Administrator, System Administrator, System Operator, Root)	Provides access to the applications on the Front Panel with the necessary privileges.	

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
	BuildDataBaseRequest, Compose, Compress, Df, Diff, DtPrint, DtPrintManager, DtTTMediaOpen, DtUnlink, Dtappmgr, Dtcalc, Dtcm, Dtdevmgr, Dtfile, DtfileHome, Dthelpview, Dthelpview, Dtmail, Dtmanpageview, Dtpad, Dtprintinfo, Dtprintinfo, Dtterm, DttermConsole, Dttrash, DuSort, Env, FPHelp, FileProperties, Grep, InvokeFILEMGR, InvokeMAILER, OWanswerbook, Open, OpenCD-ROM, OpenDtIntro, OpenFloppy, OpenFolder, OpenTerminal, Print, ReOpenRestrictedFolder, ReloadActions, ReloadActionsNotice, ReloadApps, ReloadResources, Rm, SDTimage, SDtPersonalBookmarks, SDtSampleBookmarks, SDtWebClient, Terminal, TextEditor, Trash, WebBrowser, Xrefresh	
Basic Commands (Security Administrator, System Administrator, System Operator, Root)	Provides access to rudimentary commands necessary for all roles.	

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
adminvi, awk, cat, cd, chmod, clear, cmp, col, compress, cp, cut, df, diff, diff3, dircmp, dirname, du, echo, egrep, env, expr, false, fgrep, file, fold, getlabel, grep, head, hostid, hostname, id, join, ldd, ln, look, lp, lpstat, ls, mailq, man, mkdir, more, mv, niscat, nisdefaults, niserror, nisgrep, nismatch, nistest, nroff, page, pfsh, pg, ping, pr, pwd, rcp, rdist, rlogin, rm, rmdir, rsh, script, sdiff, sed, sleep, sort, spell, stty, sync, tail, tbl, test, tfind, time, touch, troff, true, tty, uname, uncompress, uniq, whereis, which, who, whoami, xhost		
Convenient Authorizations	Provides authorizations for normal users.	
		allocate device, enable logins, print a PostScript file, print without labels, remote login, shutdown the system
Cron Management (System Administrator)	For managing cron and at jobs.	
cron, crontab		modify at users, modify cron users
Cron Security (Security Administrator)	For managing cron and at jobs for administrative roles.	
crontab		modify at admin, modify cron admin
Custom Admin Role (System Administrator)	For customizing the system administrator role.	
false, true		
Custom Oper Role (System Operator)	For customizing the system operator role.	
false, true		
Custom Root Role (System Administrator)	For customizing the root role.	
trusted_edit	TrustedEditor	

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
Custom Secadmin Role (Security Administrator)	For customizing the security administrator role.	
trusted_edit	TrustedEditor	
Device Management (System Administrator)	For allocating and deallocating devices, and correcting error conditions.	
allocate, deallocate		allocate device, revoke or reclaim device
Device Security (Security Administrator)	For managing and configuring devices.	
add_allocatable, buttons_n_dials-setup, config, devlinks, drvconfig, dtlogin, eeprom, initpcmcia, keymap, leoconfig, list_devices, mkdtab, pcmcia, remove_allocatable, serialmgr, strace, tsol_dev_allocate, tsol_dev_clean, tsol_dev_policy, volmgt	AddAllocDev, Serialmgr	configure device attributes
Enable Login (System Administrator)	Provides the authorization for allowing yourself and other users to log in after boot.	
		enable logins
File System Management (System Administrator)	For managing file systems.	
autofs, automount, automountd, buildmnttab, client, clri, devinfo, dfmounts, dfshares, eject, format, fsck, fsdb, fsirand, fstyp, fusage, fuser, getfattrflag, getfsattr, mkdir, mkfile, mkfs, mount, mountall, ncheck, newfs, nfsstat, rmdir, server, setfattrflag, share, shareall, showmount, standardmounts, swap, tunefs, ufs_quota, umount, umountall, unshare, unshareall	ShareFS, Vfstab	
File System Security (Security Administrator)	For managing file system labels and other security attributes.	
getfsattr, newsecfs, setfsattr	Vfstab_adjunct	
Mail Management (System Administrator)	For configuring sendmail, modifying aliases, and checking mail queues.	

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
dbmgr, mailq, mconnect, newaliases, sendmail, sendmail	Dbmgr, SendMail	modify aliases
Maintenance and Repair (System Administrator)	Provides commands needed to maintain or repair a system.	
adb, crash, date, dmesg, eeprom, halt, init, ldd, poweroff, prtconf, reboot, sysetup, syslog, syslogd, tsol_sync_time, vmstat		enable logins, remote login, shutdown the system, terminal login
Media Backup (System Operator)	Backup files.	
mt, tar, ufsdump	OWtapetool, Tar, TarList, OWtapetool, TarUnpack	allocate device
Media Restore (System Administrator)	Restore files from backup.	
cpio, mt, tar, ufsrestore		allocate device
Network Management (System Administrator)	For managing the host and network configuration.	
asppp, dbmgr, hostmgr, ifconfig, inetinit, inetsvc, named, net, netstat, nscd, ping, route, rpc, rup, ruptime, setuname, snoop, spray, sys, tokmapd, tsol_tcb_verify, uucp	DNS_Resolve, Dbmgr, EditMotd, Hostmgr, Nsswitch, SetRoutes, Tnchkdb, Tnchkdb_nisplus, Tsolgateways	enable logins, modify bootparams, modify ethers, modify hosts, modify locale, modify netgroup, modify netmasks, modify networks, modify protocols, modify rpc, modify services, modify timezone
Network Security (Security Administrator)	For managing network and host security, with authorizations for modifying trusted network databases.	
dbmgr, rootusr, tnchkdb, tnctl, tnd, tninfo	CheckEncodings, Dbmgr, Niscat, Tnchkdb, Tnchkdb_nisplus	enable logins, modify tnidb, modify tnrhdb, modify tnrhtp
NIS+ Administration (System Administrator)	Provides access to NIS+ scripts/commands that are not security-related.	
nischttl, nisctl, nisln, nisping, nisshowcache, nisstat, nistnsetup, nistntime, nscd	Niscat, Niscat_o	

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
NIS+ Security Administration (Security Administrator)	Provides access to NIS+ security-related scripts/commands.	
chkey, newkey, nisaddcred, nisaddent, nischgrp, nischmod, nischown, nisclient, nisd, nisgrpadm, nisinit, nislog, nismkdir, nispasswd, nispopulate, nisrm, nisrmdir, nisserver, nissetup, nistbladm, nisupdkeys	Nisclient, Nispopulate, Nisserver	
Object Access Management (Security Administrator)	For changing ownership and permissions on files.	
chgrp, chmod, chown, getfacl, getfattrflag, getlabel, mldpwd, mldrealpath, setfacl, setfattrflag	Dtfile, DtfileHome, Dttrash, InvokeFILEMGR	act as file owner, change file owner
Object Label Management (Security Administrator)	For changing labels of files and setting up system-wide labels.	
atohexlabel, chk_encodings, getlabel, getmldadorn, getsldname, hextoalabel, mldpwd, mldrealpath, setfattrflag, setlabel, tokmapctl, tsol_label_services	CheckEncodings, Dtfile, DtfileHome, Dttrash, EditEncodings, Selconfig	allocate device, bypass file view, downgrade file sensitivity label, paste to a downgraded window, paste to an upgraded window, upgrade file sensitivity label, use all defined labels
Object Privilege Management (Security Administrator)	For changing privileges on executable files.	
getfpriv, ppriv, pprivtest, runpd, setfpriv, testfpriv, tsol_priv_enable	DtfileHome, InvokeFILEMGR	set file privileges
Outside Accred (Security Administrator, System Administrator, System Operator)	Operate outside system accreditation range.	
		use all defined labels
Printer Security (Security Administrator)	For managing printer devices.	
accept, cancel, disable, enable, lp, lpadmin, lpfilter, lpforms, lpmove, lpq, lprm, lpsched, lpshut, lpstat, lpsystem, lpusers, printmgr, reject	Printermgr	administer printing, configure device attributes, print a PostScript file, print without banners, print without labels

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
Privileged Shells (Root)	For developers to run Bourne, Korn, and C shells with all privileges. *NOT intended for secure environments.*	
csh, ksh, sh		
Process Management (Security Administrator, System Administrator)	For managing current processes, including cron and at jobs.	
cron, crontab, fuser, kill, nice, pattr, pclear, pcred, perf, pfiles, pflags, plabel, pldd, pmap, ppriv, pprivtest, prun, ps, psig, pstack, pstop, ptime, ptree, pwait, pwdx, renice, truss		modify at users, modify cron users
Software Installation (System Administrator)	For adding application software to the system.	
add_drv, install, ln, make, pkgadd, pkgask, pkgchk, pkginfo, pkgmk, pkgmv, pkgparam, pkgproto, pkgrm, pkgtrans, rem_drv, swmtool		allocate device
System Management	*Obsolete execution profile – included for backwards compatibility.*	
accept, adminvi, allocate, automount, automountd, cancel, client, date, dbmgr, deallocate, disable, enable, format, getfsattr, hostmgr, init, kill, list_devices, lp, lpadmin, lpfilter, lpmove, lpshut, lpstat, lpsystem, lpusers, mailq, mkfile, mkfs, mount, mountall, named, newaliases, newfs, nice, pattr, pclear, pcred, pfiles, pflags, ping, plabel, pldd, pmap, prun, ps, psig, pstack, pstop, ptime, ptree, pwait, pwdx, rdist, reject, renice, rup, share, shareall, showmount, swap, umount, umountall, unshare, unshareall, vmstat, xhost	DNS_Resolve, Dbmgr, EditMotd, Hostmgr, ShareFS, Vfstab	administer printing, modify at users, modify bootparams, modify cron users, modify ethers, modify hosts, modify locale, modify netmasks, modify networks, modify protocols, modify rpc, modify services, modify timezone
System Security	*Obsolete execution profile – included for backwards compatibility.*	

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
accept, add_drv, adminvi, allocate, autopush, dbmgr, deallocate, disable, drvconfig, enable, format, getfsattr, mkfs, newfs, newsecfs, ping, printmgr, ps, reject, rem_drv, serialmgr, setfsattr, tnchkdb, tnctl, tnd, tninfo	Dbmgr, Nsswitch, Printermgr, SendMail, Serialmgr, Tnchkdb, Tnchkdb_nisplus, TrustedEditor, Vfstab_adjunct	modify at admin, modify cron admin, modify netgroup, modify tnidb, modify tnrhdb, modify tnrhtp, print a PostScript file, print without labels, remote login
User Management (System Administrator)	For creating and modifying users but without the ability to modify self (as a security measure).	
groupmgr, grpck, pwck, usermgr, utmpd	Groupmgr, Usermgr	modify aliases, modify group, set home directory attributes, set user identity
User Security (Security Administrator)	For creating and modifying users' security attributes but without the ability to modify self (as a security measure).	
dbmgr, passwd, profmgr, pwck, pwconv, usermgr	Dbmgr, Profmgr, Usermgr	modify auto_home, set idle time, set roles list, set user audit flags, set user labels, set user password, set user profiles, set/get file audit flags, use all defined labels
boot	For starting and shutting down the system.	
auditd, automount, automountd, cron, deallocate, inetd, lockd, lpsched, mount, mountall, mountall, mountd, named, nscd, rpcbind, savecore, sendmail, share, shareall, statd, syslogd, tnd, umount, umountall, unshare, unshareall		act as file owner, allocate device, bypass file view, change file owner, downgrade file sensitivity label, enable logins, paste to a downgraded window, paste to an upgraded window, permit self-modification, print a PostScript file, print without labels, remote login, revoke or reclaim device, set file privileges, set home directory attributes, set idle time, set roles list, set user audit flags, set user identity, set user labels, set user password, set user profiles, set/get file audit flags, terminal login, upgrade file sensitivity label, use all defined labels
cron (Root)	Provides root with commands needed for cron jobs.	

TABLE A–1 Execution Profile Contents *(continued)*

Execution Profiles (Default Role Assignments)	**Purpose**	
Commands	**Actions**	**Authorizations**
cp, mv, rdate, rm, setaudit, tfind, tsol_audit_badpromlogins		
dtwm	For using the window manager.	
	Dtdevmgr, Dtstyle, ExitSession, LockDisplay, SDTaccessx, StartDtscreenBlank, StartDtscreenFlame, StartDtscreenHop, StartDtscreenImage, StartDtscreenLife, StartDtscreenPyro, StartDtscreenQix, StartDtscreenRotor, StartDtscreenSwarm, StartDtscreenWorm	
inetd	For programs executed by the inetd daemon.	
cmsd, ftpd, getpeerinfo, rexecd, rlogind, rshd, sadmind, telnetd, tftpd, ttdbserverd		
required	Required for all profile users.	
tsolxagent, ttsession		

Execution Profile Assignment to Roles

Table A–2 shows the default execution profiles and indicates the administrative roles, if any, to which they are assigned.

TABLE A–2 Execution Profiles with Assignment to Roles

Profile Name	Security Admin	System Admin	System Oper	Root
All				Y
All Actions				
All Authorizations				
All Commands				
Audit Control	Y			
Audit Review		Y		
Basic Actions	Y	Y	Y	Y
Basic Commands	Y	Y	Y	Y
boot				
Convenient Authorizations				
cron				Y
Cron Management		Y		
Cron Security	Y			
Custom Admin Role		Y		
Custom Oper Role			Y	
Custom Root Role				Y
Custom Secadmin Role	Y			
Device Management		Y		Y
Device Security	Y			
dtwm				
Enable Login		Y		
File System Management		Y		Y

TABLE A–2 Execution Profiles with Assignment to Roles *(continued)*

Profile Name	Security Admin	System Admin	System Oper	Root
File System Security	Y			
inetd				
Mail Management		Y		
Maintenance and Repair		Y		
Media Backup			Y	
Media Restore		Y		
Network Management		Y		Y
Network Security	Y			Y
NIS+ Administration		Y		
NIS+ Security Administration	Y			Y
Object Access Management	Y			
Object Label Management	Y			
Object Privilege Management	Y			
Outside Accred	Y	Y	Y	Y
Printer Security	Y			Y
Privileged Shells				
Process Management	Y	Y		
Software Installation		Y		Y
User Management		Y		Y
User Security	Y			Y

Finding Commands in Execution Profiles

Table A–3 lists each command contained in any execution profile and the execution profile(s) to which it is assigned. Remember that a command may be contained in more than one execution profile. The table also indicates the full path of the command as well as any security attributes: minimum sensitivity label, maximum sensitivity label, setUID value, setGID value, and privileges.

TABLE A–3 Commands and their Associated Execution Profiles

Command	Profile	Path	Security Attributes
accept	Printer Security	/usr/sbin/accept	sys_devices
accept	System Management	/usr/sbin/accept	sys_devices
accept	System Security	/usr/sbin/accept	sys_devices
adb	Maintenance and Repair	/usr/bin/adb	
add_allocatable	Device Security	/usr/sbin/add_allocatable	file_chown, file_dac_write, file_downgrade_sl, file_mac_read, file_mac_write, file_setdac, sys_trans_label
add_drv	Software Installation	/usr/sbin/add_drv	euid = 0, egid = 3, min SL = ADMIN_LOW, max SL = ADMIN_LOW, file_dac_read, file_dac_write, file_mac_read, file_mac_write, sys_devices
add_drv	System Security	/usr/sbin/add_drv	
adminvi	Basic Commands	/usr/bin/adminvi	
adminvi	System Management	/usr/bin/adminvi	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
adminvi	System Security	/usr/bin/adminvi	
allocate	Device Management	/usr/sbin/allocate	file_chown, file_dac_read, file_dac_search, file_dac_write, file_downgrade_sl, file_mac_read, file_mac_search, file_owner, file_setdac, sys_audit, sys_devices, sys_mount
allocate	System Management	/usr/sbin/allocate	file_chown, file_setdac
allocate	System Security	/usr/sbin/allocate	file_chown, file_setdac
asppp	Network Management	/etc/init.d/asppp	euid = 0, egid = 3
atohexlabel	Object Label Management	/usr/sbin/atohexlabel	
audit	Audit Control	/etc/init.d/audit	euid = 0, egid = 3
audit	Audit Control	/usr/sbin/audit	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH, file_mac_read, proc_audit_tcb, proc_mac_write, sys_audit
auditconfig	Audit Control	/usr/sbin/auditconfig	euid = 0, min SL = ADMIN_LOW, max SL = ADMIN_LOW, sys_audit
auditd	Audit Control	/usr/sbin/auditd	euid = 0, file_mac_write, proc_setclr, proc_setil, proc_setsl, sys_audit
auditd	boot	/usr/sbin/auditd	file_mac_write, proc_setclr, proc_setil, proc_setsl, sys_audit
auditreduce	Audit Review	/usr/sbin/auditreduce	euid = 0, min SL = ADMIN_HIGH, file_dac_read, sys_audit
auditstat	Audit Control	/usr/sbin/auditstat	euid = 0, sys_audit

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
autofs	File System Management	/etc/init.d/autofs	euid = 0, egid = 3
automount	File System Management	/usr/lib/fs/autofs/automount	file_dac_read, file_dac_write, file_mac_read, file_mac_write, proc_nofloat, sys_mount
automount	System Management	/usr/lib/fs/autofs/automount	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_dac_read, file_dac_write, file_mac_read, file_mac_write, proc_nofloat, sys_mount
automount	boot	/usr/lib/fs/autofs/automount	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_dac_read, file_dac_write, file_mac_read, file_mac_write, proc_nofloat, sys_mount
automountd	File System Management	/usr/lib/autofs/automountd	file_dac_execute, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, file_owner, file_upgrade_il, file_upgrade_sl, net_mac_read, net_privaddr, net_upgrade_il, net_upgrade_sl, proc_audit_tcb, proc_nofloat, proc_setil, proc_setsl, sys_mount, sys_trans_label
automountd	System Management	/usr/lib/autofs/automountd	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_mac_read, file_mac_write, file_upgrade_il, file_upgrade_sl, net_mac_read, net_privaddr, net_upgrade_il, net_upgrade_sl, proc_audit_tcb, proc_nofloat, proc_setil, proc_setsl, sys_mount, sys_trans_label

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
automountd	boot	/usr/lib/autofs/automountd	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_dac_execute, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, file_owner, file_upgrade_il, file_upgrade_sl, net_mac_read, net_privaddr, net_upgrade_il, net_upgrade_sl, proc_audit_tcb, proc_nofloat, proc_setil, proc_setsl, sys_mount, sys_trans_label
autopush	System Security	/usr/sbin/autopush	
awk	Audit Review	/usr/bin/awk	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH
awk	Basic Commands	/usr/bin/awk	
buildmnttab	File System Management	/etc/init.d/buildmnttab	euid = 0, egid = 3
buttons_n_dials-setup	Device Security	/etc/init.d/ buttons_n_dials-setup	euid = 0, egid = 3
cancel	Printer Security	/usr/bin/cancel	euid = 71, file_dac_write, file_mac_read, file_mac_write
cancel	System Management	/usr/bin/cancel	euid = 71, file_mac_read, file_mac_write
cat	Audit Review	/usr/bin/cat	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH
cat	Basic Commands	/usr/bin/cat	
cd	Basic Commands	/usr/bin/cd	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
chgrp	Object Access Management	/usr/bin/chgrp	file_chown, file_dac_read, file_dac_search, file_mac_read, file_mac_search, file_owner, file_setdac, file_setid
chk_encodings	Object Label Management	/usr/sbin/chk_encodings	
chkey	NIS+ Security Administration	/usr/bin/chkey	
chmod	Basic Commands	/usr/bin/chmod	
chmod	Object Access Management	/usr/bin/chmod	file_dac_read, file_dac_search, file_mac_read, file_mac_search, file_setdac, file_setid
chown	Object Access Management	/usr/bin/chown	file_chown, file_dac_read, file_dac_search, file_mac_read, file_mac_search, file_owner
clear	Basic Commands	/usr/bin/clear	
client	File System Management	/etc/init.d/nfs.client	euid = 0, egid = 7, min SL = ADMIN_LOW, max SL = ADMIN_LOW
client	System Management	/etc/init.d/nfs.client	euid = 0, egid = 3, all privileges
clri	File System Management	/usr/sbin/clri	
cmp	Basic Commands	/usr/bin/cmp	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
cmsd	inetd	/usr/dt/bin/rpc.cmsd	file_chown, file_dac_read, file_dac_write, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_write, file_nofloat, file_owner, file_setdac, file_setid, net_broadcast, net_downgrade_il, net_downgrade_sl, net_nofloat, net_privaddr, proc_mac_read, proc_mac_write, proc_owner, proc_setid
col	Basic Commands	/usr/bin/col	
compress	Basic Commands	/usr/bin/compress	
config	Device Security	/etc/init.d/rtvc-config	euid = 0, egid = 3
cp	Basic Commands	/usr/bin/cp	
cp	cron	/usr/bin/cp	file_mac_write
cpio	Media Restore	/usr/bin/cpio	
crash	Maintenance and Repair	/usr/sbin/crash	
cron	Cron Management	/etc/init.d/cron	euid = 0, egid = 3
cron	Process Management	/etc/init.d/cron	euid = 0, egid = 3
cron	boot	/usr/sbin/cron	euid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_dac_read, file_mac_write, file_owner, net_mac_read, proc_audit_tcb, proc_nofloat, proc_setclr, proc_setid, proc_setil, proc_setsl, sys_audit

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
crontab	Cron Management	/usr/bin/crontab	
crontab	Cron Security	/usr/bin/crontab	
crontab	Process Management	/usr/bin/crontab	
csh	Privileged Shells	/usr/bin/csh	all privileges
cut	Basic Commands	/usr/bin/cut	
date	Maintenance and Repair	/usr/bin/date	sys_config
date	System Management	/usr/bin/date	sys_config
dbmgr	Mail Management	/opt/SUNWadm/2.3/bin/dbmgr	file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
dbmgr	Network Management	/opt/SUNWadm/2.3/bin/dbmgr	min SL = ADMIN_LOW, file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
dbmgr	Network Security	/opt/SUNWadm/2.3/bin/dbmgr	min SL = ADMIN_LOW, max SL = ADMIN_LOW, file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
dbmgr	System Management	/opt/SUNWadm/2.3/bin/dbmgr	file_chown, file_dac_write, sys_trans_label
dbmgr	System Security	/opt/SUNWadm/2.3/bin/dbmgr	all privileges
dbmgr	User Security	/opt/SUNWadm/2.3/bin/dbmgr	file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
deallocate	Device Management	/usr/sbin/deallocate	file_chown, file_dac_read, file_dac_write, file_mac_read, file_setdac, sys_audit, sys_devices, sys_mount, sys_net_config

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
deallocate	System Management	/usr/sbin/deallocate	file_chown, file_setdac
deallocate	System Security	/usr/sbin/deallocate	file_chown, file_setdac
deallocate	boot	/usr/sbin/deallocate	file_chown, file_dac_read, file_dac_write, file_mac_read, file_setdac, sys_audit, sys_devices, sys_mount, sys_net_config
devinfo	File System Management	/usr/sbin/devinfo	
devlinks	Device Security	/etc/init.d/devlinks	euid = 0, egid = 3
df	Basic Commands	/usr/bin/df	
dfmounts	File System Management	/usr/sbin/dfmounts	
dfshares	File System Management	/usr/sbin/dfshares	
diff	Basic Commands	/usr/bin/diff	
diff3	Basic Commands	/usr/bin/diff3	
dircmp	Basic Commands	/usr/bin/dircmp	
dirname	Basic Commands	/usr/bin/dirname	
disable	Printer Security	/usr/bin/disable	sys_devices
disable	System Management	/usr/bin/disable	sys_devices

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
disable	System Security	/usr/bin/disable	sys_devices
dmesg	Maintenance and Repair	/usr/sbin/dmesg	min SL = ADMIN_HIGH, max SL = ADMIN_HIGH
drvconfig	Device Security	/etc/init.d/drvconfig	euid = 0, egid = 3
drvconfig	System Security	/usr/sbin/drvconfig	
dtlogin	Device Security	/etc/init.d/dtlogin	euid = 0, egid = 3
du	Basic Commands	/usr/bin/du	
echo	Basic Commands	/usr/bin/echo	
eeprom	Device Security	/usr/sbin/eeprom	euid = 0
eeprom	Maintenance and Repair	/usr/sbin/eeprom	
egrep	Basic Commands	/usr/bin/egrep	
eject	File System Management	/usr/bin/eject	file_dac_read
enable	Printer Security	/usr/bin/enable	sys_devices
enable	System Management	/usr/bin/enable	sys_devices
enable	System Security	/usr/bin/enable	sys_devices
env	Basic Commands	/usr/bin/env	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
expr	Basic Commands	/usr/bin/expr	
false	Basic Commands	/usr/bin/false	
false	Custom Admin Role	/usr/bin/false	
false	Custom Oper Role	/usr/bin/false	
fgrep	Basic Commands	/usr/bin/fgrep	
file	Basic Commands	/usr/bin/file	
fold	Basic Commands	/usr/bin/fold	
format	Audit Control	/usr/sbin/format	euid = 0, sys_devices
format	File System Management	/usr/sbin/format	euid = 0, sys_devices
format	System Management	/usr/sbin/format	euid = 0
format	System Security	/usr/sbin/format	euid = 0, all privileges
fsck	File System Management	/usr/sbin/fsck	
fsdb	File System Management	/usr/sbin/fsdb	
fsirand	File System Management	/usr/sbin/fsirand	
fstyp	File System Management	/usr/sbin/fstyp	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
ftpd	inetd	/usr/sbin/in.ftpd	file_mac_write, net_privaddr, proc_audit_appl, proc_audit_tcb, proc_chroot, proc_setid
fusage	File System Management	/usr/sbin/fusage	
fuser	File System Management	/usr/sbin/fuser	file_dac_search, file_mac_search, proc_audit_tcb, proc_owner, sys_mount
fuser	Process Management	/usr/sbin/fuser	file_dac_search, file_mac_search, proc_audit_tcb, proc_owner
getfacl	Object Access Management	/usr/bin/getfacl	file_dac_search, file_mac_read, file_mac_search
getfattrflag	File System Management	/usr/bin/getfattrflag	file_audit, file_dac_search, file_mac_read, file_mac_search
getfattrflag	Object Access Management	/usr/bin/getfattrflag	file_audit, file_dac_search, file_mac_read, file_mac_search
getfpriv	Object Privilege Management	/usr/bin/getfpriv	
getfsattr	File System Management	/usr/sbin/getfsattr	file_dac_read, file_dac_search, file_mac_search, sys_trans_label
getfsattr	File System Security	/usr/sbin/getfsattr	file_dac_read, file_dac_search, file_mac_search, sys_trans_label
getfsattr	System Management	/usr/sbin/getfsattr	file_dac_read, file_dac_search, file_mac_search
getfsattr	System Security	/usr/sbin/getfsattr	egid = 3
getlabel	Basic Commands	/usr/bin/getlabel	
getlabel	Object Access Management	/usr/bin/getlabel	file_dac_search, file_mac_read, file_mac_search

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
getlabel	Object Label Management	/usr/bin/getlabel	file_dac_read, file_dac_search, file_mac_read, file_mac_search, sys_trans_label
getmldadorn	Object Label Management	/usr/bin/getmldadorn	
getpeerinfo	inetd	/usr/sbin/rpc.getpeerinfo	net_downgrade_sl, net_upgrade_sl, proc_audit_appl, proc_audit_tcb
getsldname	Object Label Management	/usr/bin/getsldname	
grep	Audit Review	/usr/bin/grep	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH
grep	Basic Commands	/usr/bin/grep	
groupmgr	User Management	/opt/SUNWadm/2.3/bin/groupmgr	file_dac_write, proc_audit_tcb
grpck	User Management	/usr/sbin/grpck	
halt	Maintenance and Repair	/usr/sbin/halt	euid = 0, sys_boot
head	Basic Commands	/usr/bin/head	
hextoalabel	Object Label Management	/usr/sbin/hextoalabel	sys_trans_label
hostid	Basic Commands	/usr/bin/hostid	
hostmgr	Network Management	/opt/SUNWadm/2.3/bin/hostmgr	min SL = ADMIN_LOW, file_dac_write, proc_audit_tcb
hostmgr	System Management	/opt/SUNWadm/2.3/bin/hostmgr	file_chown, file_dac_write, sys_trans_label

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
hostname	Basic Commands	/usr/bin/hostname	
id	Basic Commands	/usr/bin/id	
ifconfig	Network Management	/usr/sbin/ifconfig	euid = 0, file_mac_read, sys_net_config
inetd	boot	/usr/sbin/inetd	min SL = ADMIN_LOW, max SL = ADMIN_HIGH, all privileges
inetinit	Network Management	/etc/init.d/inetinit	euid = 0, egid = 3
inetsvc	Network Management	/etc/init.d/inetsvc	euid = 0, egid = 3
init	Maintenance and Repair	/usr/sbin/init	file_chown, file_dac_write, file_mac_write, proc_audit_tcb
init	System Management	/usr/sbin/init	all privileges
initpcmcia	Device Security	/etc/init.d/initpcmcia	euid = 0, egid = 3
install	Software Installation	/usr/sbin/install	file_chown, file_dac_read, file_dac_search, file_dac_write, file_setid
join	Basic Commands	/usr/bin/join	
keymap	Device Security	/etc/init.d/keymap	euid = 0, egid = 3
kill	Process Management	/usr/bin/kill	proc_mac_write, proc_owner
kill	System Management	/usr/bin/kill	proc_mac_write, proc_owner
ksh	Privileged Shells	/usr/bin/ksh	all privileges

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
ldd	Basic Commands	/usr/bin/ldd	
ldd	Maintenance and Repair	/usr/bin/ldd	
leoconfig	Device Security	/etc/init.d/leoconfig	euid = 0, egid = 3
list_devices	Device Security	/usr/sbin/list_devices	
list_devices	System Management	/usr/sbin/list_devices	
ln	Basic Commands	/usr/bin/ln	
ln	Software Installation	/usr/bin/ln	file_dac_write
lockd	boot	/usr/lib/nfs/lockd	euid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, net_mac_read, net_privaddr, net_upgrade_sl, proc_dumpcore, proc_nofloat, sys_net_config, sys_nfs, sys_suser_compat, sys_trans_label
look	Basic Commands	/usr/bin/look	
lp	Basic Commands	/usr/bin/lp	
lp	Printer Security	/etc/init.d/lp	euid = 0, egid = 3
lp	System Management	/usr/bin/lp	
lpadmin	Printer Security	/usr/sbin/lpadmin	euid = 0, file_dac_write, file_owner
lpadmin	System Management	/usr/sbin/lpadmin	euid = 0, max SL = ADMIN_LOW

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
lpfilter	Printer Security	/usr/sbin/lpfilter	euid = 0, file_dac_write
lpfilter	System Management	/usr/sbin/lpfilter	euid = 0, max SL = ADMIN_LOW
lpforms	Printer Security	/usr/sbin/lpforms	euid = 0
lpmove	Printer Security	/usr/sbin/lpmove	euid = 0
lpmove	System Management	/usr/sbin/lpmove	euid = 0, max SL = ADMIN_LOW
lpq	Printer Security	/usr/ucb/lpq	file_dac_read, file_mac_read, sys_trans_label
lprm	Printer Security	/usr/ucb/lprm	file_dac_write, file_mac_read, file_mac_write
lpsched	Printer Security	/usr/lib/lp/lpsched	euid = 0, file_chown, file_dac_read, file_dac_search, file_dac_write, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_owner, file_setdac, file_setid, file_upgrade_sl, net_downgrade_sl, net_mac_read, net_setid, net_setpriv, net_upgrade_il, proc_audit_tcb, proc_mac_write, proc_nofloat, proc_owner, proc_setclr, proc_setid, proc_setsl, proc_tranquil, sys_trans_label

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
lpsched	boot	/usr/lib/lp/lpsched	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH, file_chown, file_dac_read, file_dac_search, file_dac_write, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_owner, file_setdac, file_setid, file_upgrade_sl, net_downgrade_sl, net_mac_read, net_setid, net_setpriv, net_upgrade_il, proc_audit_tcb, proc_mac_write, proc_nofloat, proc_owner, proc_setclr, proc_setid, proc_setsl, proc_tranquil, sys_trans_label
lpshut	Printer Security	/usr/sbin/lpshut	euid = 0
lpshut	System Management	/usr/sbin/lpshut	euid = 0
lpstat	Basic Commands	/usr/bin/lpstat	
lpstat	Printer Security	/usr/bin/lpstat	file_dac_read, file_mac_read, sys_trans_label
lpstat	System Management	/usr/bin/lpstat	file_dac_read, file_mac_read
lpsystem	Printer Security	/usr/sbin/lpsystem	euid = 0
lpsystem	System Management	/usr/sbin/lpsystem	euid = 0, max SL = ADMIN_LOW
lpusers	Printer Security	/usr/sbin/lpusers	euid = 0
lpusers	System Management	/usr/sbin/lpusers	euid = 0, max SL = ADMIN_LOW

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
ls	Basic Commands	/usr/bin/ls	
mailq	Basic Commands	/usr/bin/mailq	egid = 2
mailq	Mail Management	/usr/bin/mailq	
mailq	System Management	/usr/bin/mailq	egid = 2, file_dac_read, file_mac_read
make	Software Installation	/usr/ccs/bin/make	
man	Basic Commands	/usr/bin/man	
mconnect	Mail Management	/usr/bin/mconnect	
mkdir	Audit Control	/usr/bin/mkdir	file_dac_write
mkdir	Basic Commands	/usr/bin/mkdir	
mkdir	File System Management	/usr/bin/mkdir	file_dac_write
mkdtab	Device Security	/etc/init.d/mkdtab	euid = 0, egid = 3
mkfile	File System Management	/usr/sbin/mkfile	
mkfile	System Management	/usr/sbin/mkfile	
mkfs	Audit Control	/usr/sbin/mkfs	all privileges
mkfs	File System Management	/usr/sbin/mkfs	file_dac_read, file_dac_write
mkfs	System Management	/usr/sbin/mkfs	euid = 0, all privileges

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
mkfs	System Security	/usr/sbin/mkfs	euid = 0, all privileges
mldpwd	Object Access Management	/usr/bin/mldpwd	file_dac_write
mldpwd	Object Label Management	/usr/bin/mldpwd	file_dac_write
mldrealpath	Object Access Management	/usr/bin/mldrealpath	file_dac_write
mldrealpath	Object Label Management	/usr/bin/mldrealpath	file_dac_write
more	Basic Commands	/usr/bin/more	
mount	Audit Control	/usr/sbin/mount	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
mount	File System Management	/usr/sbin/mount	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
mount	System Management	/usr/sbin/mount	euid = 0, file_dac_read, file_dac_search, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
mount	boot	/usr/sbin/mount	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
mountall	Audit Control	/usr/sbin/mountall	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
mountall	File System Management	/usr/sbin/mountall	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
mountall	System Management	/usr/sbin/mountall	euid = 0, file_dac_read, file_dac_search, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
mountall	boot	/sbin/mountall	max SL = ADMIN_HIGH, all privileges
mountall	boot	/usr/sbin/mountall	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
mountd	boot	/usr/lib/nfs/mountd	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_dac_search, file_mac_read, file_mac_search, net_mac_read, net_privaddr, proc_nofloat, proc_setclr, proc_setil, proc_setsl, sys_audit, sys_devices, sys_net_config, sys_nfs
mt	Media Backup	/usr/bin/mt	
mt	Media Restore	/usr/bin/mt	
mv	Basic Commands	/usr/bin/mv	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
mv	cron	/usr/bin/mv	file_mac_write
named	Network Management	/usr/sbin/in.named	euid = 0, file_mac_read, net_mac_read, net_privaddr, net_upgrade_sl, proc_dumpcore, proc_nofloat, proc_setclr, sys_config, sys_net_config, sys_trans_label
named	System Management	/usr/sbin/in.named	euid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_mac_read, net_mac_read, net_privaddr, net_upgrade_sl, proc_dumpcore, proc_nofloat, proc_setclr, sys_config, sys_net_config, sys_trans_label
named	boot	/usr/sbin/in.named	euid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_mac_read, net_mac_read, net_privaddr, net_upgrade_sl, proc_dumpcore, proc_nofloat, proc_setclr, sys_config, sys_net_config, sys_trans_label
ncheck	File System Management	/usr/sbin/ncheck	
net	Network Management	/etc/init.d/sysid.net	euid = 0, egid = 3
netstat	Network Management	/usr/bin/netstat	
newaliases	Mail Management	/usr/bin/newaliases	net_mac_read, net_privaddr, proc_nofloat, proc_setil
newaliases	System Management	/usr/bin/newaliases	euid = 0, net_mac_read, net_privaddr, proc_nofloat, proc_setil
newfs	Audit Control	/usr/sbin/newfs	euid = 0, all privileges
newfs	File System Management	/usr/sbin/newfs	file_dac_read, file_dac_write

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
newfs	System Management	/usr/sbin/newfs	euid = 0, all privileges
newfs	System Security	/usr/sbin/newfs	euid = 0, all privileges
newkey	NIS+ Security Administration	/usr/sbin/newkey	
newsecfs	Audit Control	/usr/sbin/newsecfs	all privileges
newsecfs	File System Security	/usr/sbin/newsecfs	file_dac_read, file_dac_write
newsecfs	System Security	/usr/sbin/newsecfs	euid = 0, all privileges
nfsstat	File System Management	/usr/bin/nfsstat	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH, file_mac_write, sys_config
nice	Process Management	/usr/bin/nice	
nice	System Management	/usr/bin/nice	
nisaddcred	NIS+ Security Administration	/usr/bin/nisaddcred	
nisaddent	NIS+ Security Administration	/usr/lib/nis/nisaddent	
niscat	Basic Commands	/usr/bin/niscat	
nischgrp	NIS+ Security Administration	/usr/bin/nischgrp	
nischmod	NIS+ Security Administration	/usr/bin/nischmod	
nischown	NIS+ Security Administration	/usr/bin/nischown	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
nischttl	NIS+ Administration	/usr/bin/nischttl	
nisclient	NIS+ Security Administration	/usr/lib/nis/nisclient	file_dac_read, file_dac_write, file_mac_read, net_mac_read, net_reply_equal, net_setclr, net_setid, net_setpriv, proc_nofloat, proc_owner, sys_net_config
nisctl	NIS+ Administration	/usr/lib/nis/nisctl	
nisd	NIS+ Security Administration	/usr/sbin/rpc.nisd	euid = 0, egid = 0, net_mac_read, net_upgrade_sl, proc_setclr, proc_setsl
nisdefaults	Basic Commands	/usr/bin/nisdefaults	
niserror	Basic Commands	/usr/bin/niserror	
nisgrep	Basic Commands	/usr/bin/nisgrep	
nisgrpadm	NIS+ Security Administration	/usr/bin/nisgrpadm	
nisinit	NIS+ Security Administration	/usr/sbin/nisinit	
nisln	NIS+ Administration	/usr/bin/nisln	
nislog	NIS+ Security Administration	/usr/sbin/nislog	
nismatch	Basic Commands	/usr/bin/nismatch	
nismkdir	NIS+ Security Administration	/usr/bin/nismkdir	
nispasswd	NIS+ Security Administration	/usr/bin/nispasswd	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
nisping	NIS+ Administration	/usr/lib/nis/nisping	
nispopulate	NIS+ Security Administration	/usr/lib/nis/nispopulate	
nisrm	NIS+ Security Administration	/usr/bin/nisrm	
nisrmdir	NIS+ Security Administration	/usr/bin/nisrmdir	
nisserver	NIS+ Security Administration	/usr/lib/nis/nisserver	file_dac_read, file_mac_read, net_mac_read, net_reply_equal, net_setclr, net_setid, net_setpriv, net_upgrade_sl, proc_nofloat, proc_setclr, proc_setsl, sys_net_config
nissetup	NIS+ Security Administration	/usr/lib/nis/nissetup	
nisshowcache	NIS+ Administration	/usr/lib/nis/nisshowcache	
nisstat	NIS+ Administration	/usr/lib/nis/nisstat	
nistbladm	NIS+ Security Administration	/usr/bin/nistbladm	
nistest	Basic Commands	/usr/bin/nistest	
nistnsetup	NIS+ Administration	/usr/lib/nis/nistnsetup	
nistntime	NIS+ Administration	/usr/lib/nis/nistntime	
nisupdkeys	NIS+ Security Administration	/usr/lib/nis/nisupdkeys	
nroff	Basic Commands	/usr/bin/nroff	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
nscd	NIS+ Administration	/usr/sbin/nscd	file_dac_write, file_setid, net_mac_read, net_upgrade_sl, proc_dumpcore, proc_nofloat, proc_setclr, sys_net_config, sys_trans_label
nscd	Network Management	/etc/init.d/nscd	euid = 0, egid = 3
nscd	boot	/usr/sbin/nscd	min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_dac_write, file_setid, net_mac_read, net_upgrade_sl, proc_dumpcore, proc_nofloat, proc_setclr, sys_net_config, sys_trans_label
page	Basic Commands	/usr/bin/page	
passwd	User Security	/usr/bin/passwd	
pattr	Process Management	/usr/proc/bin/pattr	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pattr	System Management	/usr/proc/bin/pattr	file_dac_read, proc_mac_read, proc_owner
pclear	Process Management	/usr/proc/bin/pclear	file_dac_read, proc_mac_read, proc_nofloat, proc_owner, sys_trans_label
pclear	System Management	/usr/proc/bin/pclear	file_dac_read, proc_mac_read, proc_owner, sys_trans_label
pcmcia	Device Security	/etc/init.d/pcmcia	euid = 0, egid = 3
pcred	Process Management	/usr/proc/bin/pcred	file_dac_read, proc_mac_write, proc_nofloat, proc_owner
pcred	System Management	/usr/proc/bin/pcred	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
perf	Process Management	/etc/init.d/perf	euid = 0, egid = 3

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
pfiles	Process Management	/usr/proc/bin/pfiles	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pfiles	System Management	/usr/proc/bin/pfiles	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pflags	Process Management	/usr/proc/bin/pflags	file_dac_read, proc_mac_write, proc_nofloat, proc_owner
pflags	System Management	/usr/proc/bin/pflags	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pfsh	Basic Commands	/usr/bin/pfsh	
pg	Basic Commands	/usr/bin/pg	
ping	Basic Commands	/usr/sbin/ping	
ping	Network Management	/usr/sbin/ping	
ping	System Management	/usr/sbin/ping	
ping	System Security	/usr/sbin/ping	
pkgadd	Software Installation	/usr/sbin/pkgadd	euid = 0, egid = 2, file_chown, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_owner, file_setdac, file_setid, proc_nofloat, proc_owner, proc_setid, sys_devices, sys_minfree
pkgask	Software Installation	/usr/sbin/pkgask	euid = 0
pkgchk	Software Installation	/usr/sbin/pkgchk	euid = 0
pkginfo	Software Installation	/usr/bin/pkginfo	euid = 0

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
pkgmk	Software Installation	/usr/bin/pkgmk	euid = 0
pkgmv	Software Installation	/usr/sbin/pkgmv	euid = 0, egid = 2, file_chown, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_owner, file_setdac, file_setid, proc_nofloat, proc_owner, proc_setid, sys_devices, sys_minfree
pkgparam	Software Installation	/usr/bin/pkgparam	euid = 0
pkgproto	Software Installation	/usr/bin/pkgproto	euid = 0
pkgrm	Software Installation	/usr/sbin/pkgrm	euid = 0, egid = 2, file_chown, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_owner, file_setdac, file_setid, proc_owner, sys_devices, sys_minfree
pkgtrans	Software Installation	/usr/bin/pkgtrans	euid = 0
plabel	Process Management	/usr/proc/bin/plabel	file_dac_read, proc_mac_read, proc_nofloat, proc_owner, sys_trans_label
plabel	System Management	/usr/proc/bin/plabel	file_dac_read, proc_mac_read, proc_nofloat, proc_owner, sys_trans_label
pldd	Process Management	/usr/proc/bin/pldd	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pldd	System Management	/usr/proc/bin/pldd	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pmap	Process Management	/usr/proc/bin/pmap	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pmap	System Management	/usr/proc/bin/pmap	file_dac_read, proc_mac_read, proc_nofloat, proc_owner

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
poweroff	Maintenance and Repair	/usr/sbin/poweroff	euid = 0, sys_boot
ppriv	Object Privilege Management	/usr/proc/bin/ppriv	proc_mac_read, proc_owner
ppriv	Process Management	/usr/proc/bin/ppriv	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pprivtest	Object Privilege Management	/usr/proc/bin/pprivtest	proc_mac_read, proc_nofloat, proc_owner
pprivtest	Process Management	/usr/proc/bin/pprivtest	file_dac_read, file_mac_read, proc_nofloat, proc_owner
pr	Basic Commands	/usr/bin/pr	
praudit	Audit Review	/usr/sbin/praudit	euid = 0, min SL = ADMIN_HIGH, file_dac_read, sys_audit
printmgr	Printer Security	/opt/SUNWadm/2.3/bin/printmgr	min SL = ADMIN_LOW, max SL = ADMIN_LOW, file_dac_write, file_owner, proc_audit_tcb
printmgr	System Security	/opt/SUNWadm/2.3/bin/printmgr	all privileges
profmgr	User Security	/opt/SUNWadm/2.3/bin/profmgr	file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
prtconf	Maintenance and Repair	/usr/sbin/prtconf	
prun	Process Management	/usr/proc/bin/prun	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
prun	System Management	/usr/proc/bin/prun	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
ps	Process Management	/usr/bin/ps	file_dac_read, file_mac_read, proc_mac_read, proc_nofloat, proc_owner

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
ps	System Management	/usr/bin/ps	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
ps	System Security	/usr/bin/ps	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
psig	Process Management	/usr/proc/bin/psig	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
psig	System Management	/usr/proc/bin/psig	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pstack	Process Management	/usr/proc/bin/pstack	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pstack	System Management	/usr/proc/bin/pstack	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pstop	Process Management	/usr/proc/bin/pstop	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pstop	System Management	/usr/proc/bin/pstop	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
ptime	Process Management	/usr/proc/bin/ptime	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
ptime	System Management	/usr/proc/bin/ptime	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
ptree	Process Management	/usr/proc/bin/ptree	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
ptree	System Management	/usr/proc/bin/ptree	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pwait	Process Management	/usr/proc/bin/pwait	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pwait	System Management	/usr/proc/bin/pwait	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pwck	User Management	/usr/sbin/pwck	
pwck	User Security	/usr/sbin/pwck	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
pwconv	User Security	/usr/sbin/pwconv	all privileges
pwd	Basic Commands	/usr/bin/pwd	
pwdx	Process Management	/usr/proc/bin/pwdx	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
pwdx	System Management	/usr/proc/bin/pwdx	file_dac_read, proc_mac_read, proc_nofloat, proc_owner
rcp	Basic Commands	/usr/bin/rcp	
rdate	cron	/usr/bin/rdate	sys_config
rdist	Basic Commands	/usr/bin/rdist	
rdist	System Management	/usr/bin/rdist	
reboot	Maintenance and Repair	/usr/sbin/reboot	euid = 0, sys_boot
reject	Printer Security	/usr/sbin/reject	all privileges
reject	System Management	/usr/sbin/reject	sys_devices
reject	System Security	/usr/sbin/reject	sys_devices
rem_drv	Software Installation	/usr/sbin/rem_drv	euid = 0, egid = 3, min SL = ADMIN_LOW, max SL = ADMIN_LOW, file_dac_read, file_dac_write, file_mac_read, file_mac_write, sys_devices
rem_drv	System Security	/usr/sbin/rem_drv	sys_devices

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
remove_allocatable	Device Security	/usr/sbin/remove_allocatable	file_chown, file_dac_write, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_write, file_setdac
renice	Process Management	/usr/bin/renice	proc_mac_write, proc_owner
renice	System Management	/usr/bin/renice	proc_mac_write, proc_owner
rexecd	inetd	/usr/sbin/in.rexecd	net_privaddr, proc_audit_appl, proc_audit_tcb, proc_setid
rlogin	Basic Commands	/usr/bin/rlogin	
rlogind	inetd	/usr/sbin/in.rlogind	file_chown, file_mac_write, file_setdac, net_privaddr, proc_audit_appl, proc_audit_tcb
rm	Audit Control	/usr/bin/rm	file_dac_write, file_mac_write
rm	Basic Commands	/usr/bin/rm	
rm	cron	/usr/bin/rm	file_dac_read, file_dac_search, file_dac_write, file_mac_write
rmdir	Audit Control	/usr/bin/rmdir	file_dac_write
rmdir	Basic Commands	/usr/bin/rmdir	
rmdir	File System Management	/usr/bin/rmdir	file_dac_write
rootusr	Network Security	/etc/init.d/rootusr	euid = 0, egid = 3
route	Network Management	/usr/sbin/route	euid = 0, sys_net_config
rpc	Network Management	/etc/init.d/rpc	euid = 0, egid = 3

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
rpcbind	boot	/usr/sbin/rpcbind	min SL = ADMIN_HIGH, max SL = ADMIN_HIGH, all privileges
rsh	Basic Commands	/usr/ucb/rsh	
rshd	inetd	/usr/sbin/in.rshd	net_privaddr, proc_audit_appl, proc_audit_tcb, proc_setid
runpd	Object Privilege Management	/usr/sbin/runpd	
rup	Network Management	/usr/bin/rup	
rup	System Management	/usr/bin/rup	
ruptime	Network Management	/usr/bin/ruptime	
sadmind	inetd	/usr/sbin/sadmind	all privileges
savecore	boot	/usr/bin/savecore	min SL = ADMIN_HIGH, max SL = ADMIN_HIGH
script	Basic Commands	/usr/bin/script	
sdiff	Basic Commands	/usr/bin/sdiff	
sed	Audit Review	/usr/bin/sed	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH
sed	Basic Commands	/usr/bin/sed	
sendmail	Mail Management	/etc/init.d/sendmail	euid = 0, egid = 3

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
sendmail	Mail Management	/usr/lib/sendmail	euid = 0, file_mac_read, file_mac_search, net_privaddr, proc_nofloat, proc_setil
sendmail	boot	/usr/lib/sendmail	euid = 0, file_mac_read, file_mac_search, net_privaddr, proc_nofloat, proc_setil
serialmgr	Device Security	/opt/SUNWadm/2.3/bin/serialmgr	min SL = ADMIN_LOW, max SL = ADMIN_LOW, all privileges
serialmgr	System Security	/opt/SUNWadm/2.3/bin/serialmgr	all privileges
server	File System Management	/etc/init.d/nfs.server	euid = 0, egid = 3, max SL = ADMIN_LOW
setaudit	cron	/usr/bin/setaudit	file_dac_read, sys_audit
setfacl	Object Access Management	/usr/bin/setfacl	file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_setdac
setfattrflag	File System Management	/usr/bin/setfattrflag	file_audit, file_dac_search, file_mac_search, file_mac_write, file_owner
setfattrflag	Object Access Management	/usr/bin/setfattrflag	file_audit, file_dac_search, file_mac_search, file_mac_write, file_owner
setfattrflag	Object Label Management	/usr/bin/setfattrflag	file_audit, file_dac_search, file_mac_search, file_mac_write, file_owner
setfpriv	Object Privilege Management	/usr/bin/setfpriv	file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_owner, file_setid, file_setpriv
setfsattr	File System Security	/usr/sbin/setfsattr	all privileges

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
setfsattr	System Security	/usr/sbin/setfsattr	file_dac_write, file_downgrade_il, file_downgrade_sl, file_mac_search, file_setdac, file_setid, file_setpriv, file_upgrade_il, file_upgrade_sl
setlabel	Object Label Management	/usr/bin/setlabel	file_dac_read, file_dac_search, file_dac_write, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_nofloat, file_owner, file_upgrade_il, file_upgrade_sl
setuname	Network Management	/usr/bin/setuname	file_dac_read, file_dac_write, file_mac_read, file_mac_write, proc_nofloat
sh	Privileged Shells	/usr/bin/sh	all privileges
share	Audit Control	/usr/sbin/share	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs
share	File System Management	/usr/sbin/share	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs
share	System Management	/usr/sbin/share	euid = 0, sys_nfs
share	boot	/usr/sbin/share	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_mac_read, file_mac_search, file_mac_write, sys_nfs
shareall	Audit Control	/usr/sbin/shareall	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs
shareall	File System Management	/usr/sbin/shareall	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
shareall	System Management	/usr/sbin/shareall	euid = 0, sys_nfs
shareall	boot	/usr/sbin/shareall	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_mac_read, file_mac_search, file_mac_write, sys_nfs
showmount	File System Management	/usr/sbin/showmount	
showmount	System Management	/usr/sbin/showmount	
sleep	Basic Commands	/usr/bin/sleep	
snoop	Network Management	/usr/sbin/snoop	euid = 0, sys_net_config
sort	Basic Commands	/usr/bin/sort	
spell	Basic Commands	/usr/bin/spell	
spray	Network Management	/usr/sbin/spray	
standardmounts	File System Management	/etc/init.d/standardmounts	euid = 0, egid = 3
statd	boot	/usr/lib/nfs/statd	euid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, net_mac_read, net_privaddr, net_upgrade_sl, proc_dumpcore, proc_nofloat, sys_trans_label
strace	Device Security	/usr/sbin/strace	
stty	Basic Commands	/usr/bin/stty	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
swap	File System Management	/usr/sbin/swap	all privileges
swap	System Management	/usr/sbin/swap	sys_mount
swmtool	Software Installation	/usr/sbin/swmtool	euid = 0, egid = 2, file_chown, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_owner, file_setdac, file_setid, proc_nofloat, proc_owner, proc_setid, sys_devices, sys_minfree
sync	Basic Commands	/usr/sbin/sync	
sys	Network Management	/etc/init.d/sysid.sys	euid = 0, egid = 3
sysetup	Maintenance and Repair	/etc/init.d/sysetup	euid = 0, egid = 3
syslog	Maintenance and Repair	/etc/init.d/syslog	euid = 0, egid = 3
syslogd	Maintenance and Repair	/usr/sbin/syslogd	
syslogd	boot	/usr/sbin/syslogd	min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_dac_write, file_mac_write, file_upgrade_il, net_downgrade_il, net_downgrade_sl, net_mac_read, net_privaddr, net_upgrade_il, net_upgrade_sl, proc_nofloat, proc_setclr, proc_setil, proc_setsl, sys_trans_label
tail	Audit Review	/usr/bin/tail	euid = 0, min SL = ADMIN_HIGH, max SL = ADMIN_HIGH
tail	Basic Commands	/usr/bin/tail	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
tar	Media Backup	/usr/bin/tar	file_audit, file_dac_read, file_dac_search, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_upgrade_il, file_upgrade_sl, sys_trans_label
tar	Media Restore	/usr/bin/tar	file_audit, file_chown, file_dac_read, file_dac_search, file_dac_write, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_owner, file_setdac, file_setid, file_setpriv, file_upgrade_il, file_upgrade_sl, sys_devices, sys_trans_label
tbl	Basic Commands	/usr/bin/tbl	
telnetd	inetd	/usr/sbin/in.telnetd	file_chown, file_mac_write, file_setdac, net_privaddr, proc_audit_appl, proc_audit_tcb
test	Basic Commands	/usr/bin/test	
testfpriv	Object Privilege Management	/usr/bin/testfpriv	file_dac_search, file_mac_read, file_mac_search
tfind	Basic Commands	/usr/bin/tfind	
tfind	cron	/usr/bin/tfind	file_dac_read, file_dac_search
tftpd	inetd	/usr/sbin/in.tftpd	proc_audit_appl, proc_chroot, proc_owner, proc_setid
time	Basic Commands	/usr/bin/time	
tnchkdb	Network Security	/usr/sbin/tnchkdb	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
tnchkdb	System Security	/usr/sbin/tnchkdb	file_mac_read, sys_trans_label
tnctl	Network Security	/usr/sbin/tnctl	sys_net_config, sys_trans_label
tnctl	System Security	/usr/sbin/tnctl	sys_net_config, sys_trans_label
tnd	Network Security	/usr/sbin/tnd	net_downgrade_sl, net_mac_read, net_privaddr, proc_setclr, proc_setsl, sys_net_config
tnd	System Security	/usr/sbin/tnd	net_downgrade_sl, net_mac_read, net_privaddr, proc_setclr, proc_setsl, sys_net_config
tnd	boot	/usr/sbin/tnd	net_downgrade_sl, net_mac_read, net_privaddr, proc_setclr, proc_setsl, sys_net_config
tninfo	Network Security	/usr/sbin/tninfo	file_dac_read, file_mac_read, proc_nofloat, sys_net_config, sys_trans_label
tninfo	System Security	/usr/sbin/tninfo	file_dac_read, file_mac_read, sys_net_config, sys_trans_label
tokmapctl	Object Label Management	/usr/sbin/tokmapctl	net_mac_read, net_privaddr
tokmapd	Network Management	/usr/sbin/tokmapd	all privileges
touch	Basic Commands	/usr/bin/touch	
troff	Basic Commands	/usr/bin/troff	
true	Basic Commands	/usr/bin/true	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
true	Custom Admin Role	/usr/bin/true	
true	Custom Oper Role	/usr/bin/true	
truss	Process Management	/usr/bin/truss	
trusted_edit	Custom Root Role	/usr/dt/bin/trusted_edit	file_dac_read, file_dac_search, file_dac_write, proc_audit_appl, proc_audit_tcb
trusted_edit	Custom Secadmin Role	/usr/dt/bin/trusted_edit	file_dac_read, file_dac_search, file_dac_write, proc_audit_appl, proc_audit_tcb
tsol_audit_badpromlogins	Audit Control	/etc/init.d/ tsol_audit_badpromlogins	euid = 0, egid = 3
tsol_audit_badpromlogins	cron	/etc/init.d/ tsol_audit_badpromlogins	euid = 0, egid = 3
tsol_dev_allocate	Device Security	/etc/init.d/tsol_dev_allocate	euid = 0, egid = 3
tsol_dev_clean	Device Security	/etc/init.d/tsol_dev_clean	euid = 0, egid = 3
tsol_dev_policy	Device Security	/etc/init.d/tsol_dev_policy	euid = 0, egid = 3
tsol_label_services	Object Label Management	/etc/init.d/tsol_label_services	euid = 0, egid = 3
tsol_priv_enable	Object Privilege Management	/etc/init.d/tsol_priv_enable	euid = 0, egid = 3
tsol_sync_time	Maintenance and Repair	/etc/init.d/tsol_sync_time	euid = 0, egid = 3
tsol_tcb_verify	Network Management	/etc/init.d/tsol_tcb_verify	euid = 0, egid = 3
tsolxagent	required	/usr/dt/bin/tsolxagent	

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
ttdbserverd	inetd	/usr/dt/bin/rpc.ttdbserverd	file_chown, file_dac_read, file_dac_write, file_mac_read, file_mac_write, file_nofloat, file_owner, file_setdac, file_setid, net_broadcast, net_downgrade_il, net_downgrade_sl, net_mac_read, net_nofloat, net_privaddr, net_reply_equal, proc_mac_read, proc_mac_write, proc_owner
ttsession	required	/usr/dt/bin/ttsession	
tty	Basic Commands	/usr/bin/tty	
tunefs	Audit Control	/usr/sbin/tunefs	euid = 0, egid = 3, all privileges
tunefs	File System Management	/usr/sbin/tunefs	euid = 0, all privileges
ufs_quota	File System Management	/etc/init.d/ufs_quota	euid = 0, egid = 3
ufsdump	Media Backup	/usr/sbin/ufsdump	egid = 3, file_audit, file_dac_read, file_dac_search, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_upgrade_il, file_upgrade_sl, sys_trans_label
ufsrestore	Media Restore	/usr/sbin/ufsrestore	file_audit, file_chown, file_dac_read, file_dac_search, file_dac_write, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_nofloat, file_owner, file_setdac, file_setid, file_setpriv, file_upgrade_il, file_upgrade_sl, sys_trans_label

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
umount	Audit Control	/usr/sbin/umount	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
umount	File System Management	/usr/sbin/umount	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
umount	System Management	/usr/sbin/umount	euid = 0, file_dac_read, file_dac_search, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
umount	boot	/sbin/umount	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
umountall	Audit Control	/usr/sbin/umountall	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
umountall	File System Management	/usr/sbin/umountall	euid = 0, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
umountall	System Management	/usr/sbin/umountall	file_dac_read, file_dac_search, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
umountall	boot	/sbin/umountall	euid = 0, max SL = ADMIN_HIGH, file_dac_read, file_dac_search, file_dac_write, file_mac_read, file_mac_search, file_mac_write, net_privaddr, proc_setil, proc_setsl, sys_mount, sys_trans_label
uname	Basic Commands	/usr/bin/uname	
uncompress	Basic Commands	/usr/bin/uncompress	
uniq	Basic Commands	/usr/bin/uniq	
unshare	Audit Control	/usr/sbin/unshare	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs
unshare	File System Management	/usr/sbin/unshare	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs
unshare	System Management	/usr/sbin/unshare	euid = 0, sys_nfs
unshare	boot	/usr/sbin/unshare	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_mac_read, file_mac_search, file_mac_write, sys_nfs
unshareall	Audit Control	/usr/sbin/unshareall	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs
unshareall	File System Management	/usr/sbin/unshareall	euid = 0, egid = 0, file_mac_read, file_mac_search, file_mac_write, sys_nfs
unshareall	System Management	/usr/sbin/unshareall	euid = 0, sys_nfs

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
unshareall	boot	/usr/sbin/unshareall	euid = 0, egid = 0, min SL = ADMIN_LOW, max SL = ADMIN_HIGH, file_mac_read, file_mac_search, file_mac_write, sys_nfs
usermgr	User Management	/opt/SUNWadm/2.3/bin/usermgr	file_chown, file_dac_read, file_dac_write, proc_audit_tcb, sys_trans_label
usermgr	User Security	/opt/SUNWadm/2.3/bin/usermgr	file_chown, file_dac_read, file_dac_write, proc_audit_tcb, sys_trans_label
utmpd	User Management	/etc/init.d/utmpd	euid = 0, egid = 3
uucp	Network Management	/etc/init.d/uucp	euid = 0, egid = 3
vmstat	Maintenance and Repair	/usr/bin/vmstat	
vmstat	System Management	/usr/bin/vmstat	
volmgt	Device Security	/etc/init.d/volmgt	euid = 0, egid = 3
whereis	Basic Commands	/usr/ucb/whereis	
which	Basic Commands	/usr/bin/which	
who	Basic Commands	/usr/bin/who	
whoami	Basic Commands	/usr/ucb/whoami	
writeaudit	Audit Control	/usr/bin/writeaudit	proc_audit_appl, proc_audit_tcb

TABLE A–3 Commands and their Associated Execution Profiles *(continued)*

Command	Profile	Path	Security Attributes
xhost	Basic Commands	/usr/openwin/bin/xhost	
xhost	System Management	/usr/openwin/bin/xhost	win_config

Finding Actions in Execution Profiles

Table A–4 lists each action contained in any execution profile and the execution profile(s) to which it is assigned. Remember that an action may be contained in more than one execution profile. The table also indicates any security attributes assigned to the action: minimum sensitivity label, maximum sensitivity label, setUID value, setGID value, and privileges.

TABLE A–4 Actions and their Associated Execution Profiles

Actions	Profiles	Security Attributes
AddAllocDev	Device Security	file_chown, file_dac_write, file_downgrade_sl, file_mac_read, file_mac_write, file_setdac, sys_trans_label
AuditClass	Audit Control	min label = ADMIN_LOW, max label = ADMIN_LOW, file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
AuditControl	Audit Control	min label = ADMIN_LOW, max label = ADMIN_LOW, file_dac_read, file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
AuditEvent	Audit Control	min label = ADMIN_LOW, max label = ADMIN_LOW, file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
AuditStartup	Audit Control	min label = ADMIN_LOW, max label = ADMIN_LOW, file_dac_read, file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb

TABLE A–4 Actions and their Associated Execution Profiles *(continued)*

Actions	Profiles	Security Attributes
AuditUser	Audit Control	min label = ADMIN_LOW, max label = ADMIN_LOW, file_dac_read, file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
BuildDataBaseRequest	Basic Actions	
CheckEncodings	Network Security	
CheckEncodings	Object Label Management	file_dac_read, file_dac_write, proc_audit_appl, proc_audit_tcb, sys_trans_label
Compose	Basic Actions	
Compress	Basic Actions	
DNS_Resolve	Network Management	file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
DNS_Resolve	System Management	file_dac_write, proc_audit_appl, proc_audit_tcb
Dbmgr	Mail Management	file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
Dbmgr	Network Management	min label = ADMIN_LOW, file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
Dbmgr	Network Security	min label = ADMIN_LOW, file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
Dbmgr	System Management	file_chown, file_dac_write, sys_trans_label
Dbmgr	System Security	all privileges
Dbmgr	User Security	min label = ADMIN_LOW, max label = ADMIN_LOW, file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
Df	Basic Actions	
Diff	Basic Actions	
DtPrint	Basic Actions	
DtPrintManager	Basic Actions	
DtTTMediaOpen	Basic Actions	
DtUnlink	Basic Actions	

TABLE A–4 Actions and their Associated Execution Profiles *(continued)*

Actions	Profiles	Security Attributes
Dtappmgr	Basic Actions	
Dtcalc	Basic Actions	
Dtcm	Basic Actions	
Dtdevmgr	Basic Actions	
Dtdevmgr	dtwm	
Dtfile	Basic Actions	
Dtfile	Object Access Management	
Dtfile	Object Label Management	
DtfileHome	Basic Actions	
DtfileHome	Object Access Management	
DtfileHome	Object Label Management	
DtfileHome	Object Privilege Management	
Dthelpview	Basic Actions	
Dthelpview	Basic Actions	
Dtmail	Basic Actions	
Dtmanpageview	Basic Actions	
Dtpad	Basic Actions	
Dtprintinfo	Basic Actions	
Dtprintinfo	Basic Actions	
Dtstyle	dtwm	max label = ADMIN_LOW
Dtterm	Basic Actions	

TABLE A–4 Actions and their Associated Execution Profiles *(continued)*

Actions	Profiles	Security Attributes
DttermConsole	Basic Actions	
Dttrash	Basic Actions	
Dttrash	Object Access Management	
Dttrash	Object Label Management	
DuSort	Basic Actions	
EditEncodings	Object Label Management	euid = 0, egid = 3, file_dac_read, file_dac_write, proc_audit_appl, proc_audit_tcb
EditMotd	Network Management	file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
EditMotd	System Management	max label = ADMIN_LOW, file_dac_write, proc_audit_appl, proc_audit_tcb
Env	Basic Actions	
ExitSession	dtwm	
FPHelp	Basic Actions	
FileProperties	Basic Actions	
Grep	Basic Actions	
Groupmgr	User Management	min label = ADMIN_LOW, max label = ADMIN_LOW, file_dac_write, proc_audit_tcb
Hostmgr	Network Management	min label = ADMIN_LOW, file_dac_write, proc_audit_tcb
Hostmgr	System Management	file_chown, file_dac_write, sys_trans_label
InvokeFILEMGR	Basic Actions	
InvokeFILEMGR	Object Access Management	
InvokeFILEMGR	Object Privilege Management	
InvokeMAILER	Basic Actions	

TABLE A–4 Actions and their Associated Execution Profiles *(continued)*

Actions	Profiles	Security Attributes
LockDisplay	dtwm	
Niscat	NIS+ Administration	
Niscat	Network Security	
Niscat_o	NIS+ Administration	
Nisclient	NIS+ Security Administration	euid = 0
Nispopulate	NIS+ Security Administration	euid = 0
Nisserver	NIS+ Security Administration	euid = 0
Nsswitch	Network Management	file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
Nsswitch	System Security	file_dac_write, proc_audit_appl, proc_audit_tcb
OWanswerbook	Basic Actions	
OWtapetool	Media Backup	file_audit, file_dac_read, file_dac_search, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_upgrade_il, file_upgrade_sl, sys_trans_label
OWtapetool	Media Restore	
Open	Basic Actions	
OpenCD	Basic Actions-ROM	
OpenDtIntro	Basic Actions	
OpenFloppy	Basic Actions	
OpenFolder	Basic Actions	
OpenTerminal	Basic Actions	
Print	Basic Actions	
Printermgr	Printer Security	min label = ADMIN_LOW, max label = ADMIN_LOW, file_dac_write, file_owner, proc_audit_tcb

TABLE A–4 Actions and their Associated Execution Profiles *(continued)*

Actions	Profiles	Security Attributes
Printermgr	System Security	all privileges
Profmgr	User Security	min label = ADMIN_LOW, max label = ADMIN_LOW, file_chown, file_dac_write, proc_audit_tcb, sys_trans_label
ReOpenRestrictedFolder	Basic Actions	
ReloadActions	Basic Actions	
ReloadActionsNotice	Basic Actions	
ReloadApps	Basic Actions	
ReloadResources	Basic Actions	
Rm	Basic Actions	
SDTaccessx	dtwm	
SDTimage	Basic Actions	
SDtPersonalBookmarks	Basic Actions	
SDtSampleBookmarks	Basic Actions	
SDtWebClient	Basic Actions	
Selconfig	Object Label Management	file_dac_read, file_dac_write, proc_audit_appl, proc_audit_tcb
SendMail	Mail Management	file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
SendMail	System Security	file_dac_write, proc_audit_appl, proc_audit_tcb
Serialmgr	Device Security	min label = ADMIN_LOW, max label = ADMIN_LOW, all privileges
Serialmgr	System Security	all privileges
SetRoutes	Network Management	file_dac_read, file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
ShareFS	File System Management	file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
ShareFS	System Management	file_dac_write, proc_audit_appl, proc_audit_tcb

TABLE A–4 Actions and their Associated Execution Profiles *(continued)*

Actions	Profiles	Security Attributes
StartDtscreenBlank	dtwm	
StartDtscreenFlame	dtwm	
StartDtscreenHop	dtwm	
StartDtscreenImage	dtwm	
StartDtscreenLife	dtwm	
StartDtscreenPyro	dtwm	
StartDtscreenQix	dtwm	
StartDtscreenRotor	dtwm	
StartDtscreenSwarm	dtwm	
StartDtscreenWorm	dtwm	
Tar	Media Backup	file_audit, file_dac_read, file_dac_search, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_upgrade_il, file_upgrade_sl, sys_trans_label
TarList	Media Backup	file_audit, file_dac_read, file_dac_search, file_downgrade_il, file_downgrade_sl, file_mac_read, file_mac_search, file_mac_write, file_upgrade_il, file_upgrade_sl, sys_trans_label
TarUnpack	Media Restore	
Terminal	Basic Actions	
TextEditor	Basic Actions	
Tnchkdb	Network Management	
Tnchkdb	Network Security	
Tnchkdb	System Security	
Tnchkdb_nisplus	Network Management	
Tnchkdb_nisplus	Network Security	

TABLE A–4 Actions and their Associated Execution Profiles *(continued)*

Actions	Profiles	Security Attributes
Tnchkdb_nisplus	System Security	
Trash	Basic Actions	
TrustedEditor	Custom Root Role	file_dac_read, file_dac_search, file_dac_write, proc_audit_appl, proc_audit_tcb
TrustedEditor	Custom Secadmin Role	file_dac_read, file_dac_search, file_dac_write, proc_audit_appl, proc_audit_tcb
TrustedEditor	System Security	file_dac_read, file_dac_write, proc_audit_appl, proc_audit_tcb
Tsolgateways	Network Management	file_dac_read, file_dac_write, proc_audit_appl, proc_audit_tcb
Usermgr	User Management	min label = ADMIN_LOW, max label = ADMIN_LOW, file_chown, file_dac_read, file_dac_write, proc_audit_tcb, sys_trans_label
Usermgr	User Security	min label = ADMIN_LOW, max label = ADMIN_LOW, file_chown, file_dac_read, file_dac_write, proc_audit_tcb, sys_trans_label
Vfstab	File System Management	file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
Vfstab	System Management	file_dac_write, proc_audit_appl, proc_audit_tcb
Vfstab_adjunct	File System Security	file_dac_write, file_setdac, proc_audit_appl, proc_audit_tcb
Vfstab_adjunct	System Security	file_dac_write, proc_audit_appl, proc_audit_tcb
WebBrowser	Basic Actions	
Xrefresh	Basic Actions	

Index

B

C

D

E

F

G

H

I

K

L

M

R

S

T

U

V

W

X