Advanced System Administration for the Solaris™ 9 Operating Environment
SA-299

Student Guide

Sun Microsystems, Inc.
UBRM05-104
500 Eldorado Blvd.
Broomfield, CO 80021
U.S.A.

Revision A

THIS MANUAL IS DESIGNED TO SUPPORT AN INSTRUCTOR-LED TRAINING (ILT) COURSE AND IS INTENDED TO BE USED FOR REFERENCE PURPOSES IN CONJUNCTION WITH THE ILT COURSE. THE MANUAL IS NOT A STANDALONE TRAINING TOOL. USE OF THE MANUAL FOR SELF-STUDY WITHOUT CLASS ATTENDANCE IS NOT RECOMMENDED.

Export Control Classification Number (ECCN): 5E992

Please
Recycle

CE MANUEL DE RÉFÉRENCE DOIT ÊTRE UTILISÉ DANS LE CADRE D'UN COURS DE FORMATION DIRIGÉ PAR UN INSTRUCTEUR (ILT). IL NE S'AGIT PAS D'UN OUTIL DE FORMATION INDÉPENDANT. NOUS VOUS DÉCONSEILLONS DE L'UTILISER DANS LE CADRE D'UNE AUTO-FORMATION.

Please
Recycle

Table of Contents

About This Course

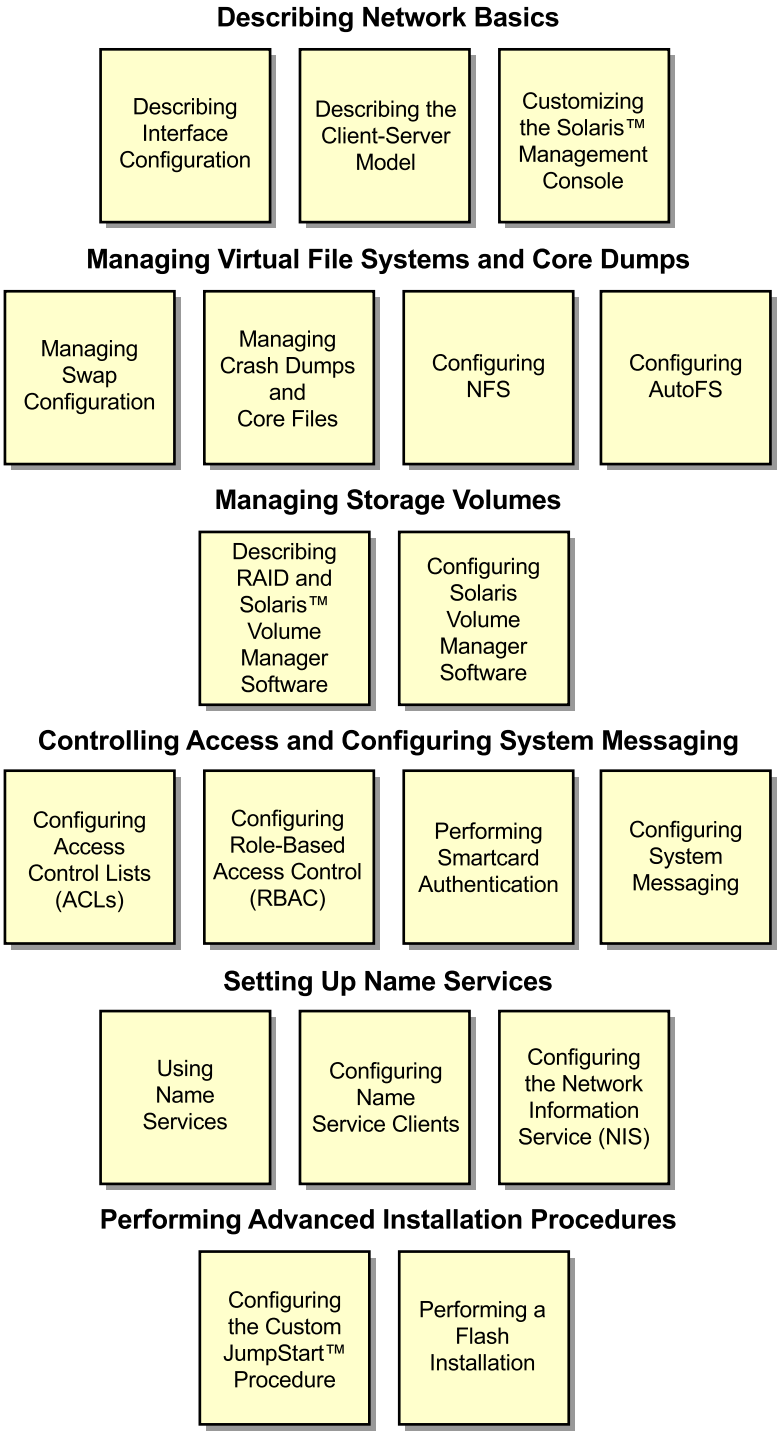
Instructional Goals

Upon completion of this course, you should be able to:

- Describe network basics
- Manage virtual file systems and core dumps
- Manage storage volumes
- Control access and configure system messaging
- Set up name services
- Perform advanced installation procedures

Course Map

The course map enables you to see what you have accomplished and where you are going in reference to the instructional goals.



Topics Not Covered

This course does not cover the following topics. Many of these topics are covered in other courses offered by Sun Educational Services:

- Basic UNIX® commands – Covered in SA-119: *Fundamentals of Solaris™ 9 Operating Environment for System Administrators*
- The `vi` editor – Covered in SA-119: *UNIX® Essentials Featuring the Solaris™ 9 Operating Environment*
- Basic UNIX file security – Covered in SA-119: *UNIX® Essentials Featuring the Solaris™ 9 Operating Environment*
- Software package administration – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- Patch maintenance – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- Adding users using the Solaris Management Console software – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- Basic system security – Covered in SA-119: *UNIX® Essentials Featuring the Solaris™ 9 Operating Environment*
- Administering initialization files – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- Advanced file permissions – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- Backup and recovery – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- The `lp` print service and print commands – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- Process control – Covered in SA-239: *Intermediate System Administration for the Solaris™ 9 Operating Environment*
- Hardware or software troubleshooting – Covered in ST-350: *Sun™ Systems Fault Analysis Workshop*
- System tuning – Covered in SA-400: *Enterprise System Performance Management*

- Detailed shell programming – Covered in SA-245: *Shell Programming for System Administrators*
- Detailed network administration concepts – Covered in SA-399: *Network Administration for the Solaris™ 9 Operating Environment*

  Refer to the Sun Educational Services catalog for specific information on course content and registration.

How Prepared Are You?

To be sure you are prepared to take this course, can you answer yes to the following questions?

- Can you install and boot the Solaris 9 Operating Environment (Solaris 9 OE) on a standalone workstation?
- Can you implement basic system security?
- Can you add users to the system using the Solaris™ Management Console software?
- Can you use the `pkgadd` command to add software packages?
- Can you set file permissions using access control lists (ACLs)?
- Can you monitor and mount file systems?
- Can you manage disk devices and processes?
- Can you perform backups and restorations?

Introductions

Now that you have been introduced to the course, introduce yourself to the other students and the instructor, addressing the following items:

- Name
- Company affiliation
- Title, function, and job responsibility
- Experience related to topics presented in this course
- Reasons for enrolling in this course
- Expectations for this course

How to Use Course Materials

To enable you to succeed in this course, these course materials use a learning module that is composed of the following components:

- Objectives – You should be able to accomplish the objectives after completing a portion of instructional content. Objectives support goals and can support other higher-level objectives.
- Lecture – The instructor will present information specific to the objective of the module. This information will help you learn the knowledge and skills necessary to succeed with the activities.
- Activities – The activities take on various forms, such as an exercise, self-check, discussion, and demonstration. Activities are used to facilitate the mastery of an objective.
- Visual aids – The instructor might use several visual aids to convey a concept, such as a process, in a visual form. Visual aids commonly contain graphics, animation, and video.

Note – Many system administration tasks for the Solaris™ Operating Environment (Solaris OE) can be accomplished in more than one way. The methods presented in the courseware reflect recommended practices used by Sun Educational Services.

Conventions

The following conventions are used in this course to represent various training elements and alternative learning resources.

Icons

Discussion – Indicates a small-group or class discussion on the current topic is recommended at this time.

Note – Indicates additional information that can help students but is not crucial to their understanding of the concept being described. Students should be able to understand the concept or complete the task without this information. Examples of notational information include keyword shortcuts and minor system adjustments.

Caution – Indicates that there is a risk of personal injury from a nonelectrical hazard, or risk of irreversible damage to data, software, or the operating system. A caution indicates that the possibility of a hazard (as opposed to certainty) might happen, depending on the action of the user.

Typographical Conventions

`Courier` is used for the names of commands, files, directories, user names, host names, programming code, and on-screen computer output; for example:

> Use the `ls -al` command to list all files.
> `host1# cd /home`

`Courier bold` is used for characters and numbers that you type; for example:

> To list the files in this directory, type the following:
> `# ` **`ls`**

`Courier italics` is used for variables and command-line placeholders that are replaced with a real name or value; for example:

> To delete a file, use the `rm` *`filename`* command.

`Courier italic bold` is used to represent variables whose values are to be entered by the student as part of an activity; for example:

> Type **`chmod a+rwx`** ***`filename`*** to grant read, write, and execute rights for *`filename`*.

Palatino italics is used for book titles, new words or terms, or words that you want to emphasize; for example:

> Read Chapter 6 in the *User's Guide*.
>
> These are called *class* options.

Module 1

Describing Interface Configuration

Objectives

The network interfaces that a system uses to communicate with other systems on the network use both hardware and software configuration components. When adding a network interface to a system, you must configure specific files to establish a relationship between the hardware and the software addresses.

Upon completion of this module, you should be able to:

- Control and monitor network interfaces
- Configure Internet Protocol Version 4 (IPv4) interfaces at boot time

The following course map shows how this module fits into the current instructional goal.

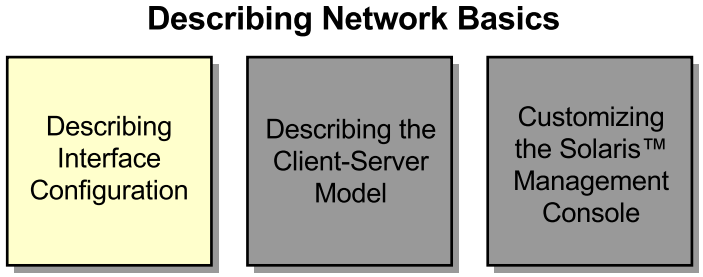


Figure 1-1 Course Map

Controlling and Monitoring Network Interfaces

Network commands, such as the `ifconfig` command, the `ping` command, and the `snoop` command, control and monitor the functionality of network interfaces.

Displaying the MAC Address

The media access control (MAC) address is your computer's unique hardware address on a local area network (LAN). The MAC address is also the Ethernet address on an Ethernet LAN. When you are connected to a LAN, an address resolution table maps your computer's physical MAC address to an Internet Protocol (IP) address on the LAN. Two ways to display the MAC address or the Ethernet address are:

- Use the `ifconfig -a` command
- Use the boot programmable read-only memory (PROM) `banner` command

Note – The MAC address is displayed only if the `root` user issues the `ifconfig` command. Only the IP address information is displayed if a non-`root` user issues the `ifconfig` command.

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

The MAC address is listed as `8:0:20:93:c9:af` in this example.

You can also retrieve the MAC address from a system that has not yet been booted by performing the `banner` command at the `ok` prompt.

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 300MHz), Keyboard Present
OpenBoot 3.31 256 MB (60ns) memory installed, Serial #9685423.
Ethernet address 8:0:20:93:c9:af, Host ID: 8093c9af.
```

Displaying the IP Address

The `ifconfig` command displays the current configuration for a network interface.

With the `-a` option, the `ifconfig` command displays the current configuration for all network interfaces in the system.

```
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

The previous example shows that the loopback interface (`lo0`) is up, running, and configured with an IP address of `127.0.0.1`. The `hme0` interface is up, running, and configured with an IP address of `192.168.30.41`.

Marking an Ethernet Interface as Down

When an Ethernet interface is marked as down, it means that it cannot communicate. You can use the `ifconfig` command to mark an Ethernet interface as up or down. For example, to mark the `hme0` interface as down, perform the commands:

```
# ifconfig hme0 down
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

Note – The `UP` flags are no longer present. When an interface is flagged as `UP`, it is ready to communicate.

The following example shows that when you mark an interface as up, the UP status appears in the flags field of the ifconfig command output:

```
# ifconfig hme0 up
# ifconfig -a
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
        inet 192.168.30.41 netmask ffffff00 broadcast 192.168.30.255
        ether 8:0:20:93:c9:af
```

Sending ICMP ECHO_REQUEST Packets

To determine if you can contact another system over the network, enter the ping command:

```
# ping sys41
sys41 is alive
```

A response of no answer from sys41 indicates that you cannot contact host sys41. This implies a problem with host sys41, or a problem with the network.

For the ping command to succeed, the following conditions must be satisfied:

- The interface must be plumbed.
- The interface must be configured.
- The interface must be up.
- The interface must be physically connected.
- The interface must have valid routes configured.

Note – Configuration of routes is an advanced networking topic. Detailed network administration concepts are covered in SA-399*: Network Administration for the Solaris™ 9 Operating Environment*.

Capturing and Inspecting Network Packets

You can use the `snoop` utility to capture and inspect network packets to determine what kind of data is transferred between systems. You can use the `snoop` utility to see what happens when one system uses the `ping` command to communicate with another system. To view network traffic between two specific systems, perform the command:

```
# snoop sys41 sys42
sys41 -> sys42 ICMP Echo request
sys42 -> sys41 ICMP Echo reply
```

Use the `-a` option to enable audible clicks, which notify you of any network traffic. Although noisy, the clicks are useful when troubleshooting.

The following example shows how to turn on audible clicks for all network traffic related to a Dynamic Host Configuration Protocol (DHCP) boot:

```
# snoop -a dhcp
```

Some additional `snoop` options include:

`snoop`	Summary output
`snoop -V`	Summary verbose output
`snoop -v`	Detailed verbose output
`snoop -o` *`filename`*	Redirects the `snoop` utility output to *`filename`* in summary mode
`snoop -i` *`filename`*	Displays packets that were previously captured in *`filename`*

Note – Press Control-C to stop the `snoop` utility.

Configuring IPv4 Interfaces at Boot Time

This section describes the files and scripts involved with configuring IPv4 network interfaces.

Introducing IPv4 Interface Files

You can get a basic understanding of network interfaces within the Solaris™ Operating Environment (Solaris OE) by learning the function of only a few files. Three of these files are:

- The `/etc/rcS.d/S30network.sh` file
- The `/etc/hostname.`*xxn* file
- The `/etc/inet/hosts` file

The `/etc/rcS.d/S30network.sh` File

The `/etc/rcS.d/S30network.sh` file is one of the startup scripts that runs each time you boot the system. This script uses the `ifconfig` utility to configure each interface with an IP address and other required network information. The script searches for files called `hostname.`*xxn* in the `/etc` directory, where *xx* is an interface type and *n* is the instance of the interface. For every file named `/etc/hostname.`*xxn,* the script uses the `ifconfig` command with the `plumb` option to make the `kernel` ready to talk to this type of interface. The script then configures the named interface. The `/etc/hostname.hme0` file is an example of an interface configuration file.

Note – The `/etc/rcS.d/S30network.sh` file first appeared in the Solaris 8 OE. It is functionally similar to the `/etc/rcS.d/S30rootusr.sh` file in previous Solaris OE releases.

The /etc/hostname.*xxn* File

The /etc/hostname.*xxn* file contains an entry that configures a corresponding interface. The variable component of the file name is replaced by an interface type and a number that differentiates between multiple interfaces of the same type configured in the system. Table 1-1 shows some examples.

Table 1-1 The /etc/hostname.*xxn* File Entries and Corresponding Interfaces

Entry	Interface
/etc/hostname.le0	First le Ethernet interface in the system
/etc/hostname.hme0	First hme Ethernet interface in the system
/etc/hostname.hme1	Second hme Ethernet interface in the system
/etc/hostname.qfe0	First qfe Ethernet interface in the system
/etc/hostname.eri0	First eri Ethernet interface in the system

The codes for the interface types are product codes. These codes originate from varying sources. For example, the le code is an abbreviation of the original interface, Lance Ethernet, and the qfe code is an abbreviation for Quadfast Ethernet.

The /etc/hostname.hme0 file contains either the host name or the IP address of the system that contains the hme0 interface. The host name contained in the file must exist in the /etc/hosts file so that it can be resolved to an IP address at system boot time. You can edit the /etc/hostname.hme0 file to contain either the host name or the IP address from the /etc/hosts file.

```
# cat /etc/hostname.hme0
sys41
```

or

```
# cat /etc/hostname.hme0
192.168.30.41
```

The /etc/inet/hosts File

The /etc/inet/hosts file is a local database that associates the IP addresses of hosts with their names. You can use the /etc/inet/hosts file with, or instead of, other hosts databases, including the Domain Name System (DNS), the Network Information Service (NIS) hosts map, and the Network Information Service Plus (NIS+) hosts table. Programs use library interfaces to access information in the /etc/inet/hosts file.

The /etc/inet/hosts file contains at least the loopback and host information. The file has one entry for each IP address of each host. If a host has more than one IP address, this file will have one entry for each address, on separate lines. The format of each line is:

IP-address official-host-name [*nicknames*] . . .

Items are separated by any number of space or tab characters. The first item on a line is the host's IP address. The second entry is the host's official name. Subsequent entries on the same line are alternative names for the same machine, or nicknames. Nicknames are optional.

For a host with more than one IP address, consecutive entries for these addresses will contain different host names.

```
# cat /etc/inet/hosts
.
< output truncated>
.
127.0.0.1localhost
.
< output truncated>
.
192.168.30.41  sys41               loghost      #connection to hme
interface
192.168.4.1    sys41 -internal                #connection to qfe interface
.
<output truncated>
.
```

Note – The /etc/inet/hosts file is the official (system V release 4) SVr4 name of the hosts file. The symbolic link /etc/hosts exists for Berkeley Software Distribution (BSD) compatibility.

Changing the System Host Name

The host name of a system is contained in six files on the system. You must modify all of these files to successfully change a system's host name. The files that contain the host name of a system are:

- The `/etc/nodename` file
- The `/etc/hostname.`*xxn* file
- The `/etc/inet/hosts` file
- The `/etc/net/ticlts/hosts` file
- The `/etc/net/ticots/hosts` file
- The `/etc/net/ticotsord/hosts` file

Editing the `/etc/nodename` File

Each Solaris OE has a canonical name, which is the official name used when referring to a system. By convention, the system name is the same as the host name associated with the IP address of the primary network interface; for example, `hostname.hme0`.

The following example shows a system's `/etc/nodename` file:

```
# cat /etc/nodename
sys41
```

You can change the canonical name by editing the `/etc/nodename` file, and rebooting the system.

If the machine's network configuration is managed remotely and delivered by the DHCP or remote procedure calls (RPC) bootparams protocols, the `/etc/nodename` file is not used. The file is not used because the remote service delivers the canonical name.

Editing the `/etc/hostname.`*xxn* File

The `/etc/hostname.`*xxn* file contains either the host name or the IP address of the system that contains the named interface.

Editing the /etc/inet/hosts File

Network addresses are written in the conventional decimal-dot notation.

Host names are text strings up to 24 characters. Alphabetic characters, numbers, the minus sign, and a period are allowed in the host name. Periods are only allowed when they serve to delimit components of domain style names. Blanks and spaces are not allowed in the host name. No distinction is made between uppercase and lowercase characters. The first character must be an alphabetic character. The last character must not be a minus sign (-) or a dot (.).

A pound sign (#) indicates the beginning of a comment. After a comment character, all characters, up to the end of the line, are not interpreted.

Editing the Three Transport Layer Independent (TLI) Files

The /etc/net directory contains three subdirectories: /etc/net/ticlts, /etc/net/ticots, and /etc/net/ticotsord. Each of these directories contains a hosts file. These files contain configuration information for transport-independent network services. If these files become corrupted, unpredictable results can occur when trying to resolve the system host name when using network services. In addition, when you execute the /usr/sbin/sys-unconfig command, the system deletes all of the hosts files. If the files get corrupted or deleted, you can use any editor to restore them. The format of the file is:

hostname hostname

The two occurrences of the host name are separated by white space. For example, each of these files for a host named sys41 would contain:

```
sys41 sys41
```

The sys-unconfig Command

You can use the /usr/sbin/sys-unconfig command to undo a system's configuration.

You can use the /usr/sbin/sys-unconfig command to restore a system's configuration to an as-manufactured state, ready to be reconfigured again. The system's configuration includes a host name, NIS domain name, time zone, IP address, IP subnet mask, and root password.

The `sys-unconfig` command does the following:

- Saves the current `/etc/inet/hosts` file information in the `/etc/inet/hosts.saved` file.
- If the current `/etc/vfstab` file contains Network File System (NFS) mount entries, it saves the `/etc/vfstab` file to the `/etc/vfstab.orig` file.
- Restores the default `/etc/inet/hosts` file.
- Removes the default host name in the `/etc/hostname.`*xxn* files for all configured interfaces.
- Removes the default domain name in the `/etc/defaultdomain` file.
- Restores the time zone to `PST8PDT` in the `/etc/TIMEZONE` file.
- Resets naming services to local files.
- Removes the entries for this host in the `/etc/net/tic*/hosts` file.
- Removes the `/etc/inet/netmasks` file.
- Removes the `/etc/defaultrouter` file for naming services.
- Removes the password set for the `root` user in the `/etc/shadow` file.
- Removes the `/etc/.rootkey` file for NIS+.
- Executes all system configuration applications. These applications are defined by prior executions of a `sysidconfig -a` command.
- Removes the `/etc/resolv.conf` file for DNS.
- Disables Lightweight Directory Access Protocol (LDAP) by removing:
  - The `/var/ldap/ldap_client_cache` file
  - The `/var/ldap/ldap_client_file` file
  - The `/var/ldap/ldap_client_cred` file
  - The `/var/ldap/cachemgr.log` file

When the `sys-unconfig` command is finished, it performs a system shutdown. The `sys-unconfig` command is a potentially dangerous utility and can only be run by the `root` user.

When you restart the system, a configuration scripts prompts you to configure the system information. The `sys-unconfig` command is not available on diskless clients.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: The Solaris OE Network Commands (Level 1)

In this exercise, you use basic network-related commands.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in each `/etc/hosts` file on each system.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Tasks

Complete the following steps:

- Allow the `snoop` utility to run through this exercise.
- Use the `ifconfig` command to list the IP address, Ethernet address, netmask, and current status of your primary network interface. Record this information. Start a `snoop` session on both systems, and monitor the output.
- Use the `ping` command to contact your partner's system, and record the `snoop` output. On one system, mark the primary interface as down. Record the new `ifconfig` output for this interface. Use the `ping` command to contact that host, and record related `snoop` output.

Exercise: The Solaris OE Network Commands (Level 2)

In this exercise, you use basic network-related commands.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in each `/etc/hosts` file on each system.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Task Summary

Perform the following tasks:

- Allow the `snoop` utility to run through this exercise.
- Use the `ifconfig` command to list the IP address, Ethernet address, netmask, and current status of your primary network interface. Record this information. Start a `snoop` session on both systems, and monitor the output.
- Use the `ping` command to contact your partner's system, and record the `snoop` output. On one system, mark the primary interface as down. Record the new `ifconfig` output for this interface. Use the `ping` command to contact that host, and record related `snoop` output including:
  - How many requests the `ping` command makes
  - What the `ping` command requests have in common

Tasks

Complete the following steps using the `ifconfig` utility, the `ping` command, and the `snoop` utility.

1. On both systems, log in as the `root` user, and open a terminal window. Using the `ifconfig` command, display basic configuration information about your network interfaces.

   For your primary interface (usually `hme0`), what does the `ifconfig` command report for the following attributes? Enter your values into Table 1-2.

Table 1-2 Primary Interface Values

Attribute	Value
IP address	
Ethernet address	
Interface up/down	

2. On both systems, open a new terminal window. In the new window, enter the `snoop` command to display the network traffic between your two systems only.
3. Use the `ping` command to verify that your system can contact the network interface on your partner's system.
4. Observe the output from the `snoop` command. Which protocol does the `ping` command use?

   Does the `snoop` output contain requests and replies (yes or no)?

   Requests: Replies:
5. On one system, use the `ifconfig` command to mark its primary interface as down and then again to display its configuration information.

   Has anything changed in the information that the `ifconfig` command reports?

   _____
6. On the system whose interface remains up, attempt to use the `ping` command to contact the system whose interface is down.

   What does the `ping` command report?

   _____

7. Observe the output from the `snoop` utility on both systems. How does the `snoop` output differ from the `ping` command output before and after you marked the interface as down?

   How many requests does the `ping` command send by default?

   _____

   Does the target system see the `ping` command requests? If so, how are these requests handled?

   _____

8. On the system whose interface is down, use the `ifconfig` command to mark its primary interface as up. Check that the change took place.
9. On the system whose interface remained up, use the `ping` command to contact the other system.

   What does the `ping` command report?

   _____

   Does the `snoop` utility report a reply from the target host?

   _____

Exercise: The Solaris OE Network Commands (Level 3)

In this exercise, you use basic network-related commands.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in each `/etc/hosts` file on each system.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Task Summary

Complete the following steps:

- Allow the `snoop` utility to run through this exercise.
- Use the `ifconfig` command to list the IP address, Ethernet address, netmask, and current status of your primary network interface. Record this information. Start a `snoop` session on both systems, and monitor the output.
- Use the `ping` command to contact your partner's system, and record the `snoop` output. On one system, mark the primary interface as down. Record the new `ifconfig` output for this interface. Use the `ping` command to contact that host, and record related `snoop` output including:
  - How many requests the `ping` command makes
  - What the `ping` command requests have in common

Tasks and Solutions

This section describes the tasks for you to perform, and lists the solutions. Complete the following steps using the `ifconfig` utility, the `ping` command, and the `snoop` utility.

1. On both systems, log in as the `root` user, and open a terminal window. Using the `ifconfig` command, display basic configuration information about your network interfaces.

   For your primary interface (usually `hme0`), what does the `ifconfig` command report for the following attributes? Enter your values into Table 1-3.

Table 1-3 Primary Interface Values

Attribute	Value
IP address	*It varies according to the system in use.*
Ethernet address	*It varies according to the system in use.*
Interface up/down	*The interface should be* `UP`.

2. On both systems, open a new terminal window. In the new window, enter the `snoop` command to display the network traffic between your two systems only.

```
# snoop host1 host2
```

3. Use the `ping` command to verify that your system can contact the network interface on your partner's system.

```
# ping host
```

4. Observe the output from the `snoop` command. Which protocol does the `ping` command use?

   ICMP

   Does the `snoop` output contain requests and replies (yes or no)?

   Requests: *Yes* Replies: *Yes*

5. On one system, use the `ifconfig` command to mark its primary interface as down and then again to display its configuration information.

   Has anything changed in the information that the `ifconfig` command reports?

   The `ifconfig` *command no longer lists the interface as* `UP`.

6. On the system whose interface remains up, attempt to use the `ping` command to contact the system whose interface is down.

   What does the `ping` command report?

   After a time-out period, the `ping` *command reports* `no answer from host`.

7. Observe the output from the `snoop` utility on both systems. How does the `snoop` output differ from the `ping` command output before and after you marked the interface as down?

   The `snoop` *utility only shows the* `ping` *command requests—no replies.*

   How many requests does the `ping` command send by default?

   Twenty

   Does the target system see the `ping` command requests? If so, how are these requests handled?

   Yes it does, but it does not send a reply.

8. On the system whose interface is down, use the `ifconfig` command to mark its primary interface as up. Check that the change took place.

```
# ifconfig hme0 up
# ifconfig hme0
```

9. On the system whose interface remained up, use the `ping` command to contact the other system.

   What does the `ping` command report?

   The host is alive.

   Does the `snoop` utility report a reply from the target host?

   Yes.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 2

Describing the Client-Server Model

Objectives

The client-server model describes the communication process between two computers or programs. The client system makes a service request from the server system, then the server system fulfills the request. Although programs can use the client-server model internally in a single computer, the model is more widely used across a network. The client-server model provides a way to distribute services efficiently across multiple locations on a network.

Upon completion of this module, you should be able to:

- Describe client-server processes
- Start server processes

The following course map shows how this module fits into the current instructional goal.

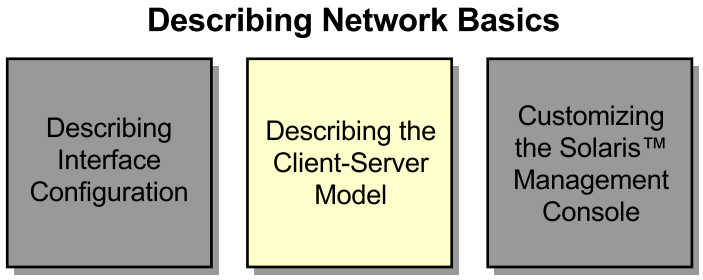


Figure 2-1 Course Map

Introducing Client-Server Processes

The client-server model describes network services and the client programs of those services. One example of the client-server relationship is the name server and resolver model of the DNS. Another example of the client and server relationship is the NFS.

Introducing Client Processes

Figure 2-2 shows a client-server process relationship. The client is a host or a process that uses services from another program, known as a server. You can apply the client-server relationship to computer programs within a single computer or use the relationship across a network to make one application server a host to one or more application clients.

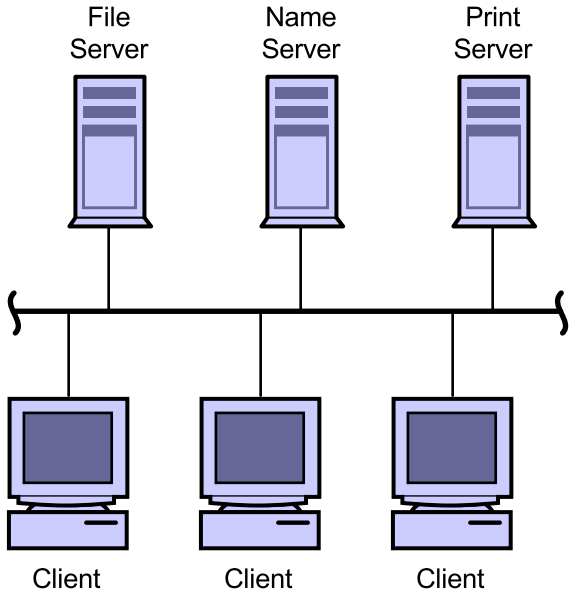


Figure 2-2 Client Processes

Examples of clients in the Solaris 9 OE are:

- For name services, a client is a host system that uses either the NIS+, NIS, DNS, or LDAP name service lookup provided by the name service server.
- In file systems, the client is a system that remotely accesses the resources of a storage server, such as a server with large disk and network capacity.
- For applications, such as `sendmail` or calendar manager, the client accesses services from a server process.

Introducing Server Processes

The server is a host or a process that provides services to another program known as a client. Client-server computing is a key factor in supporting network computing. The client-server model on the network can be multilayered. Figure 2-3 shows that multiple hosts on a subnet can be clients to a single storage host server. Multiple hosts serve as an interface to storage arrays. The storage clients rely on the storage server to access their data. Conversely, one of the storage clients, such as a printer host, can be configured to act as the interface for network printers. To perform print operations from the storage host, the storage host must assume a print client role when communicating with the print server role of the printer host.

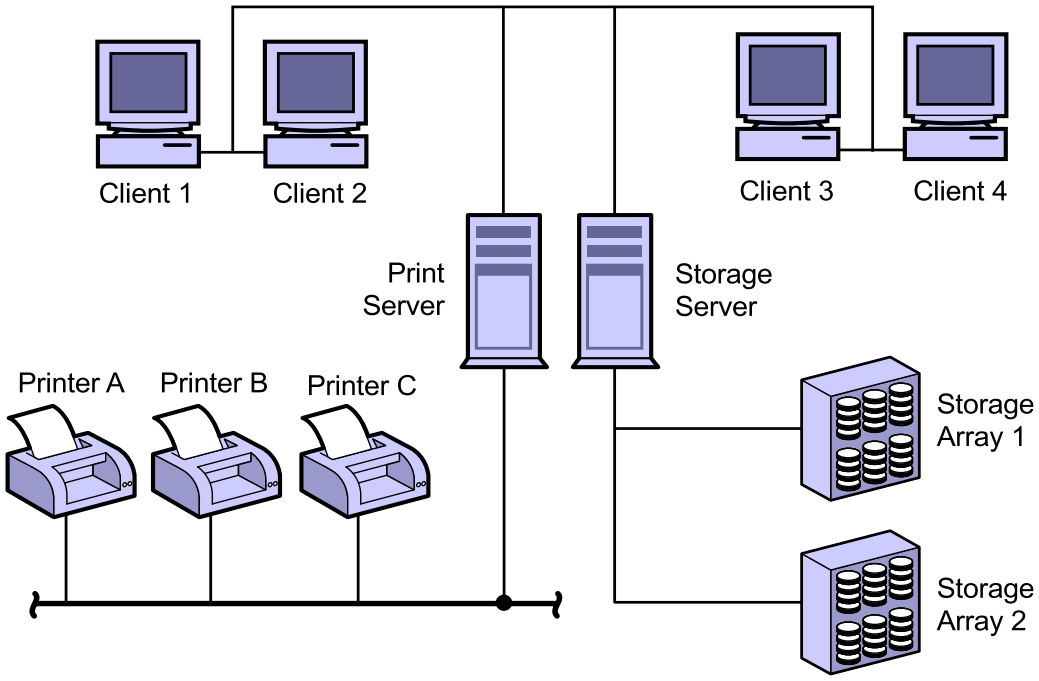


Figure 2-3 Server Processes

Examples of servers in the Solaris 9 OE are:

- A host system providing name services to a network in NIS+, NIS, DNS, and LDAP.
- A host system providing disk space to the network, such as a server with large disk and network capacity.
- A host system providing windowing services to applications. The client and the server can run on the same system or on separate systems.
- A host system providing web services to client systems.

Starting Server Processes

To start services for server processes, you must know which files to use for automatic service configuration. You must also know how to manually start the services.

Introducing the Internet Service Daemon (inetd)

The inetd daemon is a special network process that runs on each system and starts server processes that do not automatically start at boot time. The inetd daemon is the server process for both the standard Internet services and Sun Remote Procedure Call (Sun RPC) services. The inetd daemon starts at boot time using the /etc/rc2.d/S72inetsvc script. A configuration file lists the services that the inetd daemon will listen for and start in response to network requests. If you do not specify a configuration file, the inetd daemon uses the default /etc/inet/inetd.conf file.

To list some examples of services that the inetd daemon listens for, perform the command:

```
# cat /etc/inet/inetd.conf
.
.(output truncated)
.
# TELNETD - telnet server daemon
telnet  stream  tcp6    nowait  root    /usr/sbin/in.telnetd    in.telnetd
# smserverd to support removable media devices
100155/1        tli     rpc/ticotsord   wait    root
/usr/lib/smedia/rpc.smserverd rpc.smserverd
# REXD - rexd server provides only minimal authentication
#rexd/1 tli     rpc/tcp wait    root    /usr/sbin/rpc.rexd rpc.rexd
# FTPD - FTP server daemon
ftp     stream  tcp6    nowait  root    /usr/sbin/in.ftpd       in.ftpd -a
.
.(output truncated)
.
```

When the `inetd` daemon receives a network request, it runs the associated command in the `inetd.conf` file. The previous example shows three examples of remote services. Each server entry is a single line in the following form:

```
service-name endpoint-type protocol wait-status uid server-program \
server-arguments
```

Table 2-1 describes the server entry fields.

Table 2-1 Server Entry Descriptions

Field	Description
service-name	The name of a valid service listed in the `/etc/services` file.
endpoint-type	The value can be one of the following: • `stream` for a stream socket • `dgram` for a datagram socket • `raw` for a raw socket • `seqpacket` for a sequenced packet socket • `tli` for all TLI endpoints
protocol	A recognized protocol listed in the `/etc/inet/protocols` file. For servers that support the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) over the Internet Protocol Version 6 (IPv6) address, the `tcp6` and `udp6` protocol types are also recognized but are not listed in the `/etc/inet/protocols` file.
wait-status	This field has values `wait` or `nowait`. The `wait` keyword is usually associated with UDP servers and informs the `inetd` daemon that it should not listen for additional incoming requests for this service until the current server exits. The `nowait` keyword is usually associated with TCP servers and indicates that the `inetd` daemon continues to listen for incoming requests even while the current server is running.
uid	The user ID under which the server should run.
server-program	The path name of a server program that the `inetd` daemon invokes to provide a requested service, or the value `internal` if the `inetd` daemon itself provides the service.
server-arguments	To invoke a server with command-line arguments, the entire command line (including the command itself) must appear in this field (which consists of all remaining words in the entry).

Note – By specifying a protocol value of `tcp6` or `udp6` for a service, the `inetd` daemon passes the given daemon an `AF_INET6` socket. The following daemons have been modified to accept `AF_INET6` sockets and service connection requests coming from either IPv4 or IPv6-based transports: `ftp`, `telnet`, `shell`, `login`, `exec`, `tftp`, `finger`, and `printer`. Modified services do not usually require separate configuration lines for `tcp` or `udp`.

The `inetd` daemon starts a server process when it receives an appropriate service request. The `in.ftpd` server process can be invoked by the `inetd` daemon each time a connection to the File Transfer Protocol (FTP) service is requested as shown in the following example:

```
# grep ftp /etc/inet/inetd.conf
ftp   stream   tcp6    nowait   root   /usr/sbin/in.ftpd     in.ftpd -a
```

When changing the `/etc/inet/inetd.conf` file, send a hang-up (`HUP`) signal to the `inetd` process to force it to reread the configuration file. To force the `inetd` process to re-read the configuration file, perform the command:

```
# pkill -HUP inetd
```

Note – To turn off a service, add a # symbol to the beginning of the line corresponding to that service in the `/etc/inetd.conf` file, and send a `HUP` request.

Introducing Network Ports

Network ports help transport protocols distinguish between multiple service requests arriving at a given host computer. The TCP and UDP transport protocols identify ports using a positive integer between 1 and 65535, which is called a port number. Network ports can be divided into two categories, well-known ports and ephemeral (short-lived) ports.

Port Numbers

There are two fundamental approaches to port assignments:

- Central authority:
  - All users must agree to allow the central authority to assign all port numbers.
  - The central authority is responsible for publishing the list of port number assignments, called well-known port assignments.
  - Well-known port assignments dictate software requirements on a system.
- Dynamic binding:
  - The ports are unknown to the client in advance. The system software dynamically assigns ports to the programs that require them.
  - To obtain the current port assignments on any computer, the software generates a request to the target machine for the port number information. The target machine then responds with the port number.
  - These port number assignments are considered ephemeral since assignments are short lived, only lasting until the system is rebooted.

Many system applications support network services. Each network service uses a port that represents an address space reserved for that service. If a port number is not pre-assigned, the operating system allows an application to choose an unused port number. A client often communicates with a server through a well-known port. Well-known ports are stored in the `/etc/inet/services` file. To view the well-known port that the `telnet` service uses, perform the command:

```
# grep telnet /etc/inet/services
telnet          23/tcp
```

This example shows that the `telnet` service uses well-known port 23 and uses the TCP protocol.

Starting Services That Use a Well-Known Port

The list of services that use a well-known port includes:

- Services that start by default at system boot time
- Services that do not start automatically at boot, and must start on demand

Starting Well-Known Port Services at Boot Time

One of the well-known port services that starts at boot time is the `sendmail` process. The `sendmail` process uses well-known port 25 to perform network services for email using the Simple Mail Transport Protocol (SMTP). You can confirm that the name has been translated to the port number by searching for the `mail` entry in the `/etc/inet/services` file. To confirm the translation, perform the command:

```
# grep mail /etc/inet/services
smtp            25/tcp          mail
```

The `sendmail` process is initialized by the startup script `/etc/rc2.d/S88sendmail` when you boot the Solaris 9 OE. Because the `sendmail` process uses port 25, the `sendmail` process starts listening at port 25 for incoming mail activity soon after start up. There is no need for the `inetd` daemon to listen at port 25 for incoming `sendmail` requests or to start `sendmail`, because the `sendmail` process is already running.

Starting Well-Known Port Services on Demand

The `telnet` service is a well-known port service that does not automatically start at boot time. Figure 2-4 shows the process by which well-known services are started on demand. The `telnet` service uses the `inetd` daemon to listen for network requests, so that the `telnet` service does not have to continually run on the system. When the `inetd` daemon receives a network request at a port, it uses the information listed in the `/etc/inet/service` file to determine which service to start.

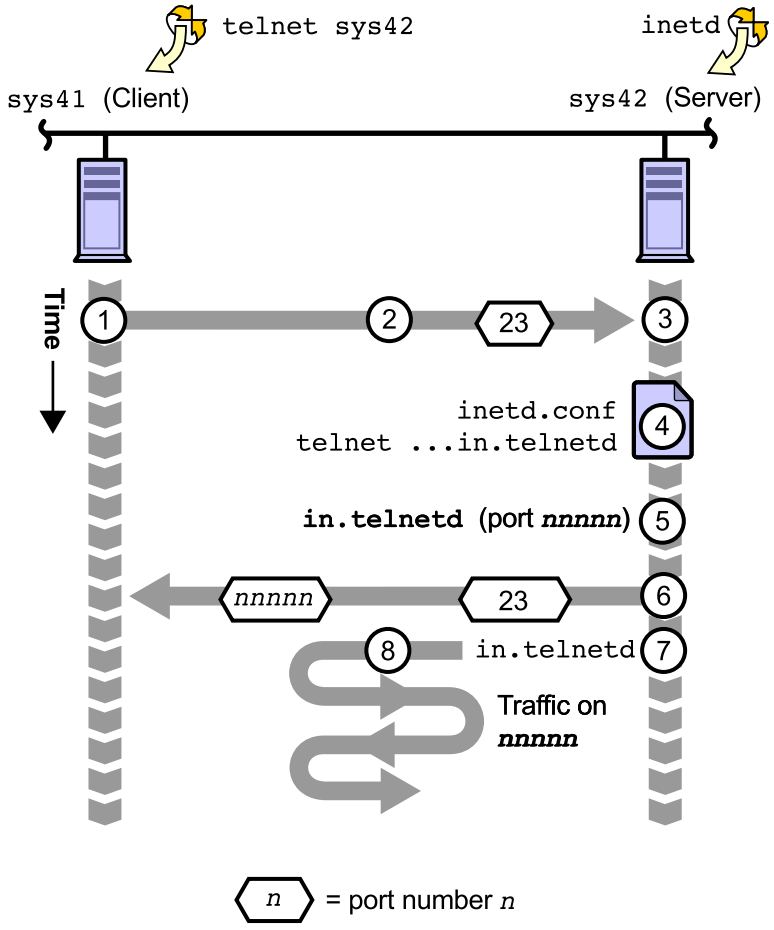


Figure 2-4 Requesting a Well-Known Service

The steps to connect to the `telnet` service are:

1. The initiating host `sys41` executes the network service to request a connection to the receiving host `sys42` by executing the `telnet sys42` command.
2. The `telnet` service is a well-known service. The port for this service is port `23`.
3. The `telnet` packet requesting a connection goes to port `23` on the host `sys42`.
4. Initially, the `inetd` daemon listens at port `23` for the `telnet` service. The `telnet sys42` command on `sys41` generates a request to port `23` that `inetd` recognizes as a `telnet` request because of the configuration entry in the `/etc/inet/services` file.
5. The `telnet` service does not continuously run on a system waiting for a connection. The `inetd` daemon must start the `telnet` service dynamically on demand.
6. The `inetd` daemon consults the `/etc/inetd.conf` file to find a matching entry for the requested service. The `inetd` daemon identifies the `telnet` service line.
7. The `inetd` daemon executes the `in.telnetd` process from the `/etc/inetd.conf` file. The `in.telnetd` daemon takes control of the current `telnet` session's communication.
8. The `in.telnetd` daemon receives this session's traffic and runs on port `23` until this `telnet` session ends.

Note – The `inetd` daemon continues to listen for new service requests.

Starting RPC Services

RPC services are services developed using a set of utilities developed by Sun Microsystems, Inc. While RPC services are assigned a unique program number by the programmer when they are written, the RPC services are not typically assigned to well-known ports.

The list of RPC services includes:

- Services that start by default at system boot time
- Services that do not start automatically at boot and must start on demand

Starting RPC Services at Boot Time

RPC services started at boot time with startup scripts run on available ports above `32768`. The `rpcbind` process associates RPC program numbers with port numbers. The `rpcbind` service must be running on the server system for you to make RPC requests to the server. When an RPC service starts at boot, it communicates the following information to the `rpcbind` process:

- The port with which it is associated
- The RPC program number

If a client wants to make an RPC call to a given program number, it must first contact the `rpcbind` service on the server machine to obtain the port address before it can send the RPC requests. If the RPC service has registered its current port number with the `rpcbind` daemon during startup, the current port number of the RPC service is returned to the client.

When you boot the Solaris 9 OE, the `/etc/rc2.d/s71rpc` startup script initializes the `rpcbind` service. The port number used by the `rpcbind` daemon is listed in the `/etc/inet/services` file. After the system starts up, the `rpcbind` daemon starts listening at port `111`. To view the port number and protocol, perform the command:

```
# grep rpcbind /etc/services
sunrpc          111/udp         rpcbind
sunrpc          111/tcp         rpcbind
```

Starting RPC Services on Demand

Some `rpcbind` services start only on demand. The port numbers are registered with the `rpcbind` process during boot. Figure 2-5 shows the steps involved in requesting an RPC port address. When a client application requests a service, the `rpcbind` process returns the port number of the service to the client machine. The client machine generates a new request using the port number that it just received for the requested service.

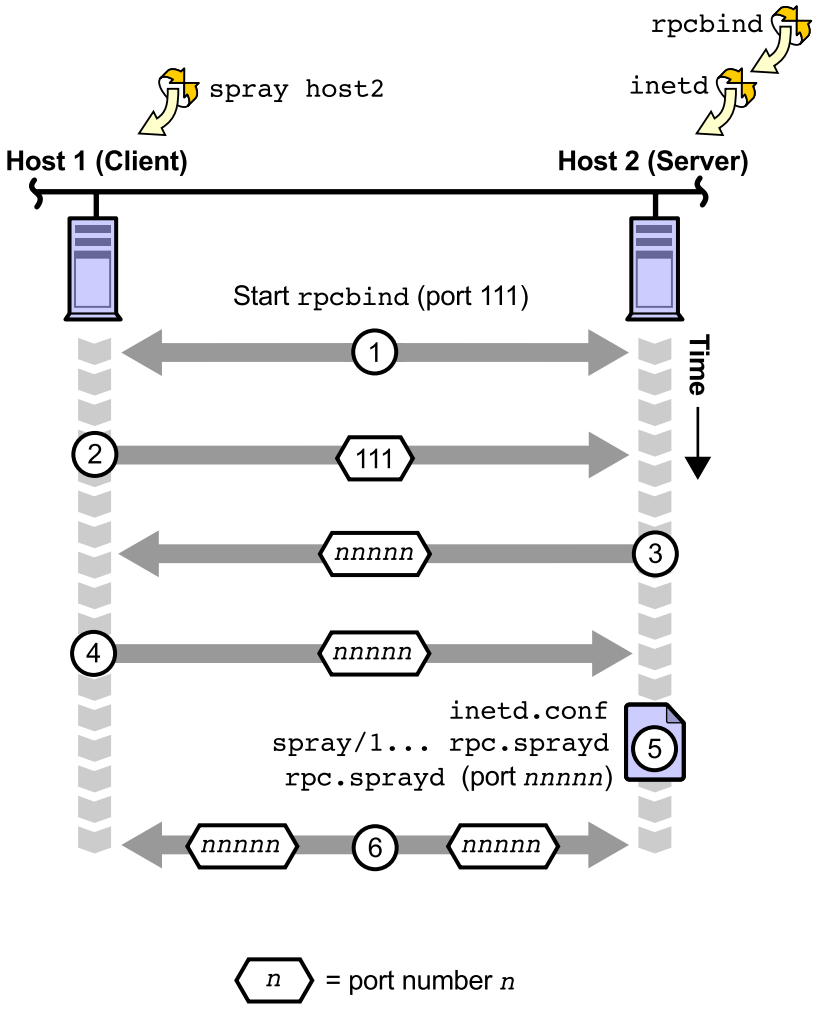


Figure 2-5 Requesting an RPC Address

RPC services on demand, such as the `sprayd` service, are implemented as follows:

1. The `rpcbind` daemon is started on all systems by a startup script. The `sprayd` service is listed in both the `/etc/rpc` and `/etc/inetd.conf` files and, therefore, registers its current port assignment and program number with the `rpcbind` process during boot.
2. A user on `host1` issues a `spray` command to `host2`. The `spray` request is initially addressed to port `111` and contains the program number of the `sprayd` service.
3. The `rpcbind` daemon on the `host2` server reads the program number and determines that the request is for the `sprayd` service. The `rpcbind` daemon returns the current port number of the `sprayd` service to the `host1` client.
4. The `host1` client sends a second request to the port number of the `sprayd` service on the `host2` server. The `inetd` daemon receives the request.
5. The `inetd` daemon consults entries in the `/etc/inetd.conf` file to find a matching entry for the service request. The `inetd` daemon starts the `sprayd` service.
6. This `rpc.sprayd` daemon takes over the `spray` session's communication.

Using the rpcinfo Commands

The rpcinfo command makes an RPC call to an RPC server, and reports what it finds. Two frequently used options to the rpcinfo command are -p and -d.

Listing Registered RPC Services

To list all the services registered with the rpcbind process, enter the rpcinfo command as follows:

```
rpcinfo -p  [ host ]
```

For example:

```
# rpcinfo -p
   program vers proto   port  service
    100000    4   tcp    111  rpcbind
    100000    3   tcp    111  rpcbind
    100000    2   tcp    111  rpcbind
    100000    4   udp    111  rpcbind
    100000    3   udp    111  rpcbind
    100000    2   udp    111  rpcbind
    100232   10   udp  32772  sadmind
    100083    1   tcp  32771
...
<output truncated>
...
```

This command returns a columnar output that includes the:

- Program number
- Version number of the RPC program number
- RPC protocol
- Port number
- RPC service

Note – Using the command rpcinfo -p *host* command returns information about registered RPC services on the specified *host*.

Deleting RPC Service Registration

To unregister the RPC service given a specified *prognum* (program number) and *versnum* (version number), perform the `rpcinfo` command:

```
rpcinfo -d prognum versnum
```

For example:

```
# rpcinfo -d 100012 1
```

This command unregisters the RPC service with program number `100012` and version number `1`.

Note – When using the `rpcinfo -d` command to unregister an RPC service, the RPC service can be identified using either the service name or the program number.

The deleted RPC service that uses program number `100012` is `sprayd`. To register the `sprayd` service again, send a `HUP` signal to the `inetd` daemon as follows:

```
# pkill -HUP inetd
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Observing the Solaris OE Network (Level 1)

In this exercise, you use basic, network-related commands to observe the `inetd` daemon and the `rpcbind` services.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in the `/etc/hosts` file on each system.
- Eliminate entries for the `root` user in `/etc/ftpd/ftpusers` file to ensure that the `root` user is not restricted from using the FTP service on both systems.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Tasks

Perform the following tasks:

- Monitor the network traffic throughout the exercise.
- Check that the FTP application is listed in the `/etc/inetd.conf` file and the `/etc/services` file. Record the name of the FTP server daemon. On both systems, check if the FTP application or server daemon is running. Use one system as the FTP client and the other as the FTP server. Establish an FTP connection, and check again for `ftp` command-related applications and daemons. Record your observations. Terminate your FTP connection.
- Check the port number assigned to the `rpcbind` service to make sure that it is a well-known port. Record the port number. Check and record the port number and program number assigned to the `sprayd` daemon. Check that your partner's system can contact your system using the `sprayd` daemon. Unregister the `sprayd` service. Check that the service has unregistered.
- Check that the `sprayd` daemon does not function from your partner's system to your system. Send the `HUP` signal to the `inetd` daemon, and check that the `sprayd` service is again a registered service, and that the `sprayd` service functions correctly between the two systems. Check the new port number assigned to the `sprayd` service and the program number that it uses.

Exercise: Observing the Solaris OE Network (Level 2)

In this exercise, you use basic, network-related commands to observe the `inetd` daemon and the `rpcbind` services.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in the `/etc/hosts` file on each system.
- Eliminate entries for the `root` user in `/etc/ftpd/ftpusers` file to ensure that the `root` user is not restricted from using the FTP service on both systems.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Task Summary

Perform the following tasks:

- Monitor the network traffic throughout the exercise.
- Check that the FTP application is listed in the `/etc/inetd.conf` file and the `/etc/services` file. Record the name of the FTP server daemon. On both systems, check if the FTP application or server daemon is running. Use one system as the FTP client and the other as the FTP server. Establish an FTP connection, and check again for `ftp` command-related applications and daemons. Record your observations. Terminate your FTP connection.
- Check the port number assigned to the `rpcbind` service to make sure that it is a well-known port. Record the port number. Check and record the port number and program number assigned to the `sprayd` daemon. Check that your partner's system can contact your system using the `sprayd` daemon. Unregister the `sprayd` service. Check that the service has unregistered.
- Check that the `sprayd` daemon does not function from your partner's system to your system. Send the `HUP` signal to the `inetd` daemon, and check that the `sprayd` service is again a registered service, and that the `sprayd` service functions correctly between the two systems. Check the new port number assigned to the `sprayd` service and the program number that it uses.

Tasks

Perform the following tasks.

Task 1– Interaction Between the `inetd` Daemon and the FTP Application

You must use two additional windows on the FTP client host for this section of the exercise. Complete the following steps:

1. In a dedicated terminal window, open a `snoop` session between the two hosts used during this exercise. This `snoop` session should remain active throughout this exercise.
2. Display the entry for the FTP application in the `/etc/inetd.conf` file, and record the name of the server daemon that is listed.
3. Check that the FTP application is a service with a well-known port listed in the `/etc/services` file.

   Is it listed?

   _____

4. Use the `pgrep` command to check if the `ftp` daemon is currently running.

   Is it running?

   _____

Note – Determine which system acts as the FTP client and which acts as the FTP server.

5. On the FTP client, in one window, establish an FTP connection to the FTP server.

6. On the FTP client in another window, check for daemons or applications related to the FTP service.

   What does the `pgrep` command report?

   _____

7. On the FTP server, in an available window, check for daemons and applications related to the FTP service.

   What does the `pgrep` command display?

   _____

8. On the FTP client, terminate your FTP connection to the server.
9. On both the FTP server and client, check for FTP-related daemons and applications.

   What does the `pgrep` command display?

   _____

10. Observe the output from the `snoop` utility on both systems. What FTP-related login information does the `snoop` command display?

    _____

11. Change the client-server roles of the two systems, and repeat Step 5 through Step 9.

Task 2 – The `rpcbind` Service Operations

Complete the following steps:

1. Use the `rpcinfo` command to display information for the `rpcbind` process.

   Which port number does the `rpcbind` process use?

   _____

   Which protocols does the `rpcbind` process use?

   _____

2. Check that the `rpcbind` service is listed in the `/etc/services` file, and that the listed port number matches the output from the `rpcinfo` command in the previous step.

   Does it?

   _____

3. Use the `rpcinfo` command to display information for the `sprayd` service.

   Which port number is the `sprayd` service using?

   _____

   Which program number is the `sprayd` service using?

   _____

4. Check the `/etc/services` file to determine if the `sprayd` service has been assigned a well-known port number.

   Has it?

   _____

5. Check the `/etc/rpc` file to see if the `sprayd` service is listed.

   Is it listed?

   _____

6. Check that your system will respond to the `sprayd` service requests. Have your partner run the `spray` command, and specify your system as the target.
7. Use the `rpcinfo` command to unregister the `sprayd` service's port number. Check that the `sprayd` service is no longer listed as a registered port number.
8. Have your partner run the `spray` command, and specify your system as the target again.

   What message does the `spray` command return?

   _____

9. Send the `HUP` signal to the `inetd` daemon for the `rpcbind` service to register all services listed in its configuration file.
10. Verify that the `sprayd` service is listed as a registered service.

    What port number is the `sprayd` service using now?

    _____

    Is the program number used by the `sprayd` service the same as the program number that was listed in Step 3?

    _____

11. To check that the `sprayd` service can now contact your system, have your partner run the `spray` command, and specify your system as the target.
12. Stop the `snoop` processes running on both systems.

Exercise: Observing the Solaris OE Network (Level 3)

In this exercise, you use basic, network-related commands to observe the `inetd` daemon and the `rpcbind` services.

Preparation

To prepare for this exercise, perform the following tasks:

- Check that you have two systems listed in the `/etc/hosts` file on each system.
- Eliminate entries for the `root` user in `/etc/ftpd/ftpusers` file to ensure that the `root` user is not restricted from using the FTP service on both systems.
- Work with a partner for this exercise, and perform all steps on both systems, unless noted otherwise.

Task Summary

Perform the following tasks:

- Monitor the network traffic throughout the exercise.
- Check that the FTP application is listed in the `/etc/inetd.conf` file and the `/etc/services` file. Record the name of the FTP server daemon. On both systems, check if the FTP application or server daemon is running. Use one system as the FTP client and the other as the FTP server. Establish an FTP connection, and check again for `ftp` command-related applications and daemons. Record your observations. Terminate your FTP connection.
- Check the port number assigned to the `rpcbind` service to make sure that it is a well-known port. Record the port number. Check and record the port number and program number assigned to the `sprayd` daemon. Check that your partner's system can contact your system using the `sprayd` daemon. Unregister the `sprayd` service. Check that the service has unregistered.

- Check that the sprayd daemon does not function from your partner's system to your system. Send the HUP signal to the inetd daemon, and check that the sprayd service is again a registered service, and that the sprayd service functions correctly between the two systems. Check the new port number assigned to the sprayd service and the program number that it uses.

Tasks and Solutions

This section describes the tasks for you to perform and lists the solutions.

Task 1 – Interaction Between the inetd Daemon and the FTP Application

You must use two additional windows on the FTP client host for this section of the exercise. Complete the following steps:

1. In a dedicated terminal window, open a snoop session between the two hosts used during this exercise. This snoop session should remain active throughout this exercise.

```
# snoop host1 host2
```

2. Display the entry for the FTP application in the /etc/inetd.conf file, and record the name of the server daemon that is listed.

```
# grep ftp /etc/inetd.conf
#       ftp telnet shell login exec tftp finger printer
# TFTPD - tftp server (primarily used for booting)
#tftp   dgram   udp6    wait    root    /usr/sbin/in.tftpd    in.tftpd -s
/tftpboot
ftp     stream  tcp6    nowait  root    /usr/sbin/in.ftpd     in.ftpd -a

                in.ftpd
```

3. Check that the FTP application is a service with a well-known port listed in the /etc/services file.

```
# grep ftp /etc/services
ftp-data        20/tcp
ftp             21/tcp
tftp            69/udp
```

Is it listed?

Yes. It uses port 21.

4. Use the pgrep command to check if the ftp daemon is currently running.

```
# pgrep -xl ftpd
#
```

Is it running?

No. It should not be running yet.

Note – Determine which system acts as the FTP client and which acts as the FTP server.

5. On the FTP client, in one window, establish an FTP connection to the FTP server.

```
# ftp host
Connected to host.
220 host FTP server ready.
Name (host:root):
331 Password required for root.
Password:xxxxxxx
230 User root logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

6. On the FTP client in another window, check for daemons or applications related to the FTP service.

```
# pgrep -l ftp
nnn ftp
```

What does the pgrep command display?

The pgrep *command should list the FTP application if the system is acting as an FTP client.*

7. On the FTP server, in an available window, check for daemons and applications related to the FTP service.

```
# pgrep -l ftp
nnnn in.ftpd
```

What does the pgrep command display?

The pgrep *command should list the* in.ftpd *daemon if the system is acting as an FTP server.*

8. On the FTP client, terminate your FTP connection to the server.

```
ftp> bye
```

9. On both the FTP server and FTP client, check for FTP-related daemons and applications.

```
# pgrep -l ftp
```

What does the `pgrep` command report?

Nothing. Both the FTP application and FTP server daemon have terminated.

10. Observe the output from the `snoop` utility on both systems. What FTP-related login information does the `snoop` command display?

The login name and password in clear text.

11. Change the client-server roles of the two systems, and repeat Step 5 through Step 9.

Task 2 – The `rpcbind` Service Operations

Complete the following steps:

1. Use the `rpcinfo` command to display information for the `rpcbind` process.

```
# rpcinfo -p |grep rpcbind
    100000    4   tcp    111  rpcbind
    100000    3   tcp    111  rpcbind
    100000    2   tcp    111  rpcbind
    100000    4   udp    111  rpcbind
    100000    3   udp    111  rpcbind
    100000    2   udp    111  rpcbind
```

Which port number does the `rpcbind` process use?

`111`

Which protocols does the `rpcbind` process use?

Both TCP and UDP.

2. Check that the `rpcbind` service is listed in the `/etc/services` file, and that the listed port number matches the output from the `rpcinfo` command in Step 1.

```
# grep rpcbind /etc/services
sunrpc          111/udp         rpcbind
sunrpc          111/tcp         rpcbind
```

Does it?
Yes

3. Use the `rpcinfo` command to display information for the `sprayd` service.

```
# rpcinfo -p |grep sprayd
    100012    1   udp  32777  sprayd
```

Which port number is the `sprayd` service using?

It varies among different systems.

Which program number is the `sprayd` service using?

`100012`

4. Check the `/etc/services` file to determine if the `sprayd` service has been assigned a well-known port number.

```
# grep sprayd /etc/services
#
```

Has it?

No

5. Check the `/etc/rpc` file to see if the `sprayd` service is listed.

```
# grep sprayd /etc/rpc
sprayd          100012  spray
```

Is it listed?

Yes

6. Check that your system will respond to the `sprayd` service requests. Have your partner run the `spray` command, and specify your system as the target.

```
# spray host1
```

7. Use the `rpcinfo` command to unregister the `sprayd` service's port number. Check that the `sprayd` service is no longer listed as a registered port number.

```
# rpcinfo -d sprayd 1
# rpcinfo -p | grep sprayd
```

8. Have your partner run the `spray` command, and specify your system as the target again.

What message does the `spray` command return?

```
# spray host
spray: cannot clnt_create host:netpath: RPC: Program not registered
```

9. Send the `HUP` signal to the `inetd` daemon for the `rpcbind` service to register all services listed in its configuration file.

```
# pkill -HUP inetd
```

10. Verify that the `sprayd` service is listed as a registered service.

```
# rpcinfo -p | grep sprayd
    100012    1   udp  32841  sprayd
```

What port number is the `sprayd` service using now?

It varies among different systems.

Is the program number used by the `sprayd` service the same as the program number that was listed in Step 3?

`100012`

11. To check that the `sprayd` service can now contact your system, have your partner run the `spray` command, and specify your system as the target.

```
# spray host
```

12. Stop the `snoop` processes running on both systems.

Press Control-C.

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 3

Customizing the Solaris™ Management Console

Objectives

The Solaris™ Management Console uses a graphical user interface (GUI) to display management tools that are stored in containers referred to as toolboxes. The console includes a default toolbox containing tools for managing users, projects, and `cron` jobs. The toolbox also contains tools for mounting and sharing file systems and for managing disks and serial ports. The Solaris Management Console toolbox editor application, which looks similar to the console, can add and modify toolboxes, add tools to a toolbox, and extend the functionality of a toolbox to other applications.

Upon completion of this module, you should be able to:

- Describe the Solaris Management Console toolbox editor actions
- Use the Solaris Management Console toolbox editor

The following course map shows how this module fits into the current instructional goal.

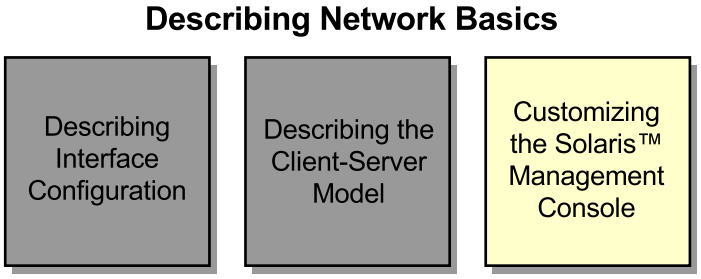


Figure 3-1 Course Map

Introducing the Solaris Management Console Toolbox Editor Actions

This section describes how to start the Solaris Management Console components and how to edit a toolbox to increase functionality with access to other Solaris Management Console servers or to legacy applications.

Starting the Solaris Management Console

The Solaris Management Console has three primary components:

- The Solaris Management Console server
- The console
- The Solaris Management Console toolbox editor

Starting the Solaris Management Console Server

If you have trouble starting the Solaris Management Console, it might be because the Solaris Management Console server is not running or because it is somehow in a problem state.

Note – Open a system console window to view Solaris Management Console load messages.

To determine if the Solaris Management Console server is running, as `root` perform the command:

```
# /etc/init.d/init.wbem status
```

If the Solaris Management Console server is running, you see a response similar to the following:

```
Solaris Management Console server version 2.1.0 running on port 898
```

To stop the Solaris Management Console server, perform the command:

```
# /etc/init.d/init.wbem stop
```

The following message appears:

```
Shutting down Solaris Management Console server on port 898.
```

To start the Solaris Management Console server, perform the command:

```
# /etc/init.d/init.wbem start
```

After a short time the following message appears:

```
Starting Solaris Management Console server version 2.1.0.
endpoint created: :898
Solaris Management Console server is ready.
```

Note – For more information, visit `http://www.dmtf.org`.

Starting the Console

You can start the console from the command line from the Tools menu of the CDE front panel, or by double-clicking a Solaris Management Console icon in the Applications Manager or in the File Manager.

To start the console from the command line, perform the command:

```
# smc
```

Note – You can start Solaris Management Console as a regular user, but some tools and applications might not load unless you log in to the Solaris Management Console server as `root`, or unless you assume a role-based access control (RBAC) role during Solaris Management Console server login.

Starting the Toolbox Editor

To start the Solaris Management Console toolbox editor, perform the command:

```
# smc edit
```

You can start the Solaris Management Console toolbox editor as a normal user, but you cannot save a server toolbox unless you log in as `root`.

Caution – In this module, you modify the contents of the Solaris Management Console's toolboxes. This module directs you to alter and save both the Management Tool (root) toolbox and the This Computer (default) toolbox. Before you modify either toolbox, create backups of both toolboxes using the following commands:

```
# cd /var/sadm/smc/toolboxes
# cp smc/smc.tbx smc.tbx.orig
# cp this_computer/this_computer.tbx this_computer.tbx.orig
```

Introducing the Solaris Management Console and the Solaris Management Console Toolbox Editor

The Solaris Management Console contains a hierarchical collection of folders, tools, legacy applications, and links to other toolboxes. A toolbox can include links to other toolboxes, individual tools, folders, and legacy applications.

- A Solaris Management Console toolbox is a collection of tools that have been registered using the `smcregister` utility.
- The root toolbox, or container, is called Management Tools. The default behavior of the Management Tools is to look for a toolbox on the local host and link to it when the Solaris Management Console starts. You can add multiple toolboxes to Management Tools.
- A toolbox Universal Resource Locator (URL), or link, is a pointer to another toolbox that might be on the current Solaris Management Console server or on any other Solaris Management Console server.
- A tool is an application or applet that is compatible with the Solaris Management Console that integrates easily into the Console. A Solaris Management Console tool is built using the Solaris Management Console software development kit (SDK).
- A folder is a container that groups tools within a toolbox.
- A legacy application is an application that is not a Solaris Management Console tool. A legacy application can be a command, X Application, or a URL.

The root toolbox URL is:

`http://`*`hostname`*`:898/toolboxes/smc.tbx`

The default location for this toolbox on the system is:

`/var/sadm/smc/toolboxes/smc/smc.tbx`

The root toolbox is loaded by default when either the `smc` or `smc edit` commands are run on a server. This toolbox only allows access to other toolboxes, not to the tools within those toolboxes. You access the actual default tools through the URL:

`http://`*`hostname`*`:898/toolboxes/this_computer.tbx`

The default location for this toolbox on the system is:

`/var/sadm/smc/toolboxes/this_computer/this_computer.tbx`

Introducing the Solaris Management Console

To start the Solaris Management Console, perform the command:

```
# smc &
```

The Solaris Management Console window appears, as shown in Figure 3-2.

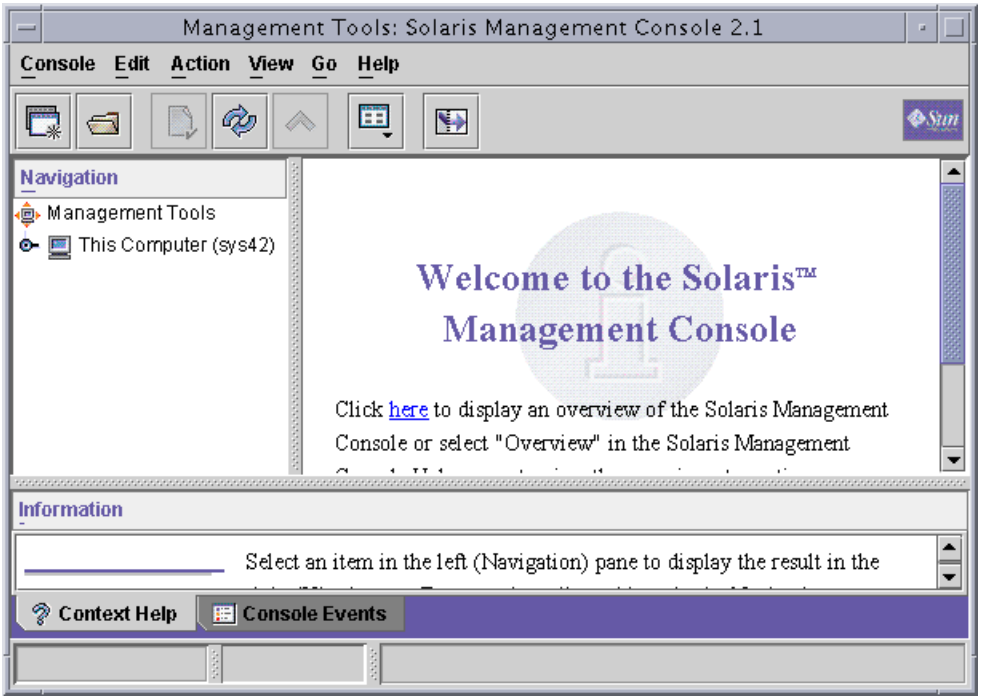


Figure 3-2 Solaris Management Console 2.1 Window

When you select a toolbox in the Navigation pane, as shown in Figure 3-3, the set of tools in that toolbox are displayed in the View pane. You can double-click a tool in the View pane to open the next layer within the toolbox hierarchy.

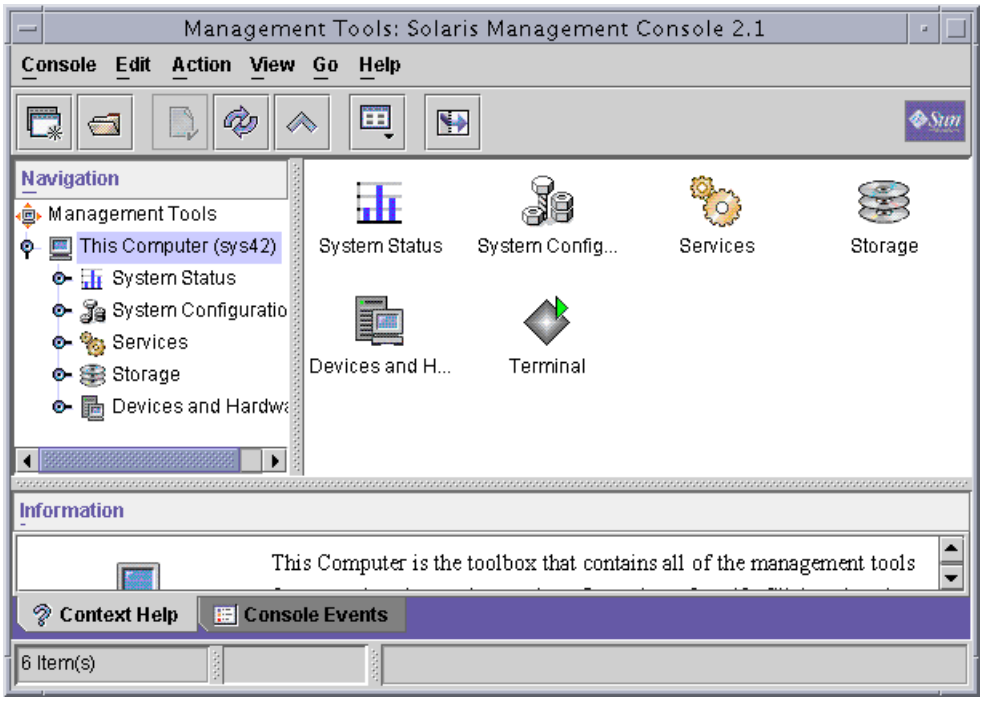


Figure 3-3 This Computer Toolbox Window

A toolbox allows for the grouping of tools into a consistent, user-friendly hierarchy. The default toolbox for a Solaris Management Console server is called This Computer. Table 3-1 describes the categories (or folders) and tools included in the default toolbox.

Table 3-1 Solaris Management Console Categories

Category	**Includes**
System Status	Processes, Log Viewer, System Information, and Performance
System Configuration	Users, Projects, Computers and Networks, and Patches
Services	Scheduled Jobs
Storage	Disks, Mounts and Shares, and Enhanced Storage Tool
Devices and Hardware	Serial Ports
Terminal	Terminal is not a category. Clicking the Terminal icon launches a terminal window.

Double-click a specific folder to view the contents of that folder category. The tools that are stored within the folder are displayed in the View pane, as shown in Figure 3-4.

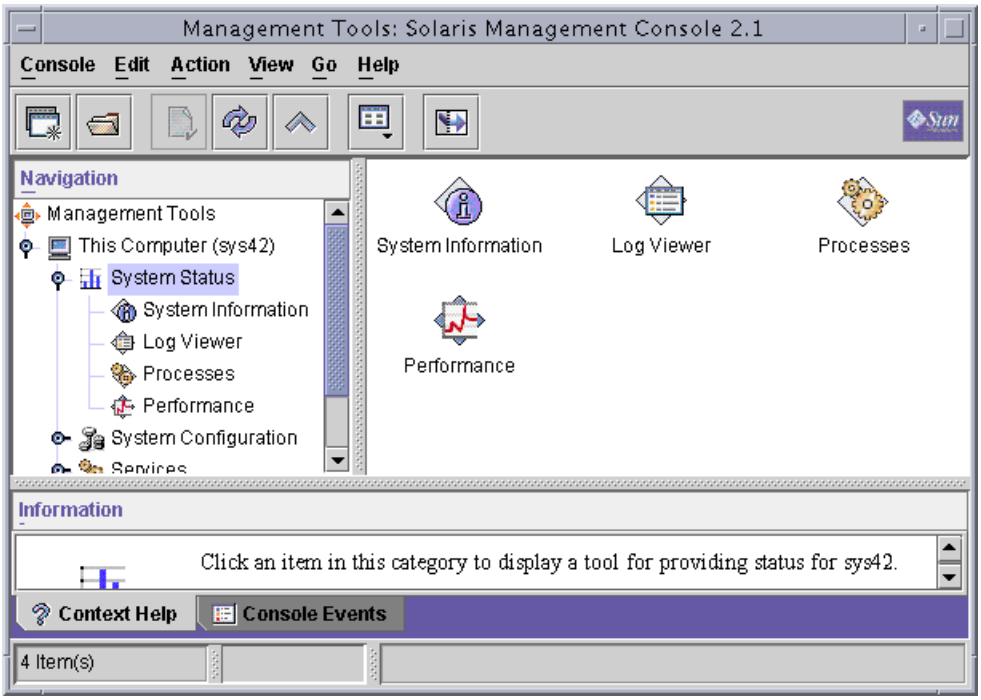


Figure 3-4 System Status Window

Double-click on a specific tool to launch that tool.

The View pane in Figure 3-5 displays the tool-specific information.

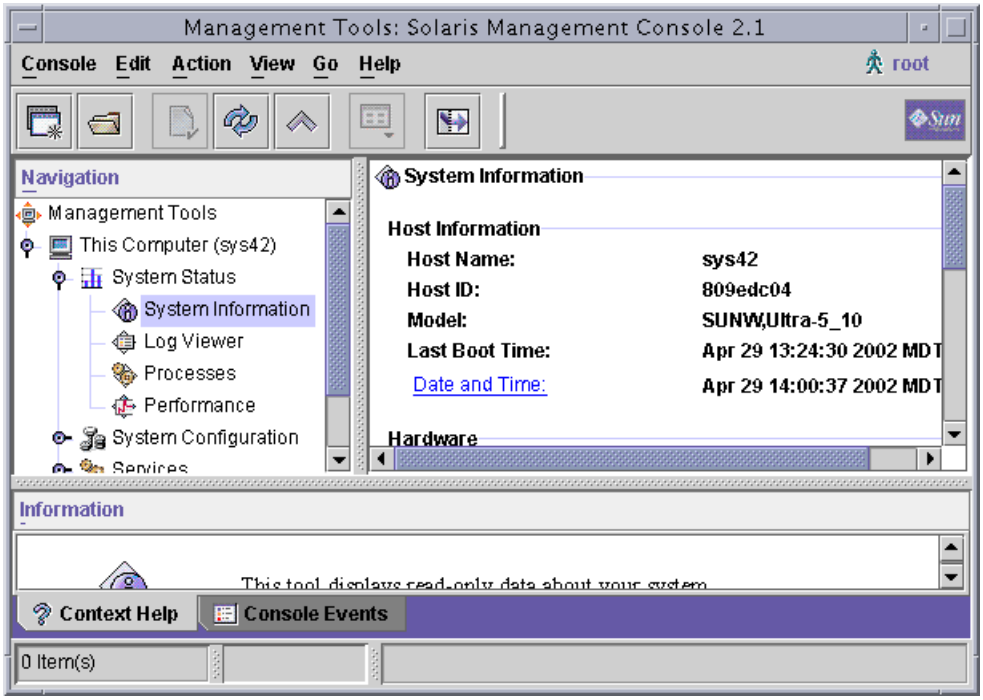


Figure 3-5 System Information Window

The System Information window, shown in Figure 3-5, collects and displays system configuration information.

Introducing the Solaris Management Console Toolbox Editor

To start the Solaris Management Console toolbox editor, perform the command:

```
# smc edit &
```

You use the Solaris Management Console Editor 2.1 Window to execute tools during daily administrative activities (Figure 3-6). You also use the Solaris Management Console toolbox editor to modify existing toolboxes or to create additional toolboxes. You can use these toolboxes to manage multiple servers from one toolbox or to group similar tools in a toolbox.

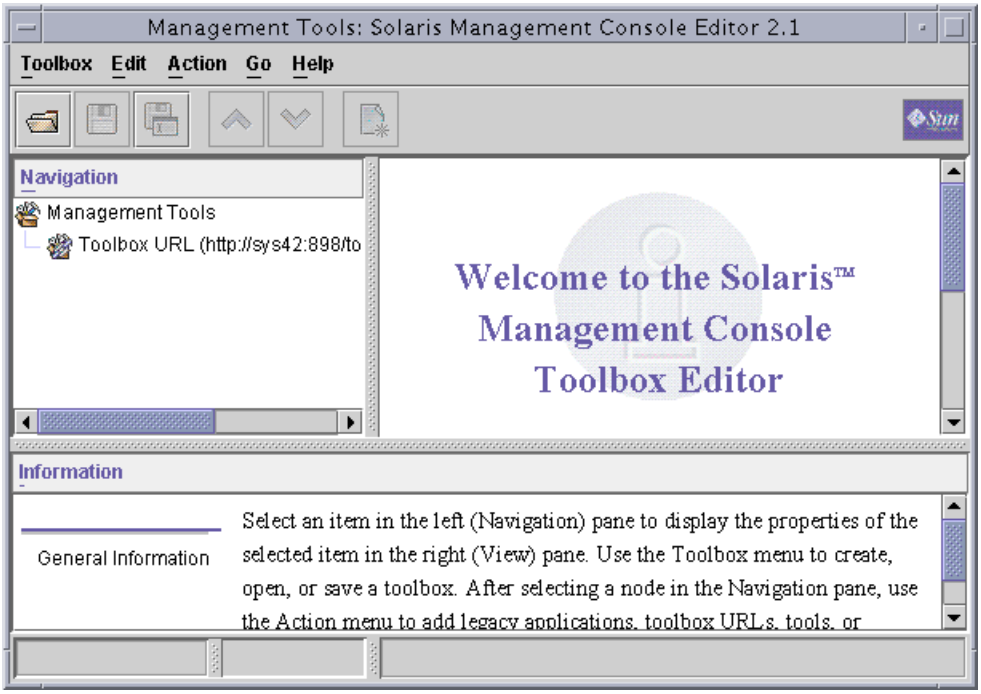


Figure 3-6 Solaris Management Console Editor 2.1 Window

Select an item in the Navigation pane, as shown in Figure 3-7 on page 3-11, to display the properties of the selected item in the View pane.

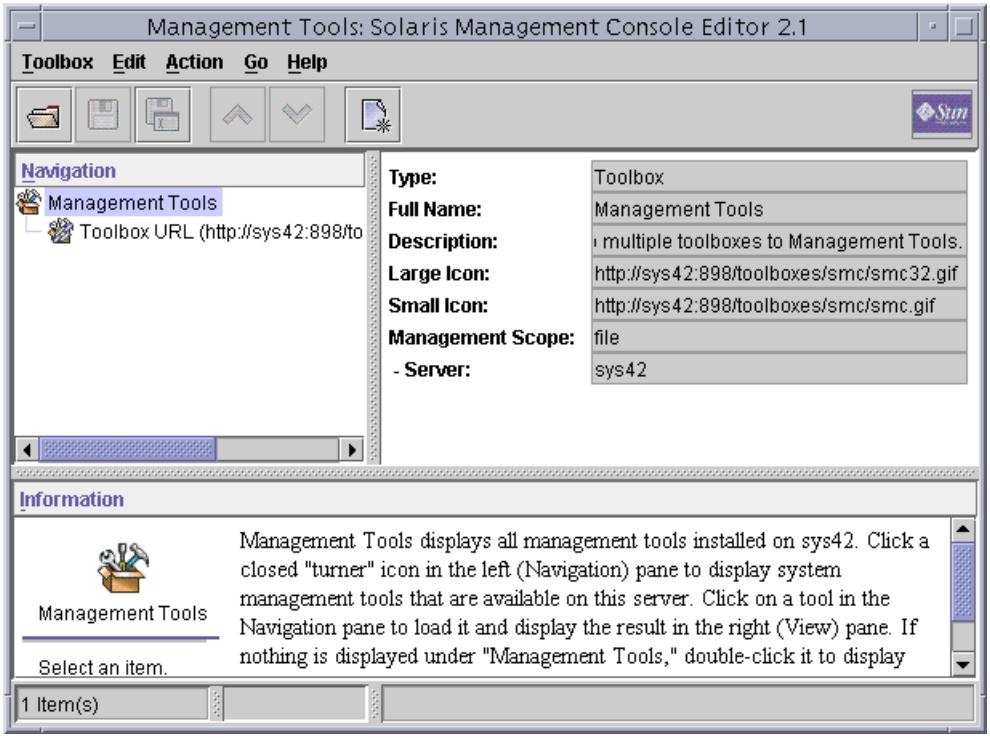


Figure 3-7 Management Tools Statistics

When displaying the root toolbox in the Solaris Management Console toolbox editor, as shown in, you can only see the server toolboxes that are linked to that root toolbox. You can use the contents of a toolbox by opening it in the Solaris Management Console. After creating or modifying any toolbox, you must save the toolbox changes and reopen the toolbox in the Solaris Management Console before you can access new tools.

Menu Bar

The menu bar is at the top of the toolbox editor and includes the following menus:

- Toolbox
- Edit
- Action
- Go
- Help

By default, the Toolbox menu, as shown in Figure 3-8, includes the following items:

New	Creates a new toolbox
Open	Opens an existing toolbox in the current console window
Save	Saves the current toolbox
Save As	Saves the current toolbox configuration after you rename the toolbox location
Exit	Exits from the toolbox editor

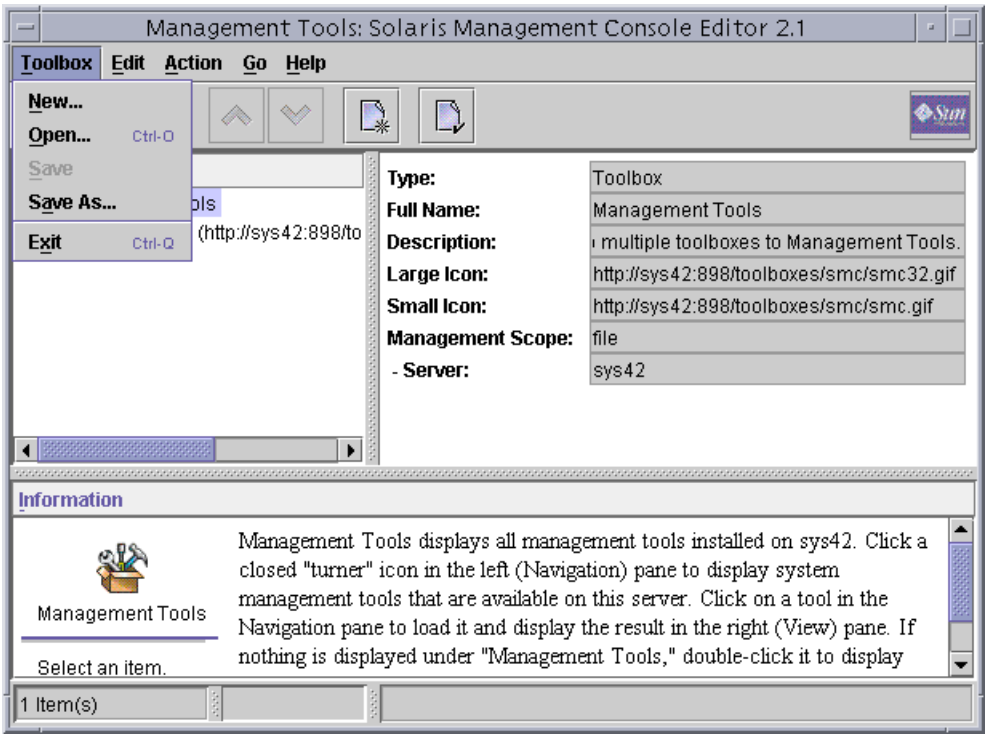


Figure 3-8 Solaris Management Console Editor 2.1 Window – Toolbox Menu

By default, the Edit menu, as shown in Figure 3-9, includes only the following item:

Delete	Deletes the objects that are selected in the Navigation pane

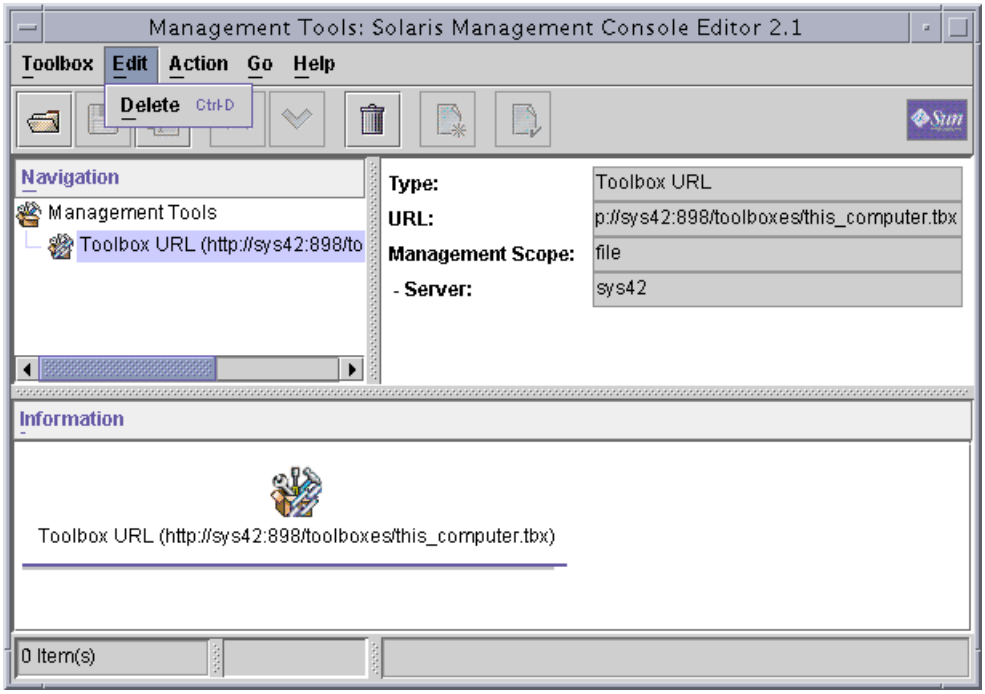


Figure 3-9 Solaris Management Console Editor 2.1 Window – Edit Menu

By default, the Action menu, as shown in Figure 3-10, includes the following items:

Add Legacy Application	Adds a legacy application that is not a Solaris Management Console tool. It could be a command-line interface, an X application, or a URL.
Add Toolbox URL	Adds a link from an existing toolbox to another toolbox, possibly on another server.
Add Tool	Adds a tool to an existing toolbox.
Add Folder	Adds a folder to an existing toolbox.
Move Up	Moves the selected item in the Navigate pane up in the hierarchy.
Move Down	Moves the selected item in the Navigate pane down in the hierarchy.
Properties	Displays the assigned characteristics for the selected tool or toolbox.

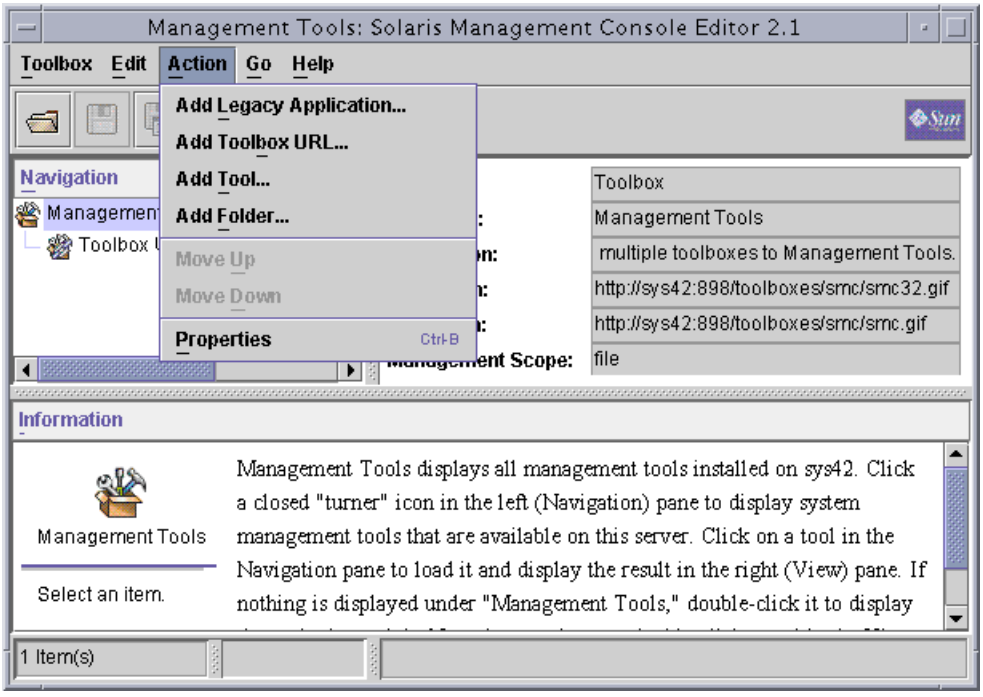


Figure 3-10 Solaris Management Console Editor 2.1 Window – Action Menu

The Go menu, as shown in Figure 3-11, includes the following items:

Up Level	Moves up one level in the toolbox hierarchy, and displays the result in the Navigation and View panes
Home Toolbox	Opens your home toolbox, as defined in the Console tab of the Preferences dialog box

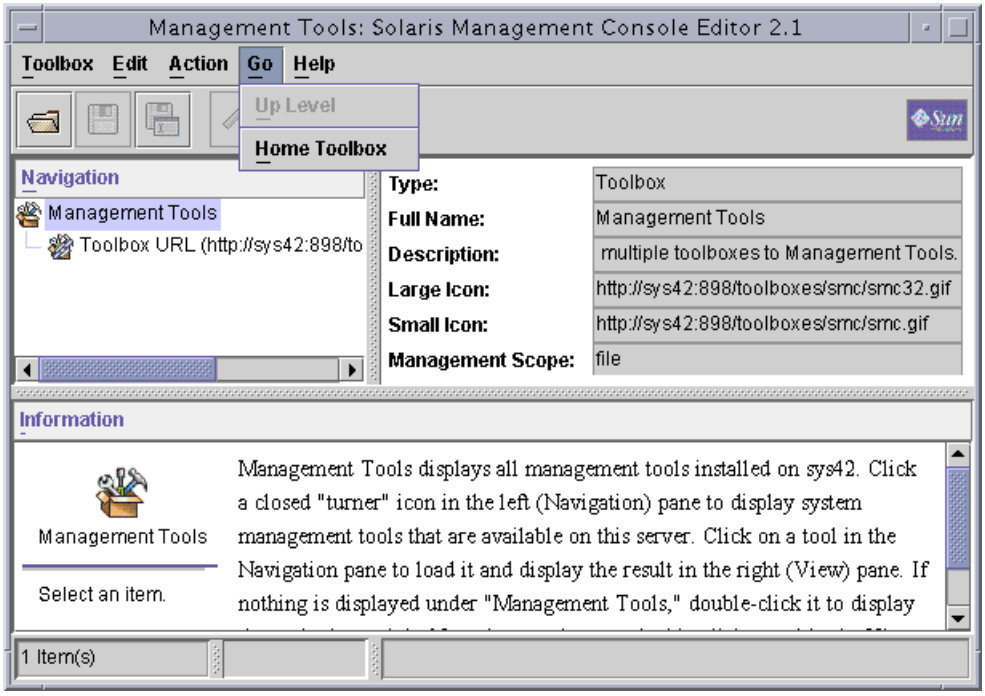


Figure 3-11 Solaris Management Console Editor Window 2.1 – Go Menu

By default, the Help menu, as shown in Figure 3-12, includes the following items:

Overview	Displays the help viewer with an Overview in the topic pane. The Overview function also provides a general description of the Solaris Management Console.
Contents	Displays the help viewer with table of contents in the Navigation pane.
Index	Displays the help viewer with an index in the Navigation pane.
Search	Displays the help viewer with a Find function in the Navigation pane.
About Console	Displays the version number of Solaris Management Console, copyright, and trademark information.

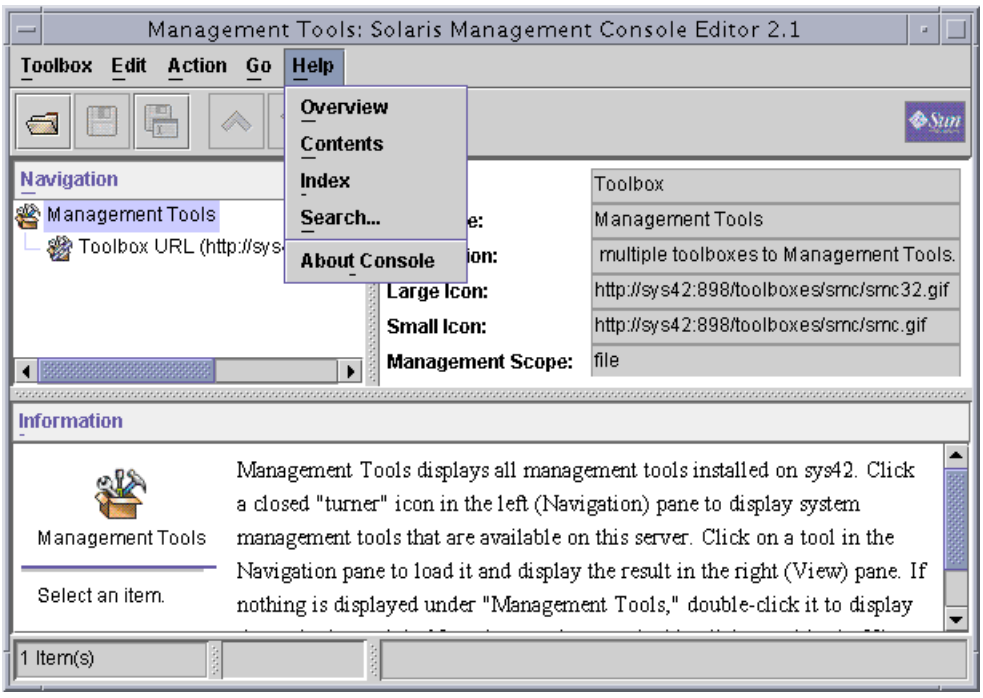


Figure 3-12 Solaris Management Console Editor 2.1 Window – Help Menu

Adding a Toolbox URL

You can add access to the Toolbox URL from one Solaris Management Console server to another Solaris Management Console server. This function provides a mechanism for centralizing control across multiple Solaris Management Console servers.

To add access to a Solaris Management Console server toolbox from other Solaris Management Console servers, follow these steps:

1. Open the toolbox to which you want to add the toolbox URL.
2. Select the node in the toolbox to which you want to add the toolbox URL.
3. Select the Add a Toolbox URL from the Action menu.
4. Follow the instructions in the Add Toolbox URL wizard.
5. Save the toolbox.

The new toolbox contents must be reloaded in the Solaris Management Console before the changes become visible.

Adding a Tool

Adding access to a specific Solaris Management Console server tool from other Solaris Management Console servers enables you to configure many different support scenarios using the Solaris Management Console toolboxes. In a single toolbox, you can configure all tools from a number of servers for a particular functionality. This access provides the capability to configure a single Solaris Management Console server for access, such as a storage server, across all the Solaris Management Console servers.

To add access to a specific Solaris Management Console server tool from other Solaris Management Console servers:

1. Open the toolbox to which you want to add the tool.
2. Select the node in the toolbox to which you want to add the tool.
3. Select Add Tool from the Action menu.
4. Follow the instructions in the Add Tool wizard.
5. Save the toolbox.

The new toolbox contents must be reloaded in the Solaris Management Console before the changes become visible.

Using the Solaris Management Console Toolbox Editor

You use the Solaris Management Console toolbox editor functions to:

- Provide visibility between the Solaris Management Console server root toolbox and the default toolbox of additional Solaris Management Console servers
- Provide visibility of specific Solaris Management Applications between the Solaris Management Console servers
- Create additional container mechanisms within the Solaris Management Console server
- Provide access to legacy applications from within the Solaris Management Console server

Adding Access to a Toolbox URL of a Solaris Management Console

This section describes how to access the toolbox URL of a Solaris Management Console server named `sys44` from a Solaris Management Console server named `sys42`. You will access the toolbox URL by customizing the configuration of the server on `sys42` with a pointer that points to the `sys44` server's URL. This procedure involves:

- Opening the toolbox
- Adding the toolbox URL
- Saving the toolbox

Opening the Toolbox

To open the toolbox, select the Management Tools (root) toolbox, as shown in Figure 3-13.

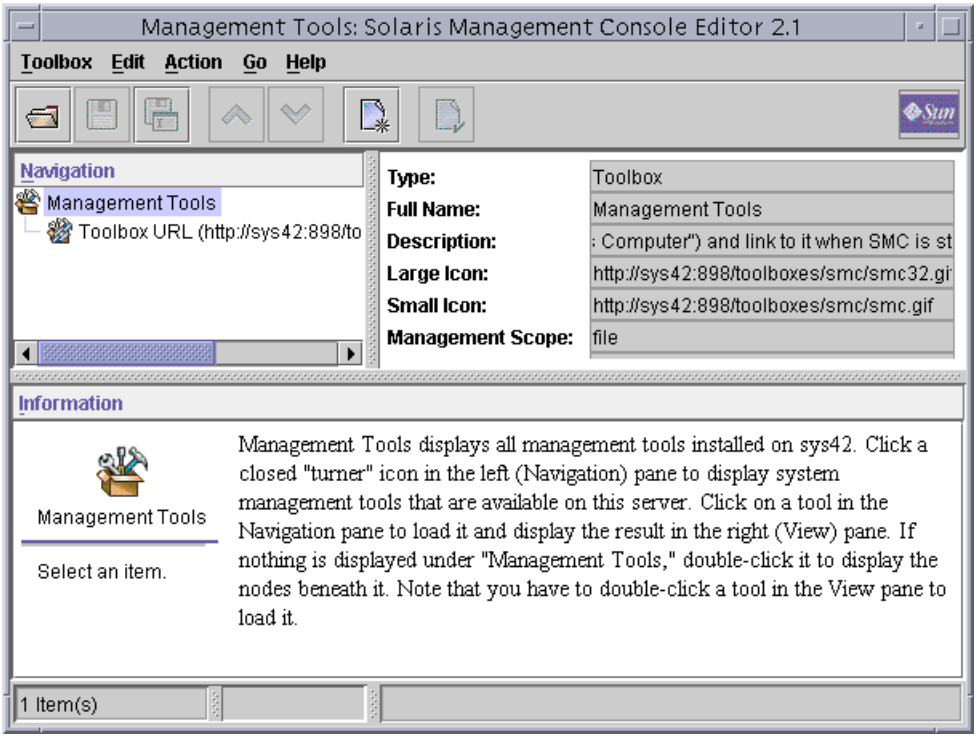


Figure 3-13 Management Tools Statistics

The system default toolbox URL (This Computer) will eventually become a component of the local root toolbox (Management Tools).

Adding a Toolbox URL

To add a toolbox URL, complete the following steps:

1. Select Add Toolbox URL from the Action menu, as shown in Figure 3-14, and follow the steps in the Toolbox URL Wizard.

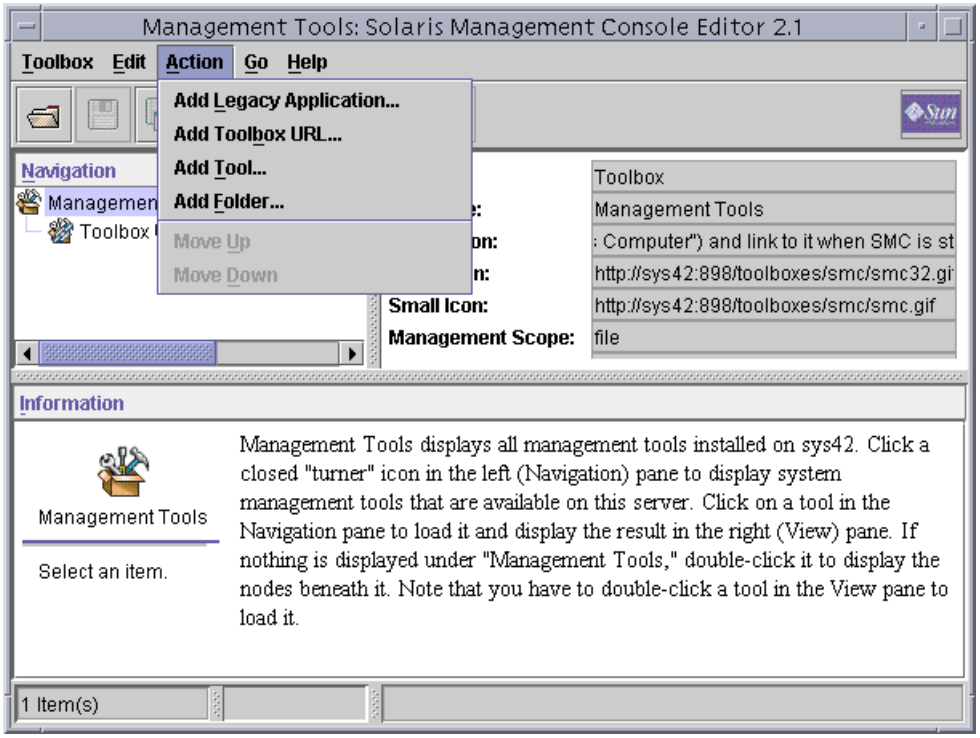


Figure 3-14 Action Menu – Add Toolbox URL

Note – These steps follow the prompts from the Toolbox URL Wizard.

The wizard displays a help screen along the left side of each window, as shown in Figure 3-15.

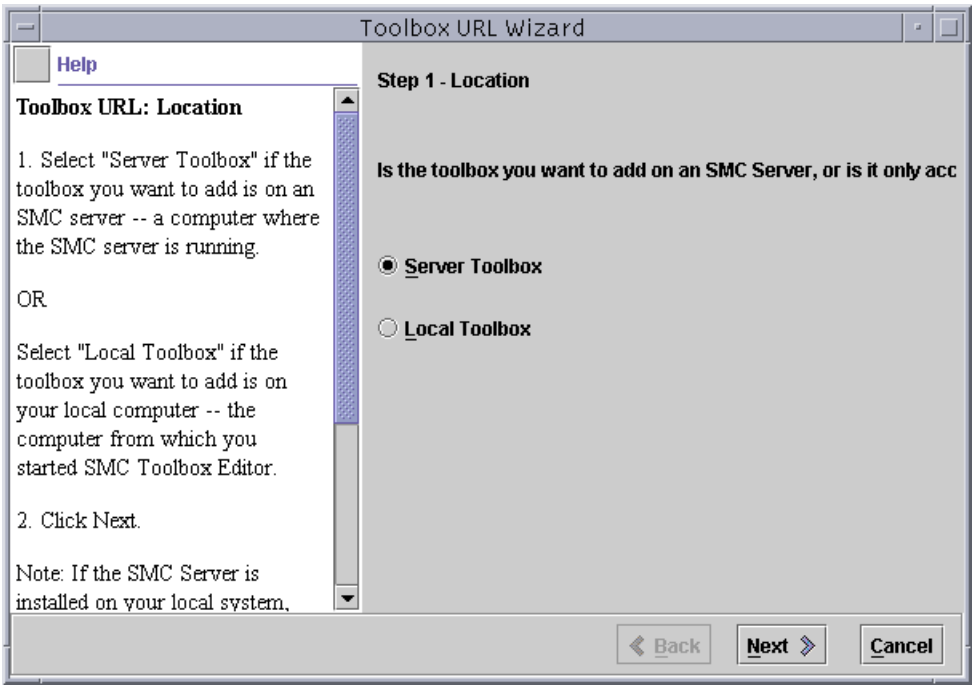


Figure 3-15 Toolbox URL Wizard – Step 1 Window

Note – To hide the help information, which expands the usable area within the wizard windows, click the gray box next to the word Help.

2. Click Next to continue.

In the Toolbox URL Wizard – Step 1 window, you either:

- Select Server Toolbox if the toolbox you want to add is on a Solaris Management Console server, which is the computer where the Solaris Management Console server is running.
- Select Local Toolbox if the toolbox you want to add is on your local computer, which is the computer from which you started the Solaris Management Console toolbox editor.

3. In this example, select Server Toolbox, as shown in Figure 3-16.

Figure 3-16 Toolbox URL Wizard – Step 1 Window

4. Click Next to continue.

In the Toolbox URL Wizard – Step 2 window, you enter the name and an optional port number of the remote Solaris Management Console server from which to retrieve the toolbox.

5. In this example, enter `sys44`, as shown in Figure 3-17.

Figure 3-17 Toolbox URL Wizard – Step 2 Window

6. Click Next to continue.

If the Solaris Management Console server is running and if any toolboxes are accessible on the server, a list of toolboxes appears in the Toolboxes field, as shown in Figure 3-18.

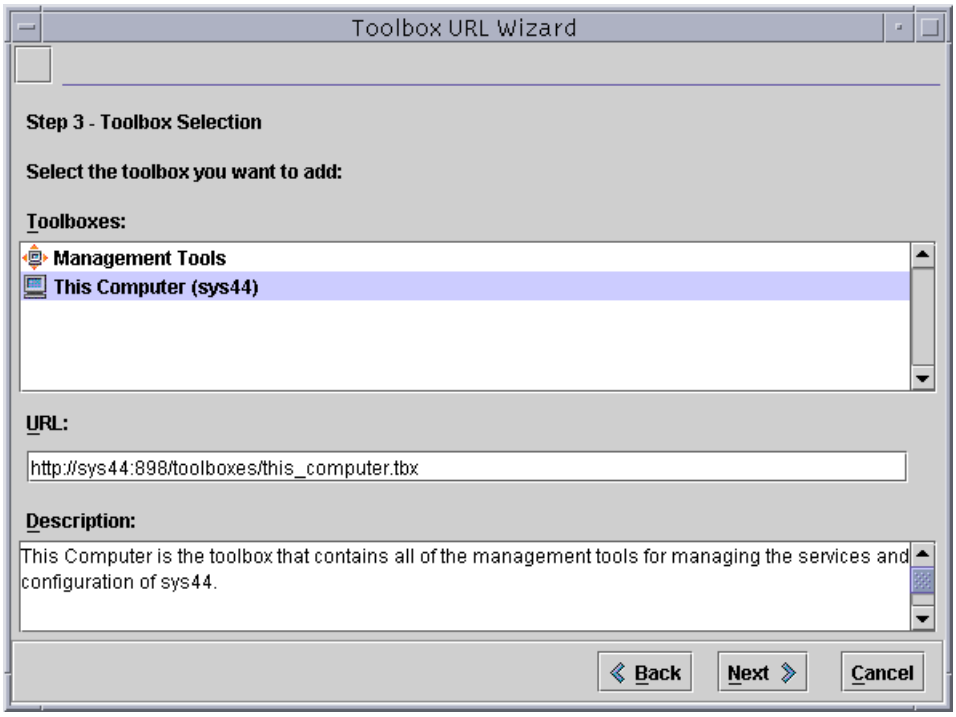


Figure 3-18 Toolbox URL Wizard – Step 3 Window

7. Select the This Computer (default) toolbox from the Toolboxes list.
8. Click Next to continue.

In the Toolbox URL Wizard – Step 4 window, you either:

- Select Use Toolbox Defaults to use the name and description specified in the toolbox definition.
- Select Override Toolbox Settings to override the name and description specified in the toolbox definition.

9. In this example, use the toolbox defaults, as shown in Figure 3-19.

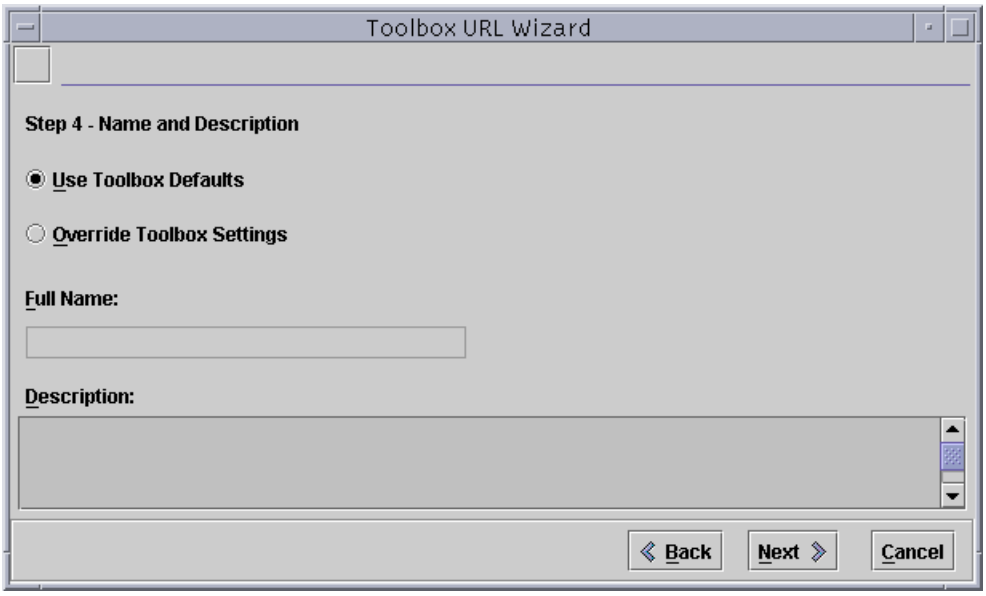


Figure 3-19 Toolbox URL Wizard – Step 4 Window

10. Click Next to continue.

In the Toolbox URL Wizard – Step 5 window, you either:

- Select Use Toolbox Defaults to use the existing toolbox icon.
- Select Override Toolbox Settings to select other toolbox icons, and then enter the full paths to the large and small icons.

11. In this example, use the toolbox defaults, as shown in Figure 3-20.

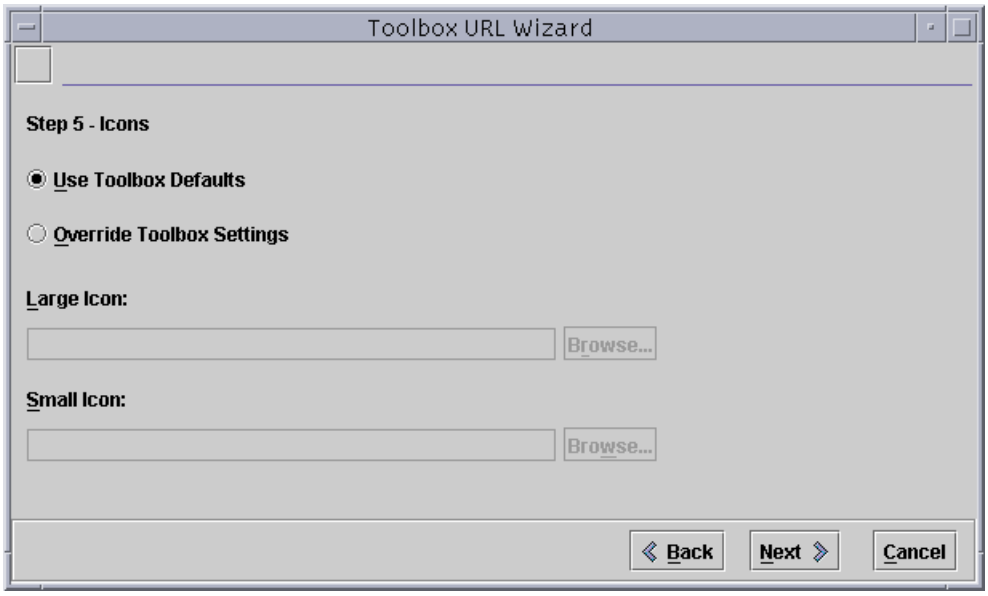


Figure 3-20 Toolbox URL Wizard – Step 5 Window

12. Click Next to continue.

Note – Management scope defines what the tool's action will update. For example, a tool can update local files on a server or a tool can update information in an NIS database. You can configure a toolbox folder and a specific tool with a scope of operation. You can create folders and tools that inherit the scope of operation from their parents, or you can configure them to override their parents' scope of operation.

In the Toolbox URL Wizard – Step 6 window, you either:

- Select Inherit from Parent to specify that the toolbox inherits its management scope from the parent node.
- Select Override to override the management scope of the parent node.

13. In this example, click Override, select the file management scope from the Management Scope pull-down menu, and then type the name of the server where the file or name service resides (sys44), as shown in Figure 3-21.

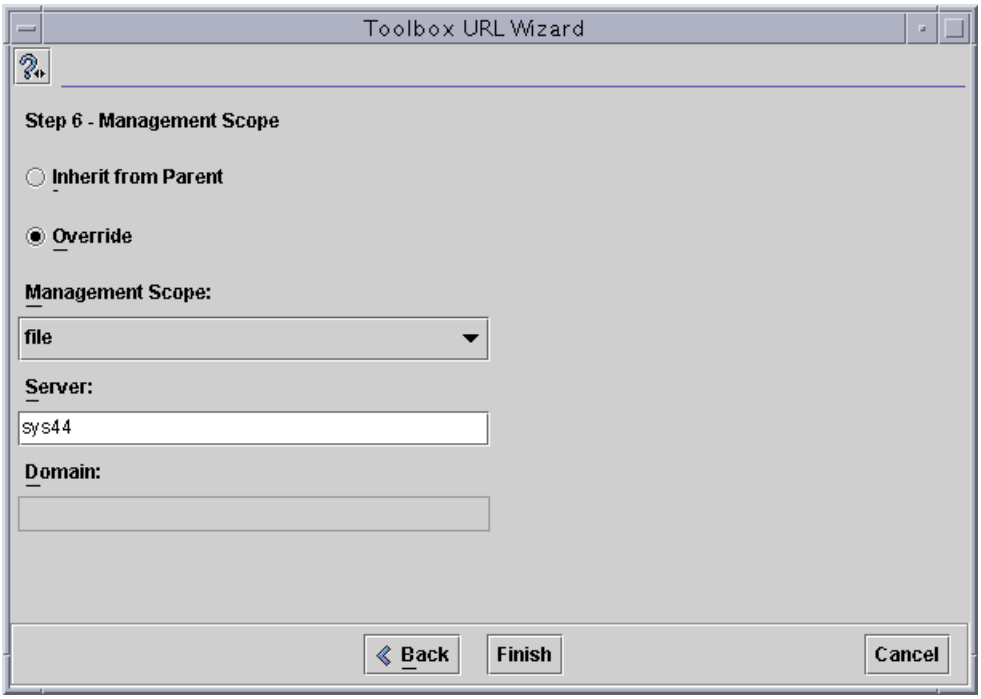


Figure 3-21 Toolbox URL Wizard – Step 6 Window

14. Click Finish.

The Add Toolbox URL wizard updates the selected toolbox with the additional toolbox URL, and returns you to the Solaris Management Console toolbox editor window, as shown in Figure 3-22.

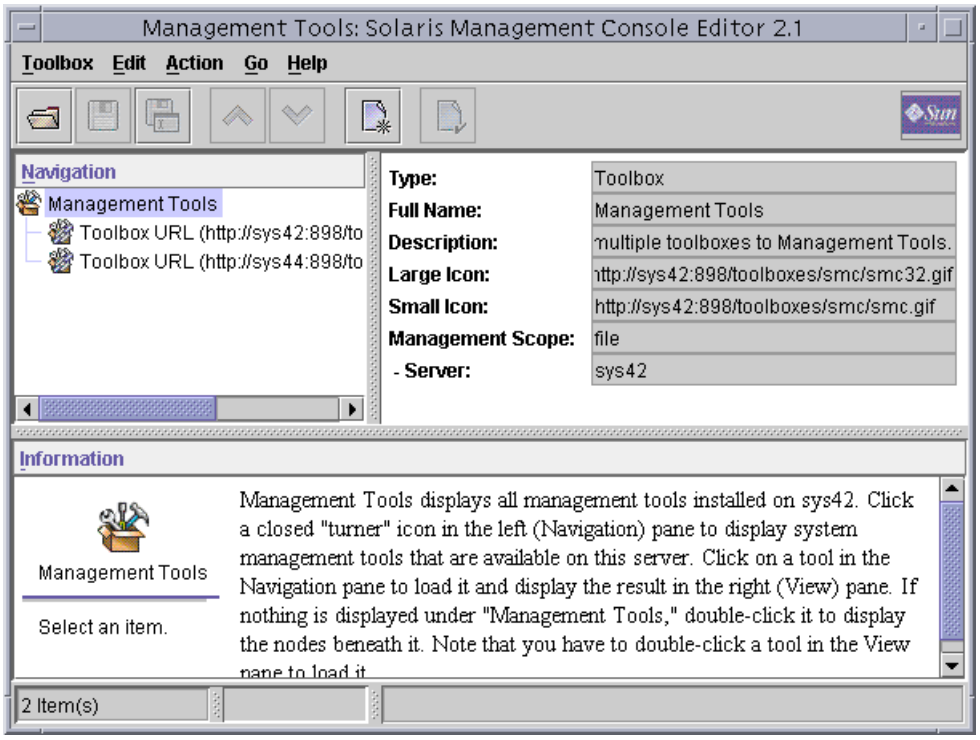


Figure 3-22 Solaris Management Console Editor 2.1 Window – Management Tools

15. To view the toolbox properties, select the new toolbox URL (sys42) in the Navigation pane.

Properties appear in View pane, as shown in Figure 3-23.

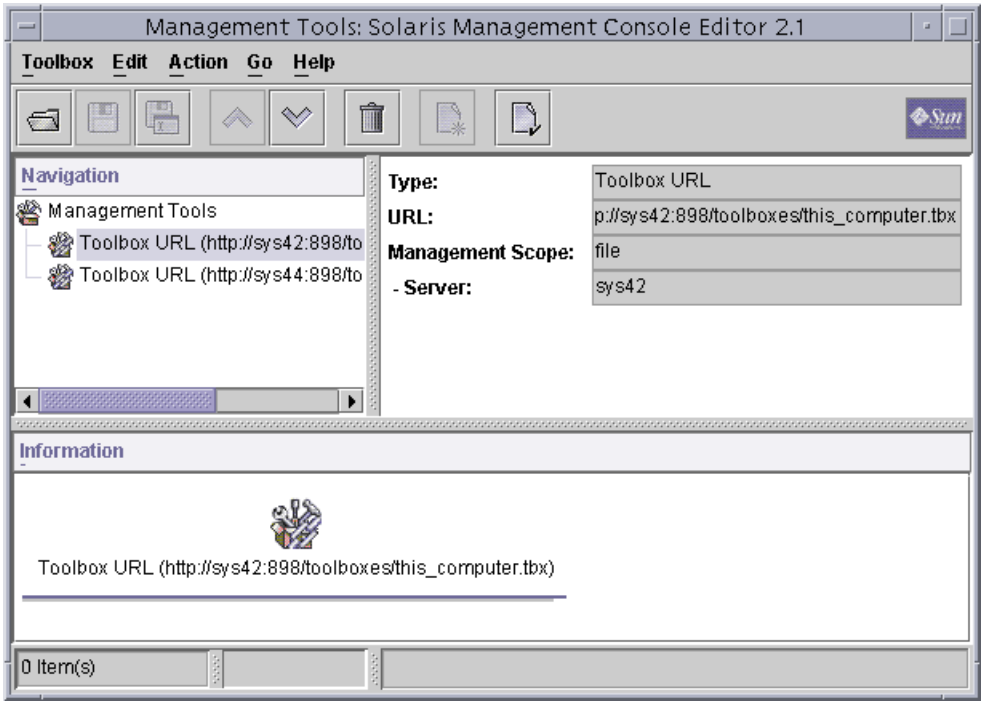


Figure 3-23 Toolbox URL Window

In the Toolbox URL window, you can:

- View the toolbox properties by selecting the toolbox URL in the Navigation pane and reading the contents in the View pane. In this example, sys42 is selected in the Navigation pane. Observe that the management scope is local files on server sys42, as shown in Figure 3-23 on page 3-30.
- Also view the other toolbox properties by selecting the new toolbox URL (sys44) in the Navigation pane and reading the view pane as shown in Figure 3-24.

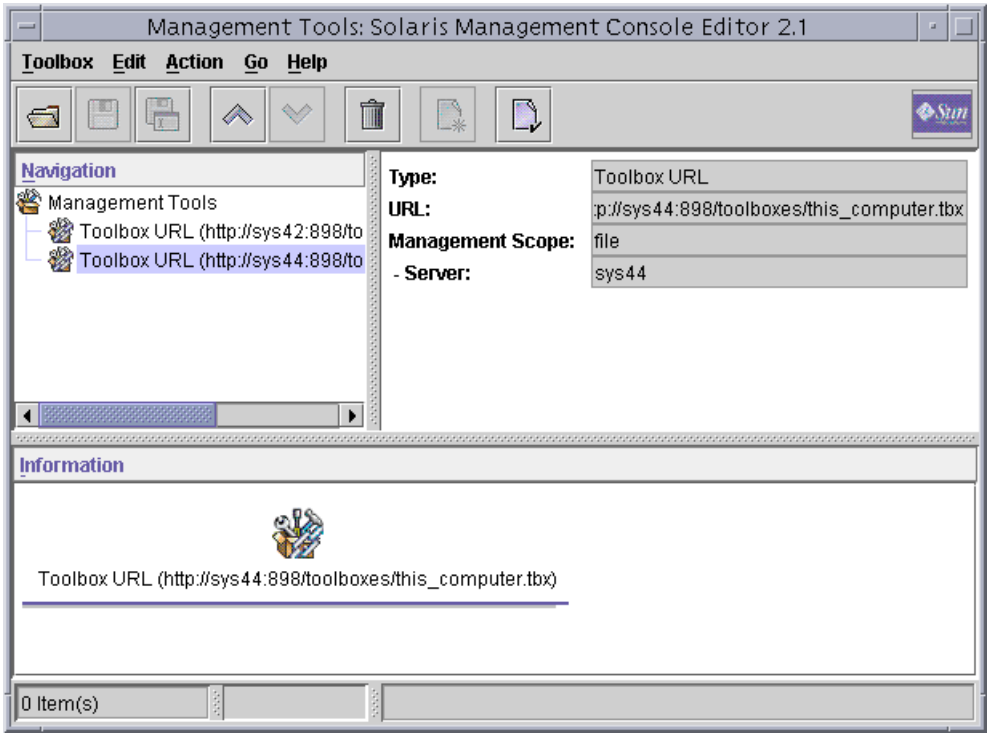


Figure 3-24 Toolbox URL Window

In this example, the management scope defines the use of local files on the system sys44.

Saving a Toolbox

Every time you make a change to a toolbox, save the changes to that toolbox by using the Solaris Management Console toolbox editor, and then reload that toolbox by using the Solaris Management Console.

To save and reload the toolbox, perform the following steps:

1. To ensure that you are saving the correct toolbox, select the toolbox that you want to save. In this example, select the Management Tools item in the Navigation pane, as shown in Figure 3-25.

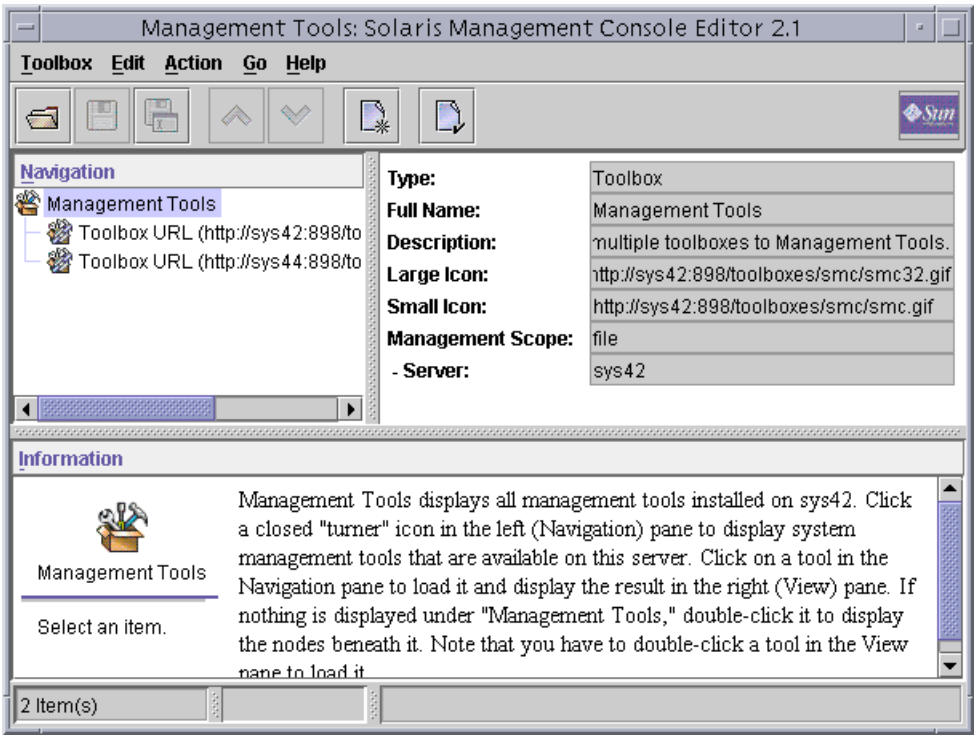


Figure 3-25 Management Tools Window

2. To save the selected toolbox, select Save As from the Toolbox menu, as shown in Figure 3-26.

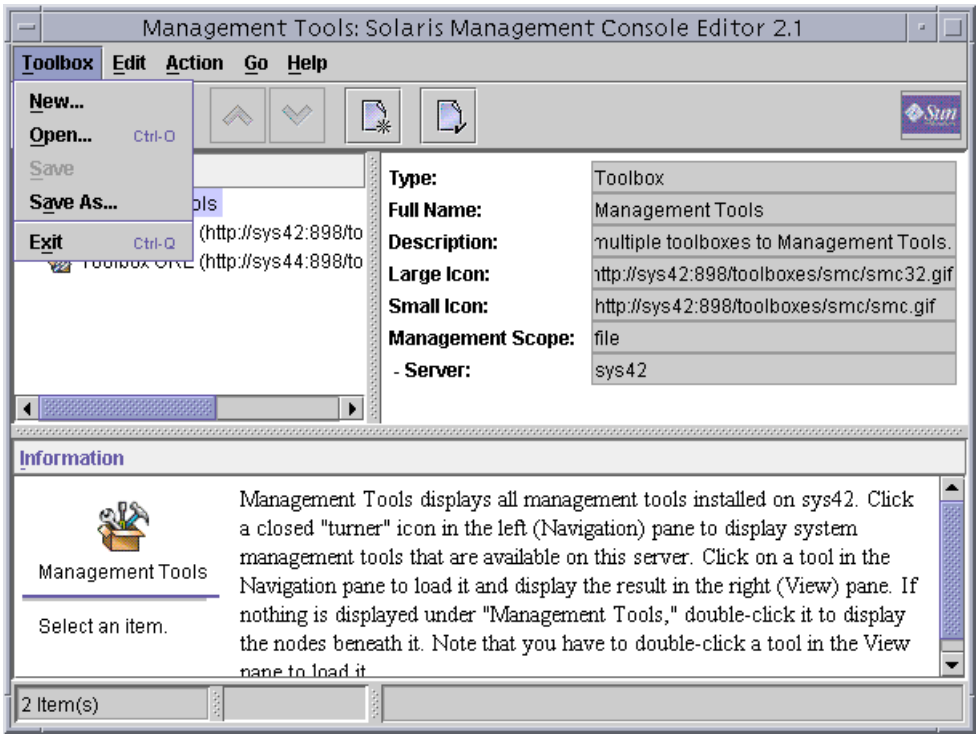


Figure 3-26 Toolbox Menu – Save As

3. In the Local Toolbox window (Figure 3-27), perform one of the following:

   - Select a toolbox from the list.
   - Navigate to a different toolbox using the appropriate folder icon.
   - Specify the root toolbox location by entering the absolute path to the toolbox into the Filename box.

     The absolute path name to the root toolbox is:
     `/var/sadm/smc/toolboxes/smc/smc.tbx`

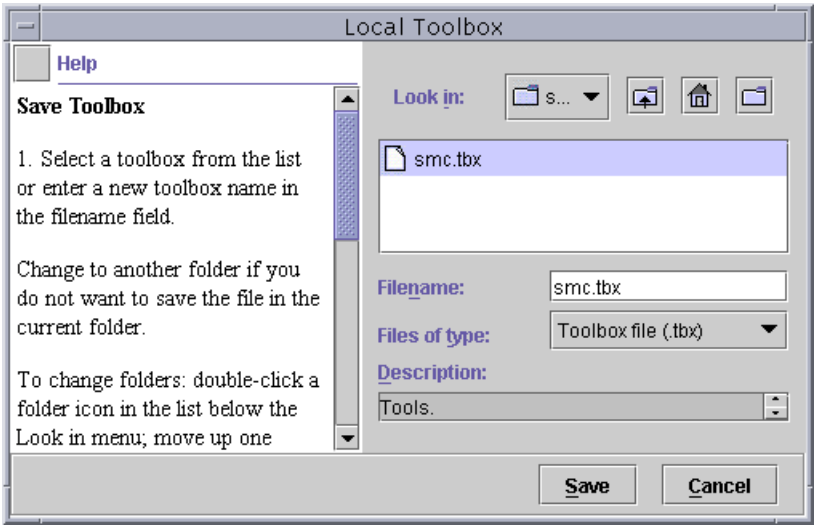


Figure 3-27 Local Toolbox Window

4. Click Save.

After you save the toolbox, you are returned to the Solaris Management Console toolbox editor window, as shown in Figure 3-28.

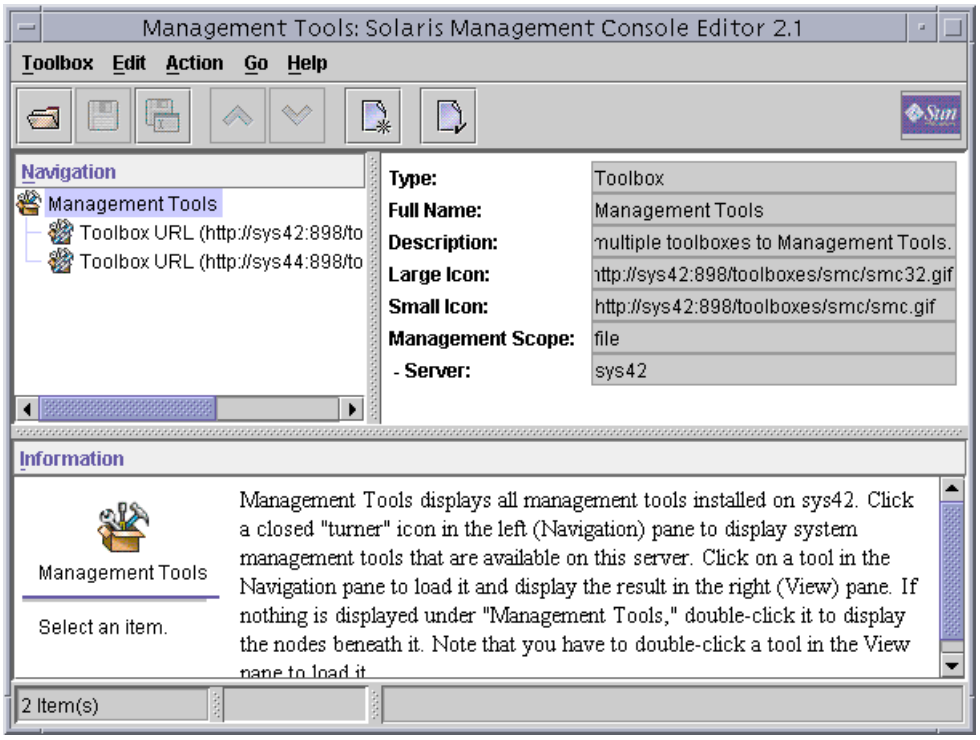


Figure 3-28 Solaris Management Console Editor 2.1 Window – Toolbox Saved

Adding a Toolbox URL Using the Command Line

The `smcregister` command configures the Solaris Management Console. This command enables you to add to, remove from, and list the contents of toolboxes.

Note – Using the `smcregister` command to edit toolboxes does not provide as many features as using the Solaris Management Console toolbox editor's graphical interface. Use the command-line interface in packaging scripts that do not require user interaction. However, to edit *all* the properties of a toolbox or to modify the hierarchy of folders in a toolbox, use the specialized graphical editor that is invoked with the `smc edit` command.

The `smcregister` command replaces the `smcconf` command for managing the Solaris Management Console repository and toolboxes.

You can add a toolbox URL using the `smcregister` command, as follows:

```
# /usr/sadm/bin/smcregister toolbox add tbxURL \
http://sys43:898/toolboxes/this_computer.tbx \
-B /var/sadm/smc/toolboxes/smc/smc.tbx
```

The previous example adds access to the default toolbox of system `sys43` (`http://sys43:898/toolboxes/this_computer.tbx`) from the root toolbox of the local system (`/var/sadm/smc/toolboxes/smc/smc.tbx`).

Adding Access to a Tool

You can configure a tool so that other Solaris Management Console servers can access it. To add access to a tool, you must provide the information needed to clearly identify the location and function of that tool. This procedure involves:

- Opening the toolbox
- Adding a tool
- Saving the toolbox

Opening the Toolbox

Prior to adding a tool to a Solaris Management Console server, you must be certain that you have opened the toolbox in which you want the tool to reside

1. Open the toolbox in which you want the tool to reside.

   The Solaris Management Console toolbox editor window displays the available toolbox structure contained within the root toolbox. These toolboxes include the root toolbox with its default toolbox and any additional toolboxes that have been added using the Add Toolbox URL function.

   If you want to add a tool to a default toolbox (`this_computer.tbx`) rather than adding a tool to a root toolbox (`smc.tbx`), you must first load the default toolbox.

2. Select Open from the Toolbox menu, as shown in Figure 3-29.

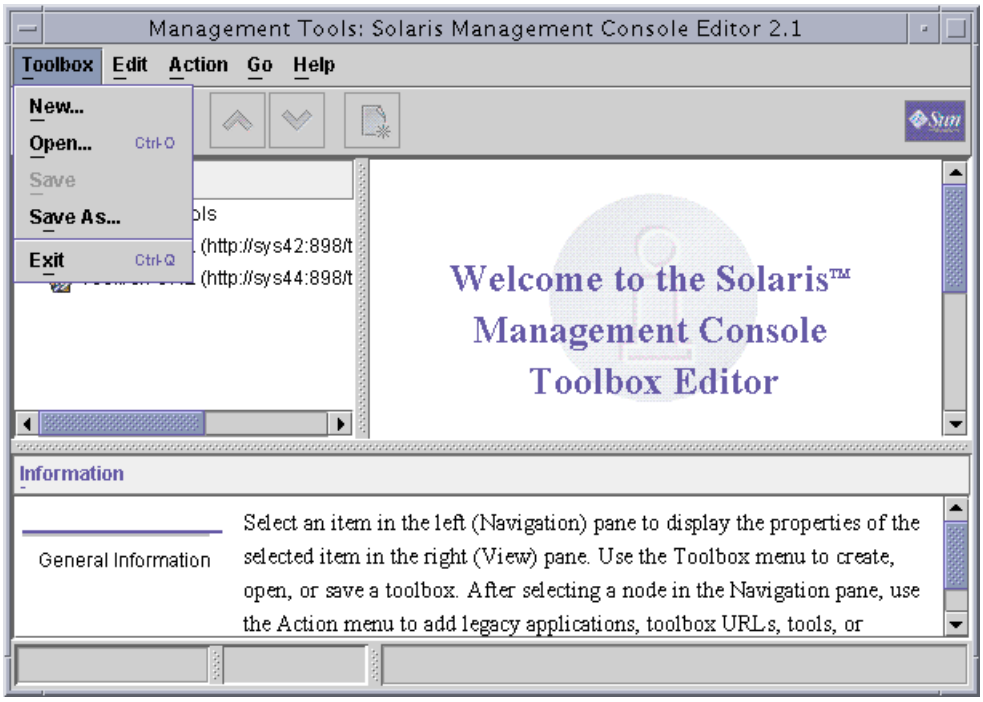


Figure 3-29 Toolbox Menu – Open

The default toolbox is listed, as shown in Figure 3-30.

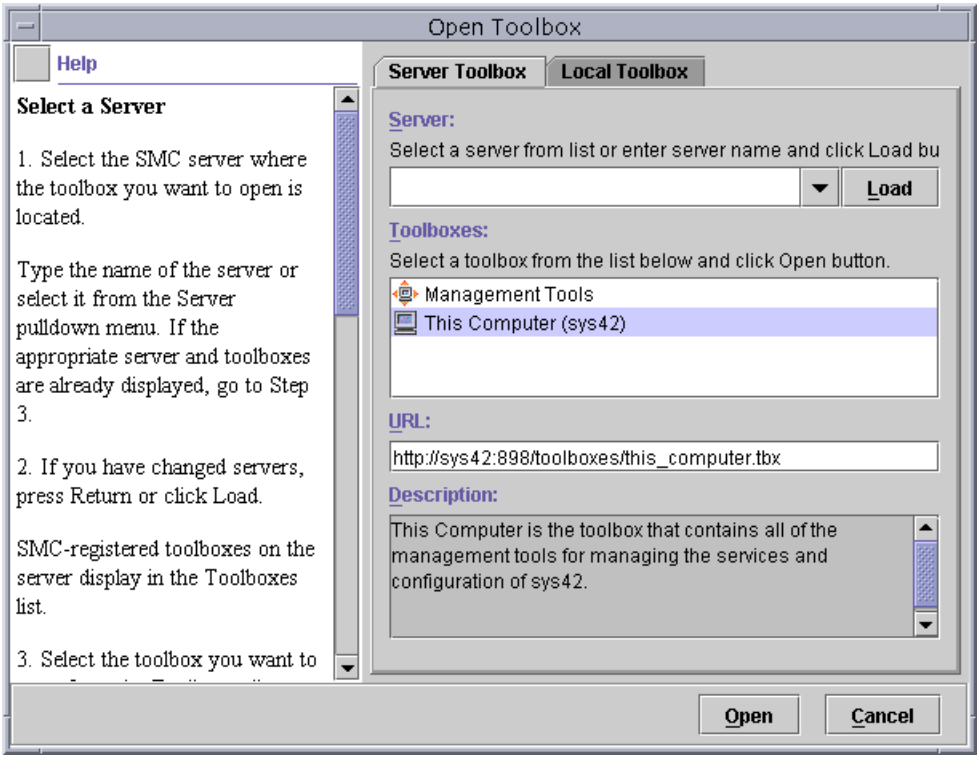


Figure 3-30 Open Toolbox Window – Server Toolbox Tab

3. Select the This Computer (sys42) line entry.
4. Click Open.

The default toolbox opens, as shown in Figure 3-31. The This Computer (sys42) toolbox has been promoted to the top-listed toolbox. You can now select this toolbox or folders within this toolbox, for subsequent add operations.

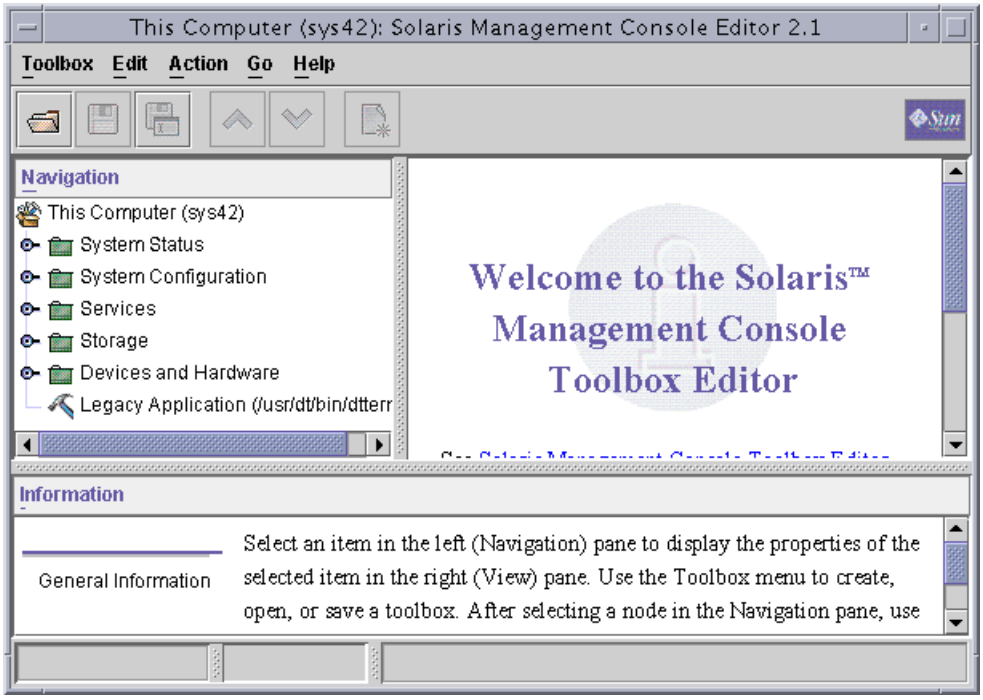


Figure 3-31 Solaris Management Console Editor 2.1 Window – Default Toolbox Expanded

5. To add visibility to the disks from sys44 to the storage folder on sys42, double-click the Storage folder to select the folder and to display its current contents, as shown in Figure 3-32.

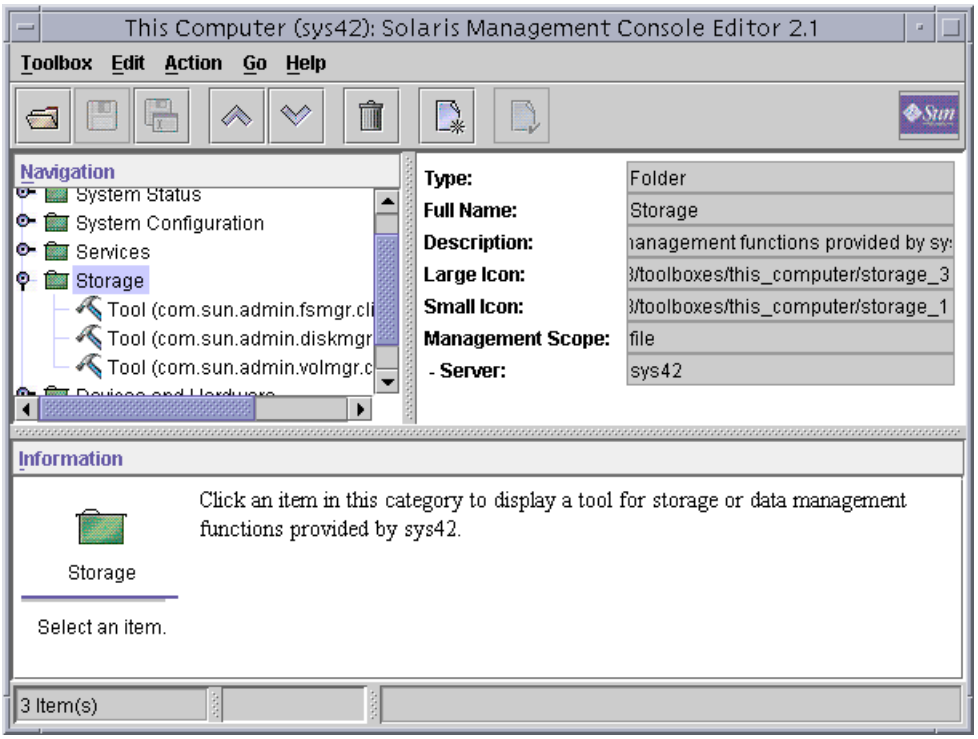


Figure 3-32 Solaris Management Console Editor 2.1 Window – Storage Folder Expanded

Adding a Tool

To make the Solaris Management Console Tools visible between Solaris Management Console servers, use the Add Tool function in the Action menu.

To make the Solaris Management Console Tools visible to other servers, follow these steps:

1. Select Add Tool from the Action menu, as shown in Figure 3-33.

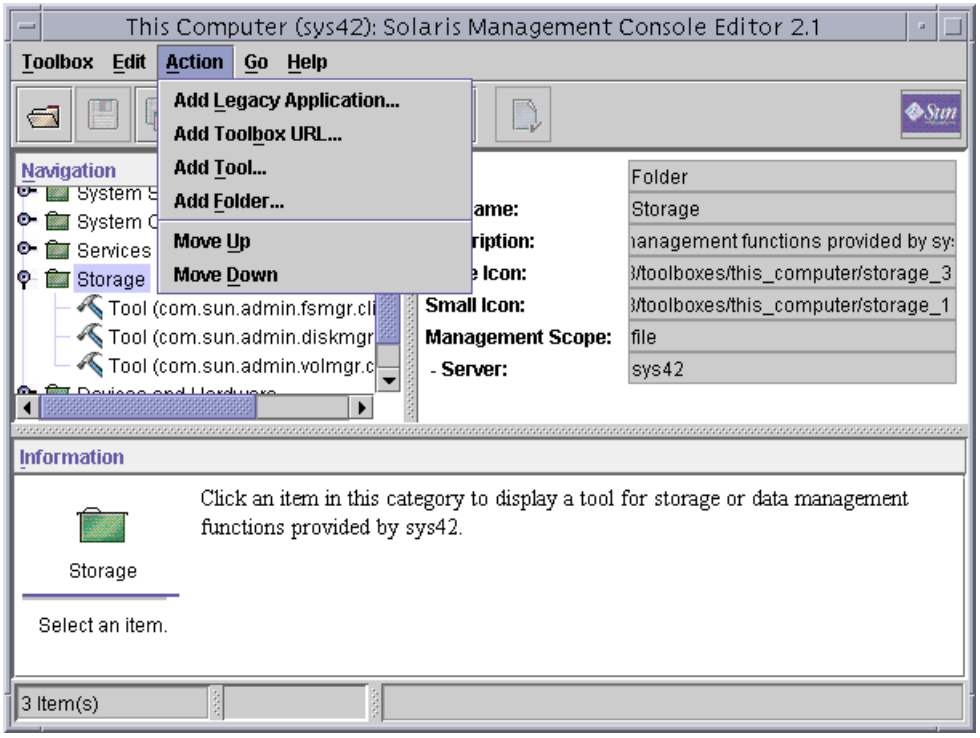


Figure 3-33 Action Menu – Add Tool

The Add Tool wizard launches.

In the Tool Wizard – Step 1 window, you enter the name and an optional port number of the Solaris Management Console server from which to retrieve the tool.

2. In this example, enter server `sys44`, as shown in Figure 3-34.

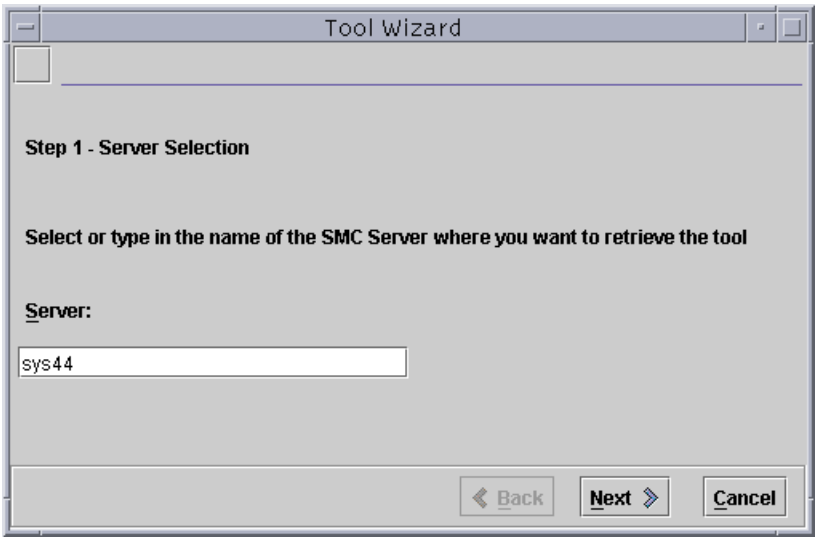


Figure 3-34 Tool Wizard – Step 1 Window

3. Click Next to continue.

In the Tool Wizard – Step 2 window (Figure 3-35):

- If the Solaris Management Console server is running and if any tools are accessible on that server, a list of tools is displayed. You can select the tool you want to add.
- If the server is not running or the host is not currently accessible, you can enter a tool class name for a tool that you know is on the server in the Tool Class Name field.
- You can also specify a tool that is not on the server by entering the tool class name in the Tool Class Name field. If the tool is later added to the server, the tool will already be in the toolbox.

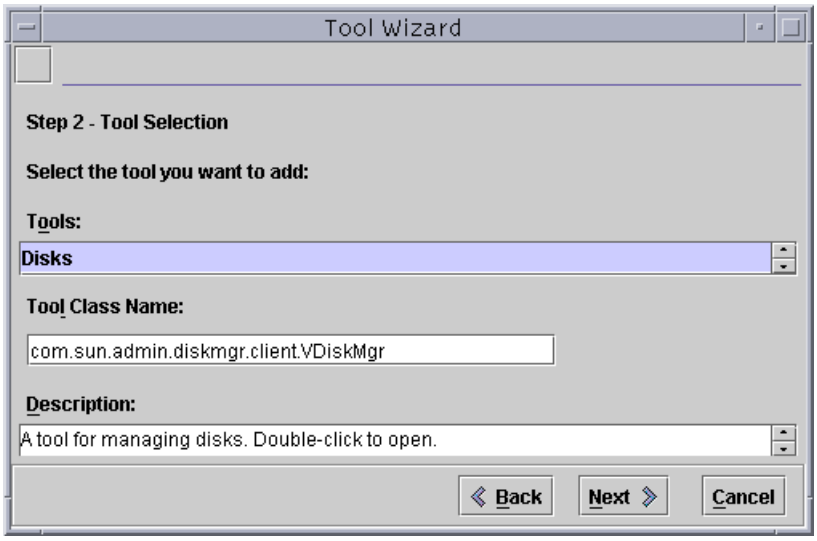


Figure 3-35 Tool Wizard – Step 2 Window

4. In this example, select a description by clicking the down arrow until the Disks tool is displayed, and then select the Disks tool, as shown in Figure 3-35.
5. Type a description.
6. Click Next to continue.

7. Select Override Tool Settings to override the name and description specified in the tool definition, as shown in Figure 3-36.

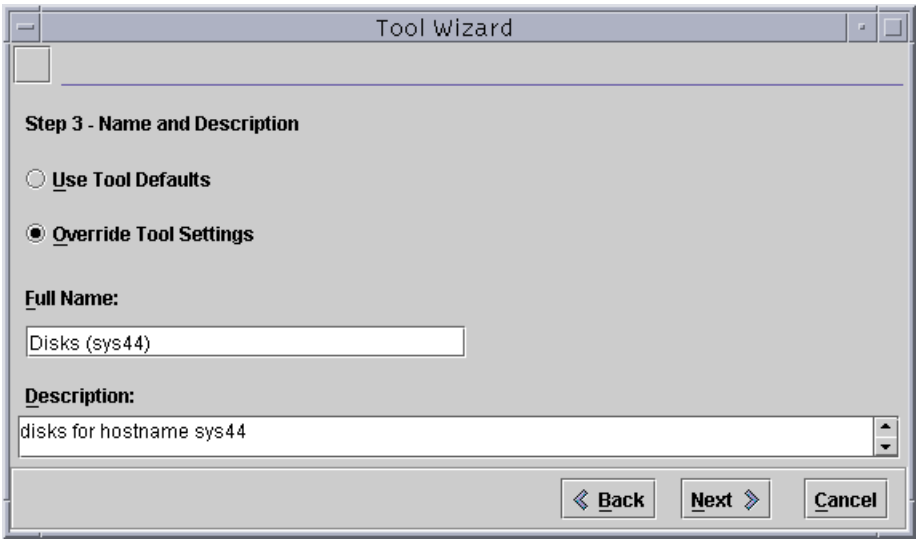


Figure 3-36 Tool Wizard – Step 3 Window

8. Enter a tool name and description that enables you to differentiate between the Disks tools for the local system and those tools on the remote system.
9. Click Next to continue.
10. Select Use Tool Defaults, as shown in Figure 3-37.

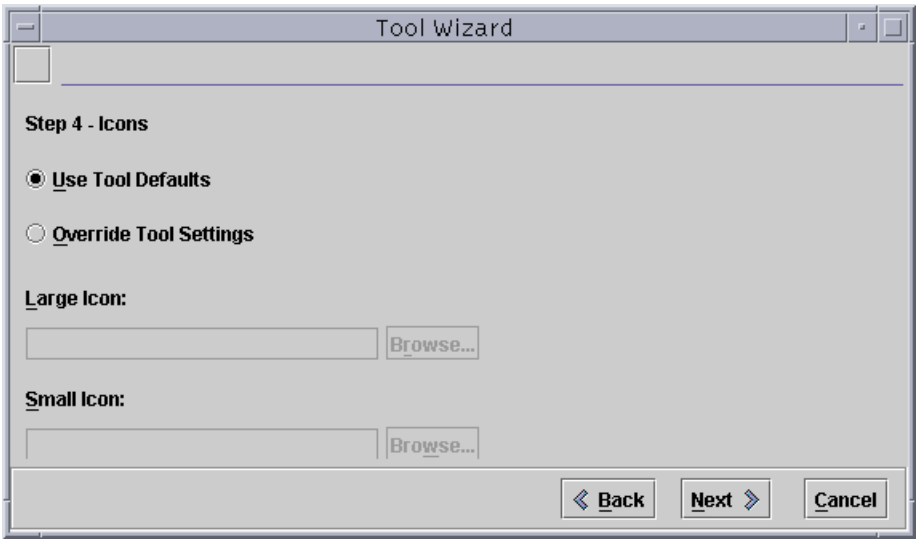


Figure 3-37 Tool Wizard – Step 4 Window

11. Click Next to use the default tool icons.

To override the management scope of the parent node in the Tool Wizard – Step 5 window (Figure 3-38), either:

- Select File from the Management Scope pull down menu, and provide the name of the server where the files are stored.
- Select an alternate management scope (name service) and enter the domain name in the Domain field.

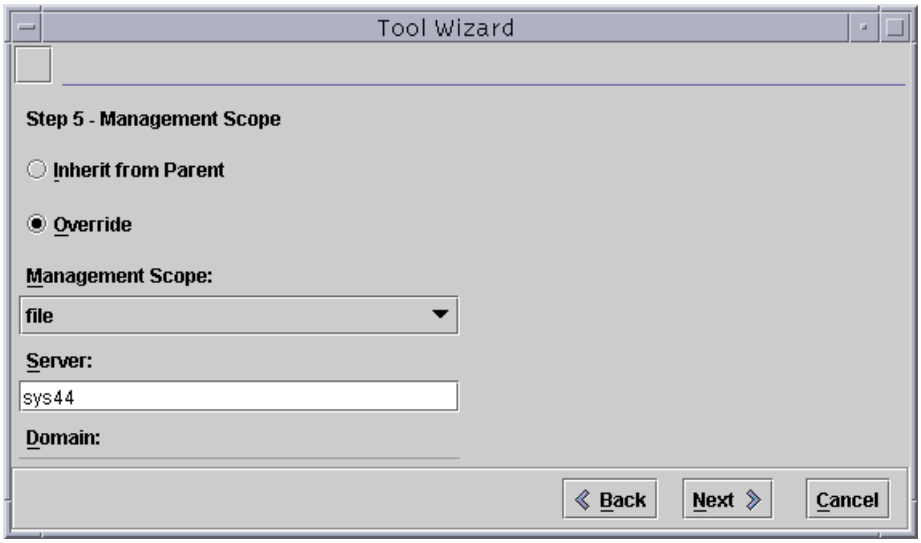


Figure 3-38 Tool Wizard – Step 5 Window

12. In this example, select Override, select file as the management scope, and type **sys44** in the Server field, as shown in Figure 3-38.
13. Click Next to continue.

In the Tool Wizard – Step 6 window (Figure 3-39), you either:

- Select the Load tool when selected option to load each tool only when the specified *tool* is selected in the Solaris Management Console.
- Select the Load tool when toolbox is opened option to immediately load the tool when the This Computer (default) *toolbox*, which contains the specified tool, is selected.

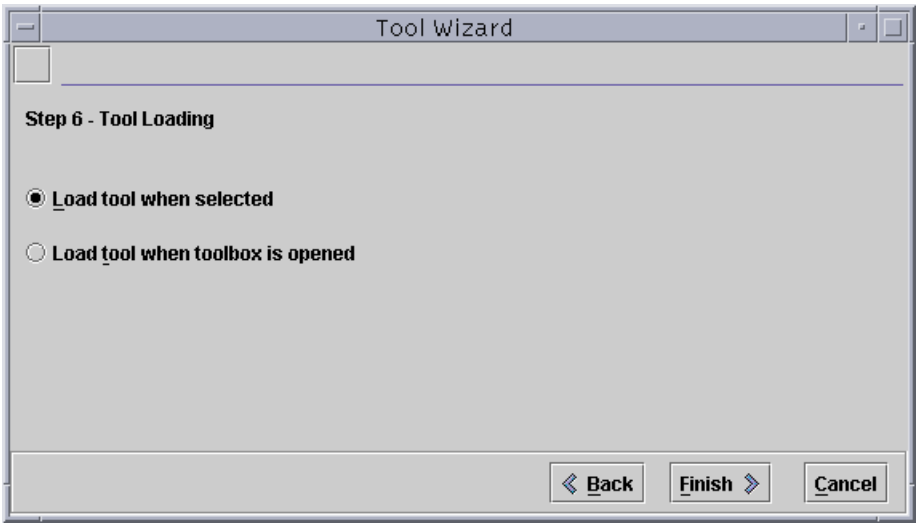


Figure 3-39 Tool Wizard – Step 6 Window

14. In this example, select Load tool when selected, as shown in Figure 3-39.
15. Click Finish.

After the tool is added, you are returned to the Solaris Management Console toolbox editor, and the sys44 disk tool is now displayed as a component of the sys42 Storage folder, as shown in Figure 3-40.

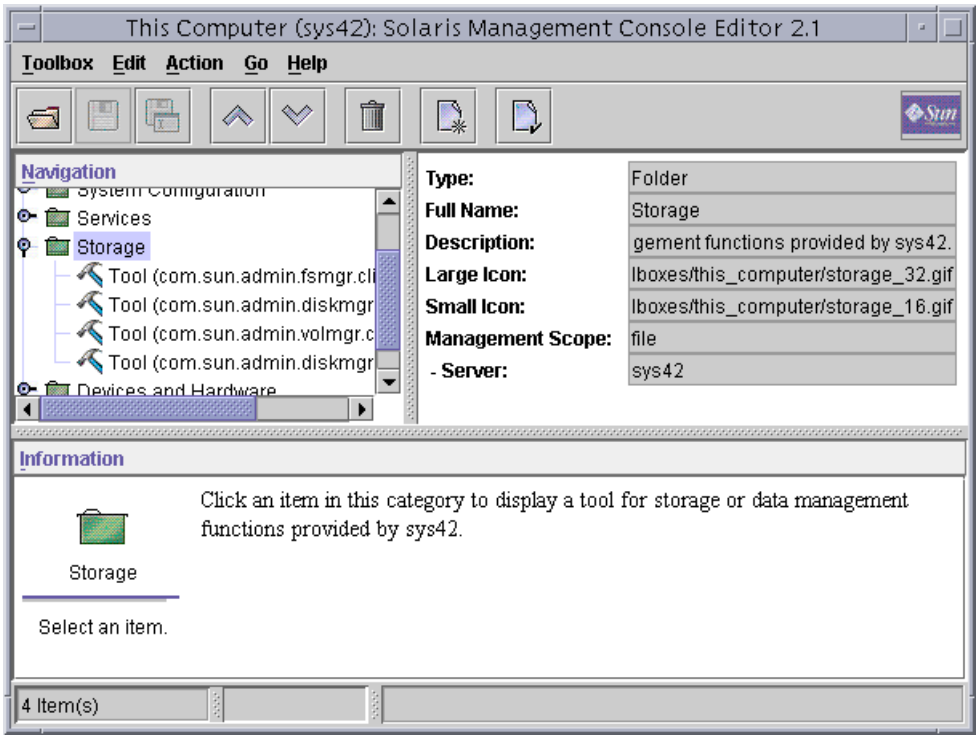


Figure 3-40 Solaris Management Console Editor 2.1 Window – Display Added Tool

Saving a Toolbox

Every time you make a change to a toolbox, you must save the changes to the toolbox using the Solaris Management Console toolbox editor. Then, you must re-open the toolbox in the Solaris Management Console before you can use the new tool. To save the current toolbox, follow these steps:

1. Select Save As from the Toolbox menu, as shown in Figure 3-41.

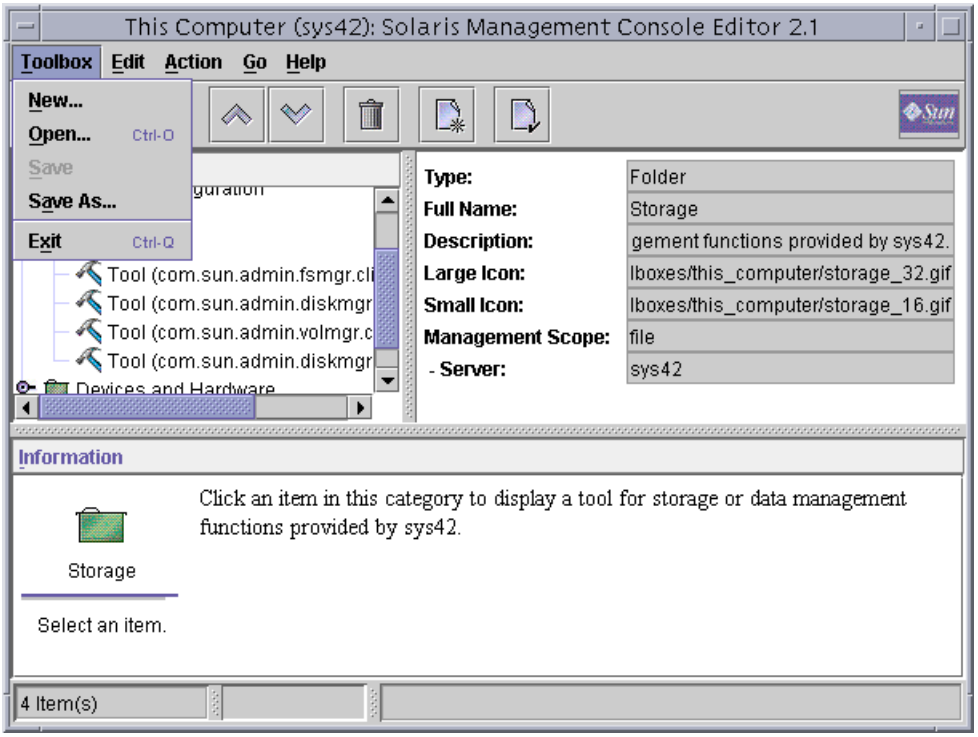


Figure 3-41 Toolbox Menu – Save As

You must be certain to save the correct toolbox, because this save operation overwrites the last toolbox that was saved. In other words, if the root toolbox was the last toolbox that was saved, subsequent save operations point to the root toolbox at:

```
/var/sadm/smc/toolboxes/smc/smc.tbx
```

instead of the default toolbox at:

```
/var/sadm/smc/toolboxes/this_computer/this_computer.tbx.
```

2. Change your path to indicate:

```
/var/sadm/smc/toolboxes/this_computer/this_computer.tbx
```

and click Save, as shown in Figure 3-42.

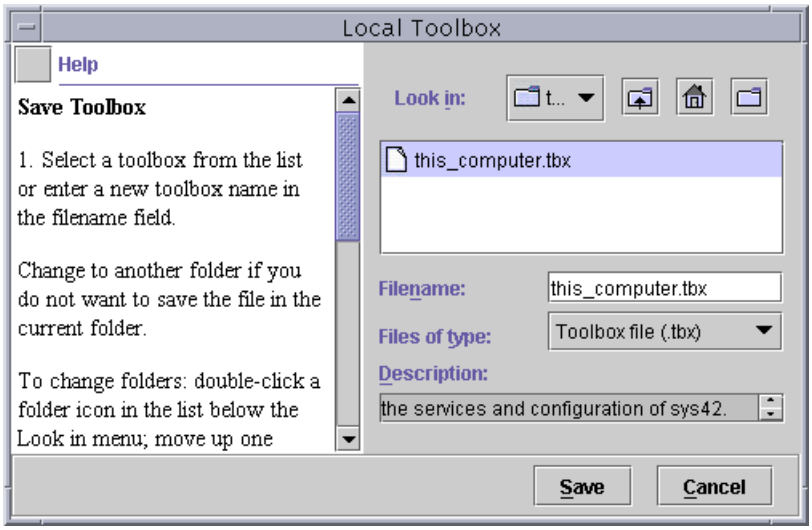


Figure 3-42 Local Toolbox Window

The toolbox changes are saved, and you are returned to the Solaris Management Console Editor 2.1 window, as shown in Figure 3-43.

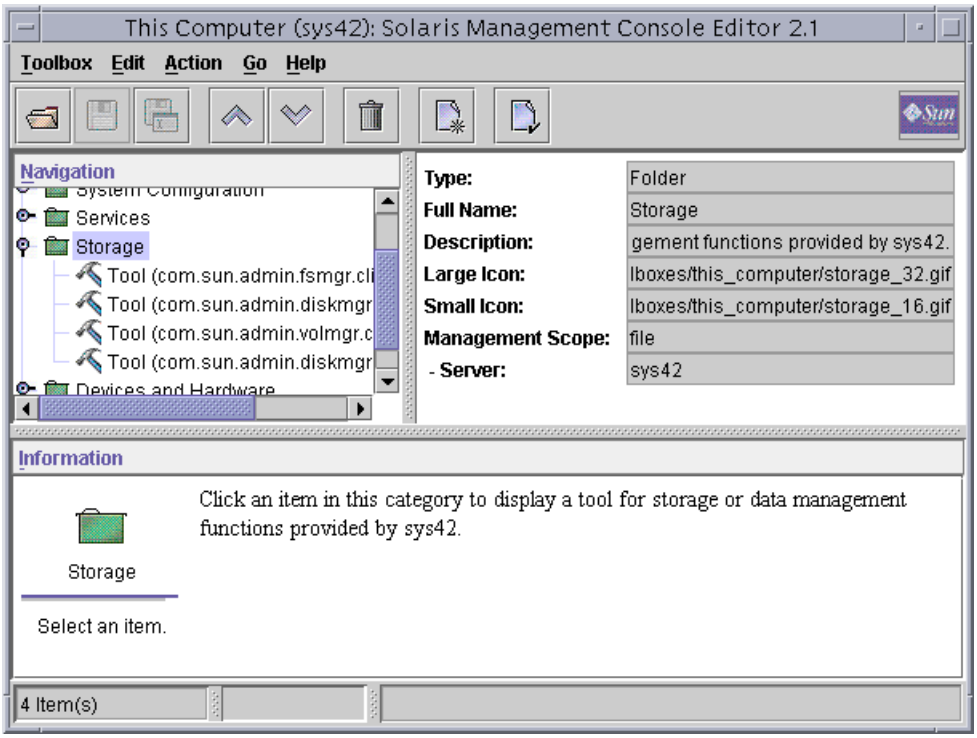


Figure 3-43 Solaris Management Console Editor 2.1 Window – Changes Saved

Testing Tool Access

To test the tool access between the Solaris Management Console servers, reload the updated toolboxes on the Solaris Management Console.

1. Start the Solaris Management Console:

```
# smc &
```

The Solaris Management Console 2.1 window displays the last tool that the Solaris Management Console accessed (Figure 3-44).

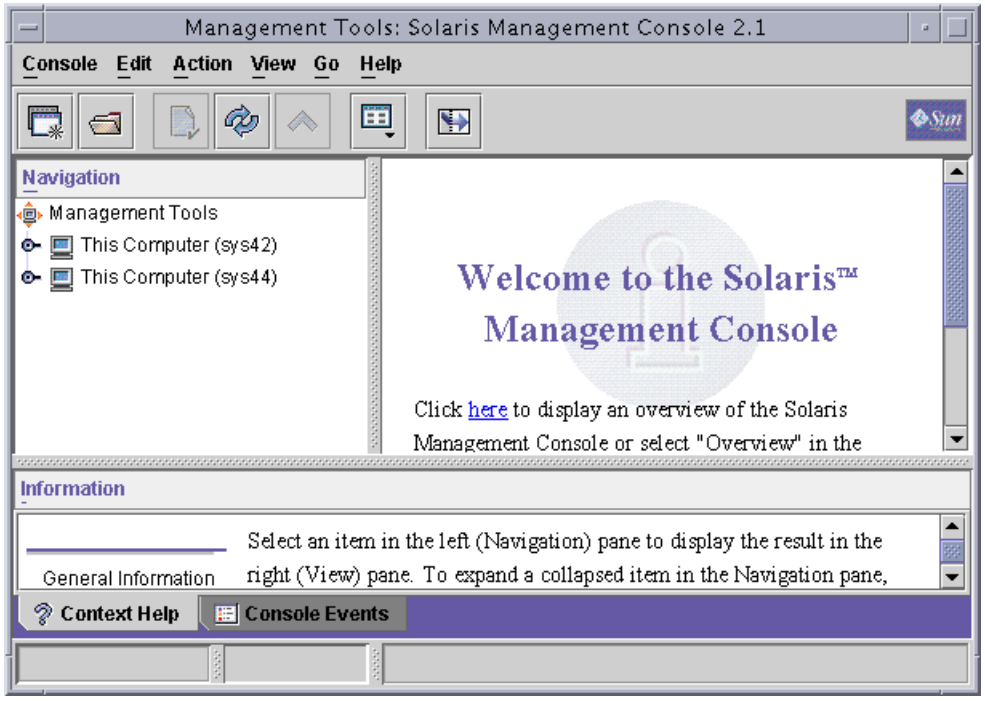


Figure 3-44 Solaris Management Console 2.1 Window – Updated Tools

2. Select Home Toolbox from the Go menu to load and reopen the Home Toolbox the root toolboxes, as shown in Figure 3-45.

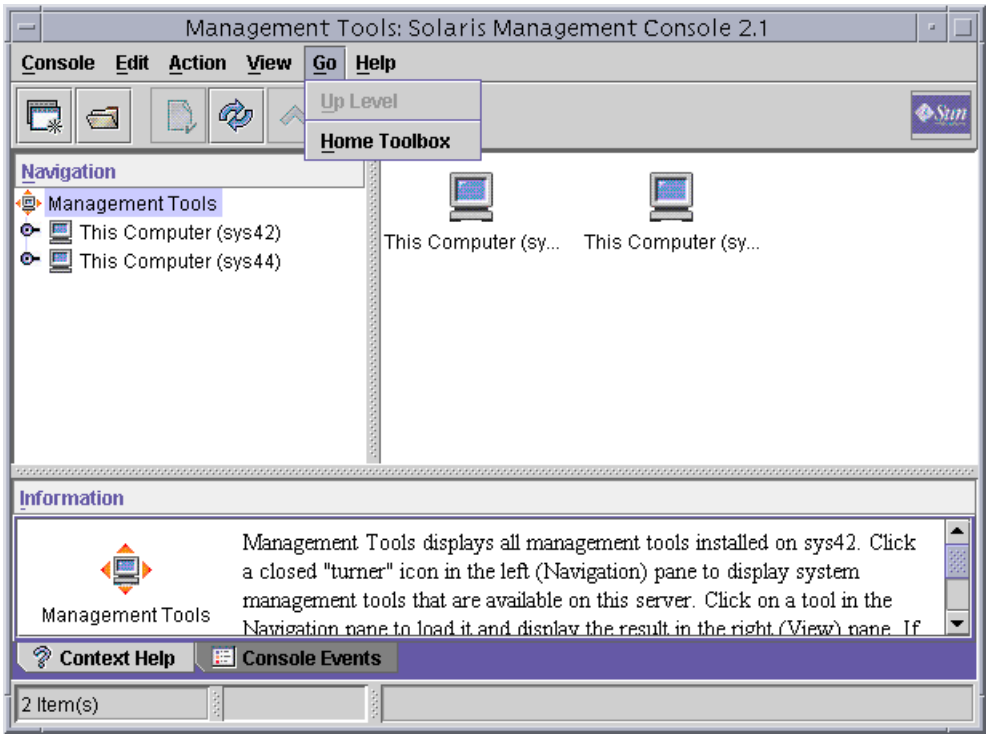


Figure 3-45 Go Menu

The Solaris Management Console 2.1 window displays the original root toolbox, as shown in Figure 3-46.

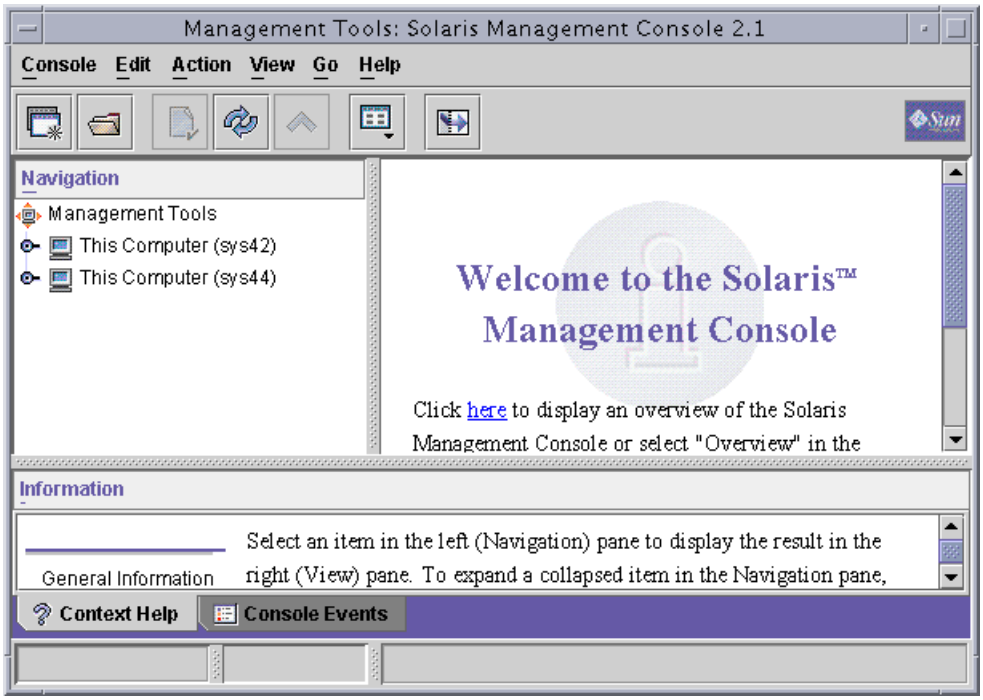


Figure 3-46 Solaris Management Console 2.1 Window – Home Toolbox

3. Double-click the This Computer (sys42) toolbox to open the toolbox, as shown in Figure 3-47.

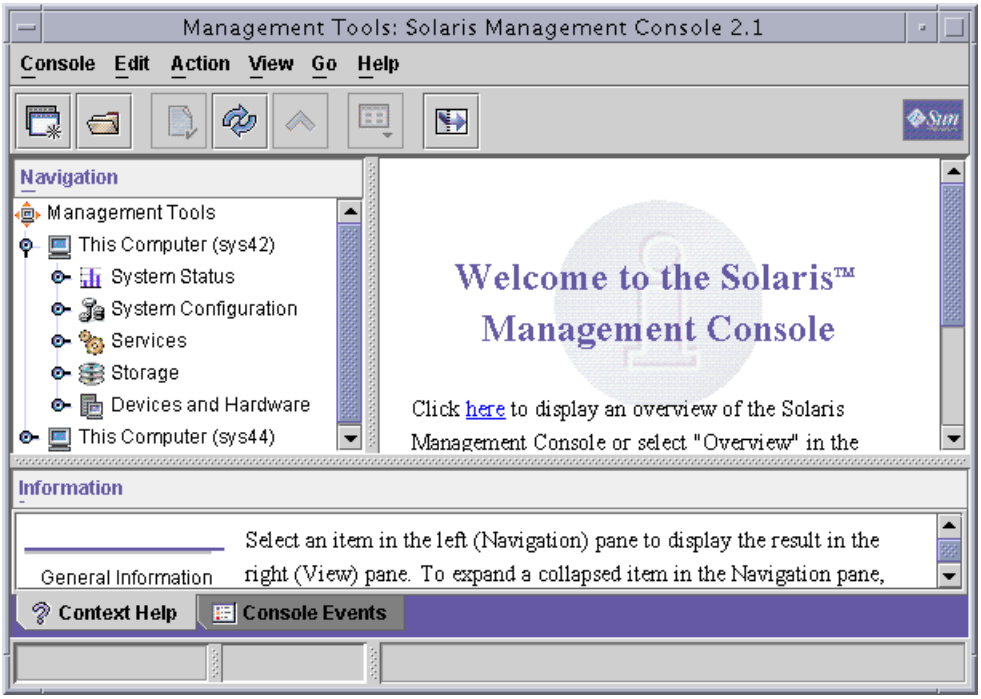


Figure 3-47 Solaris Management Console 2.1 Window – This Computer Expanded

4. Double-click the Storage folder to open the folder, as shown in Figure 3-48.

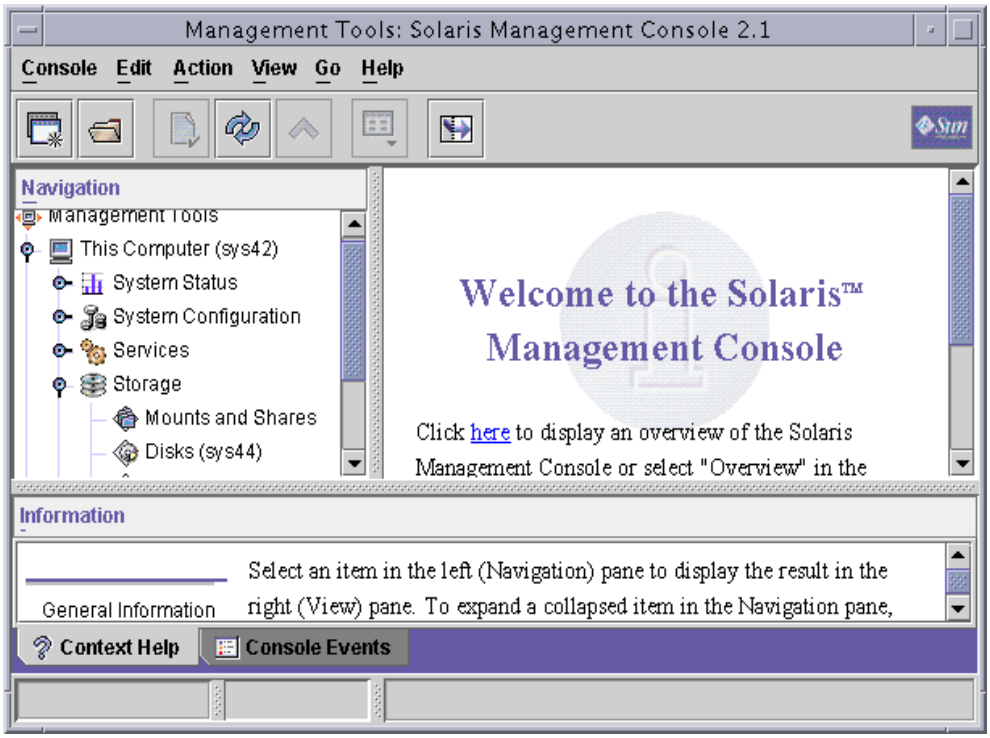


Figure 3-48 Solaris Management Console 2.1 Window – Storage Folder Expanded

The Disks tools are visible for servers sys42 and sys44.

5. Double-click the Disks tool for server sys42.

6. Because the preferences are set to force you to log in when opening a tool, you must log in as shown in Figure 3-49:

   a. Type or verify the name in the User Name field.

   b. Type the password in the Password field.

   c. Click OK.

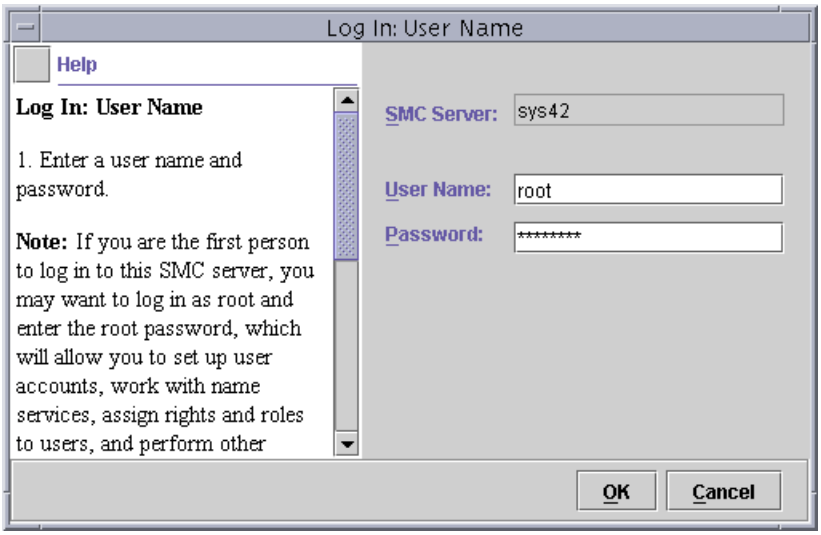


Figure 3-49 Log In: User Name Window

After the system authenticates the login, the disks for system sys42 appear, as shown in Figure 3-50.

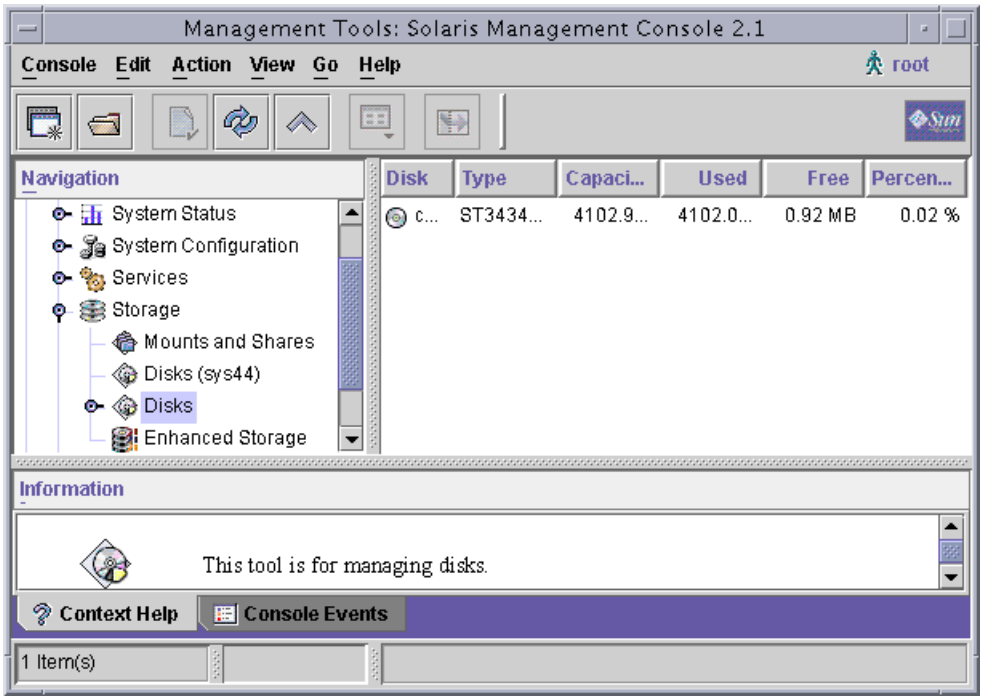


Figure 3-50 Solaris Management Console 2.1 Window – sys42 Disks

7. To display the disks from system sys44, double-click the Disks (sys44) entry in the Navigation pane.

The disks for system sys44 appear, as shown in Figure 3-51. You are not required to log in again because this tool is being accessed from the sys42 toolbox, and you have already authenticated your access to this system.

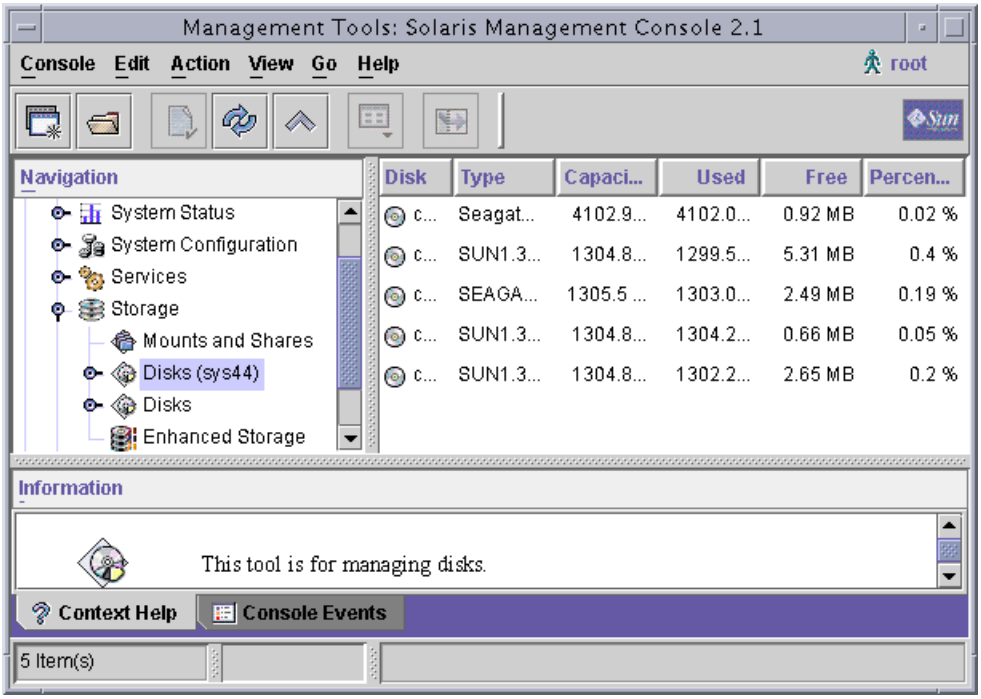


Figure 3-51 Solaris Management Console 2.1 Window – sys44 Disks

8. Close the toolbox by clicking on the turner icon next to the This Computer sys42 entry in the Navigation pane, as shown in Figure 3-52.

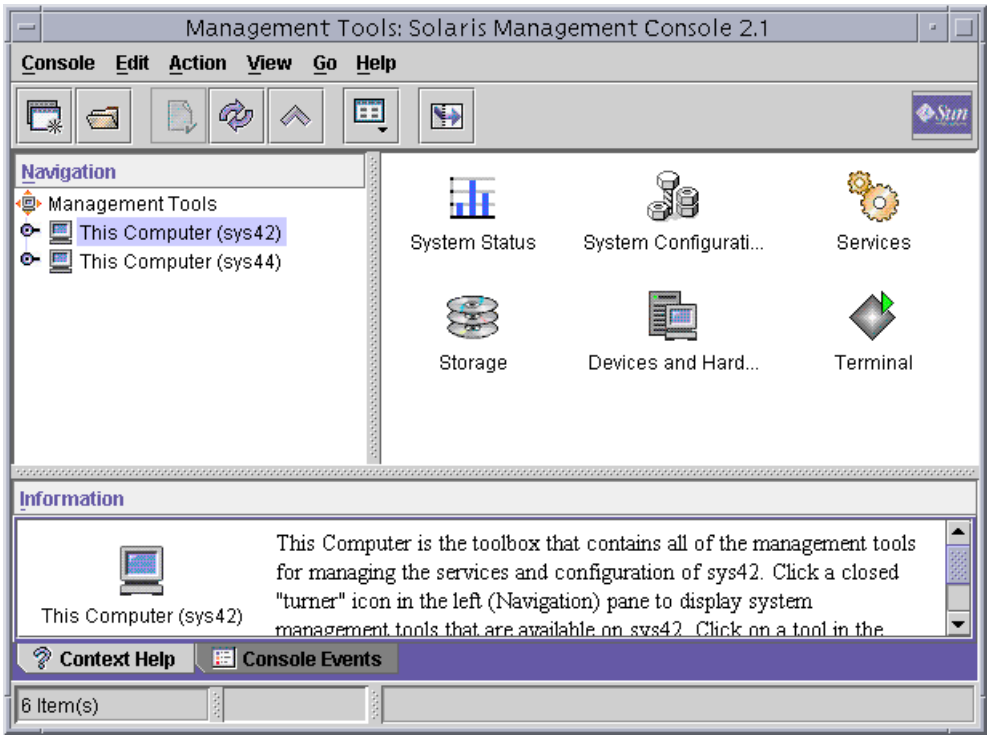


Figure 3-52 Solaris Management Console 2.1 Window – Toolbox Closed

Note – The turner icon is a graphic used to view the tree components. Clicking the turner expands or collapses a component in the hierarchy.

Adding a Tool Using the Command Line

Use the `smcregister` command to add a tool to a toolbox from the command-line as follows:

```
# /usr/sadm/bin/smcregister toolbox add tool \
com.sun.admin.diskmgr.client.VDiskMgr -H sys43:898
```

The previous example adds a Disk tool (from a system named `sys43`) to the default toolbox of the local system. You must identify the tool that you want to add with the full Java technology class name of the tool. You can get the Java technology class name from the Solaris Management Console toolbox editor display if the tool has been previously incorporated into a toolbox. For those tools that have not been previously incorporated into a toolbox, you can get the Java technology class name from the tools programmer.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using the Solaris Management Console (Level 1)

In this exercise, you launch the Solaris Management Console and the toolbox editor, and you add a tool and a toolbox.

Preparation

To prepare for this exercise, refer to your lecture notes as necessary.

You are paired with another student so that, when necessary, the lab scenarios can send commands between two systems, *system1* and *system2*. Lab instructions uses the variable names *system1* and *system2*. Use the translated names as follows:

system1:_____ *system2*:_____

Task Summary

In this exercise, you launch:

- The Solaris Management Console
- The Solaris Management Console toolbox editor
- The Solaris Management Console server

After successfully launching the Solaris Management Console toolbox editor, you update the capabilities of the Solaris Management Console server by:

- Adding a Toolbox URL to an existing root toolbox
- Imbedding a tool from a remote server into the default toolbox of a local server

Exercise: Using the Solaris Management Console (Level 2)

In this exercise, you launch the Solaris Management console and the toolbox editor, and you add a tool and a toolbox.

Preparation

To prepare for this exercise, refer to your lecture notes as necessary.

You are paired with another student so that, when necessary, the lab scenarios can send commands between two systems, *system1* and *system2*. Lab instructions uses the variable names *system1* and *system2*. Use the translated names as follows:

system1:_____ *system2*:_____

Task Summary

Launch the following:

- The Solaris Management Console
- The Solaris Management Console toolbox editor
- The Solaris Management Console server

After successfully launching the Solaris Management Console toolbox editor, you update the capabilities of the Solaris Management Console server by:

- Adding a toolbox URL to an existing root toolbox
- Imbedding a tool from a remote server into the default toolbox of a local server

Tasks

Perform the following tasks.

Task 1 – Status, Stopping, and Starting the Solaris Management Console

Complete the following steps:

1. Log in to your system, and reboot the system to establish a known starting condition for the system's operating environment.

   What is the current status of the Solaris Management Console server?

   _____

2. Start the Solaris Management Console. Allow the toolboxes to launch completely before proceeding.

   What is the current status of the Solaris Management Console server?

   _____

3. Exit the Solaris Management Console.

   What is the current status of the Solaris Management Console server?

   _____

4. Stop the Solaris Management Console server.

   What is the current status of the Solaris Management Console server?

   _____

5. Start the Solaris Management Console toolbox editor.

   _____

6. Start the Solaris Management Console server.

   What is the current status of the Solaris Management Console server?

   _____

   What happens to the Solaris Management Console server when you shut down either the Solaris Management Console or the Solaris Management Console toolbox editor?

   _____

   _____

   What happens to the Solaris Management Console or the Solaris Management Console toolbox editor when you shut down the Solaris Management Console server?

   _____

   _____

Task 2 – Opening a Toolbox

To open a toolbox, on *system1*, open the Management Tools (root) toolbox.

What is the URL for this toolbox?

_____

Task 3 – Adding a Toolbox URL

To add a toolbox URL, complete the following steps:

1. On *system1*, select the Add Toolbox URL from the Action menu.

   How does the server toolbox selection differ from the local toolbox selection?

   _____

2. On *system1*, select Server Toolbox.
3. On *system1*, enter the name of the Solaris Management Console server (*system2*), and click Next.

   What is the default port number used by the Solaris Management Console?

   _____

4. On *system1*, from the Toolboxes list, select the toolbox that contains all of the management tools for managing the services and the configuration of *system1*, and click Next.

   What is the URL for this toolbox?

   _____

5. On *system1*, use the default toolbox name and description.
6. On *system1*, use the default toolbox icons.
7. On *system1*, override the management scope of the parent node.
   a. Select the file management scope from the Management Scope pull-down menu.

      After viewing the list of selections from the Management Scope pull-down menu, what is another term that can be used to describe management scope?

      _____

   b. Enter the name of the server where the file or name service resides (*system2*), and click Finish.

      How has the Solaris Management Console toolbox editor display changed?

      _____

Task 4 – Saving a Toolbox

Complete the following steps:

1. On *system1*, select Management Tools in the Navigation pane.
2. Select Save As from the Toolbox menu.

Note – Prior to saving a Solaris Management Console Toolbox, you should make a backup of the toolbox.

3. On *system1*, select the directory and file location of the root toolbox, and click Save.

   What is directory location of the root toolbox?

   _____

Task 5 – Opening the Toolbox

Complete the following steps:

1. On *system1*, select Open from the Toolbox menu.
2. On *system1*, select the default toolbox named This Computer (*system1*).

   What is the URL for the default toolbox?

   _____

3. On *system1*, click Open.
4. On *system1*, double-click the Storage folder to select the folder and display its contents.

   What are the current contents of the Storage folder?

   _____

Task 6 – Adding a Tool

Complete the following steps:

1. On *system1*, select Add Tool from the Action Menu.
2. On *system1*, enter server *system2*.
3. On *system1*, select the Disks tool, and click Next.

4. On *system1,* select Override Tool Settings to override the name and description specified in the tool definition.
5. On *system1,* enter a tool name and description that will enable you to differentiate between the Disks' tools for the local system and those tools on the remote system.

   What are the name and description fields used?

   _____

6. On *system1,* click Use Tool Defaults, and click Next to use the default tool icons.
7. On *system1,* click Override.
8. On *system1,* select the appropriate management scope (file, in this example) from the Management Scope pull-down menu.
9. The management scope choices are `ldap`, `dns`, `nisplus`, `nis`, or `file`. What is another way to describe management scope?

   _____

10. On *system1,* enter the name of the server (*system2*).
11. On *system1,* select the item Load tool when selected option.

    What is the alternative to the loading the tool when selected option?

    _____

12. On *system1,* click Finish.

Task 7- Saving the Toolbox

Complete the following steps:

1. On *system1* and in the Solaris Management Console toolbox editor, select Save As from the Toolbox menu.
2. On *system1*, change your path to `/var/sadm/smc/toolboxes/this_computer/this_computer.tbx`, and click Save.

Caution – You must select the This Computer (default) toolbox during the save operation to prevent writing over the Management Tools (root) toolbox.

What are the current contents of the Storage folder?

_____

Task 8- Checking Tool Access

Complete the following steps:

1. On *system1*, to re-open the root toolbox, select Home Toolbox from the Go menu.

   What happens when you select the Home Toolbox?

   _____

2. On *system1*, double-click the This Computer (*system1*) toolbox to open the toolbox.

   How does double-clicking the This Computer (*system1*) toolbox differ from using the Home Toolbox in the Go menu?

   _____

3. On *system1*, double-click the Storage folder to open the folder.

   What is the current contents of the Storage folder?

   _____

4. On *system1*, double-click the Disks tool for server *system1*.
5. On *system1*, log in because this is the first tool opened since re-opening the Home Toolbox.

6. On *system1,* to display the disks from system *system2,* double-click the Disks (*system2*) entry in the Navigation pane.
7. On *system1,* close the toolbox by clicking the turner icon next to the This Computer *system1* entry in the Navigation pane.

Exercise: Using the Solaris Management Console (Level 3)

In this exercise, you launch the Solaris Management Console and the toolbox editor, and you add a tool and a toolbox.

Preparation

To prepare for this exercise, refer to your lecture notes as necessary.

You are paired with another student so that, when necessary, the lab scenarios can send commands between two systems, *system1* and *system2*. Lab instructions uses the variable names *system1* and *system2*. Use the translated names as follows:

system1:_____ *system2*:_____

Task Summary

Launch the following:

- The Solaris Management Console
- The Solaris Management Console toolbox editor
- The Solaris Management Console server

After successfully launching the Solaris Management Console toolbox editor, update the capabilities of the Solaris Management Console server by:

- Adding a toolbox URL to an existing root toolbox
- Imbedding a tool from a remote server into the default toolbox of a local server

Tasks and Solutions

Perform the following tasks.

Task 1 – Status, Stopping, and Starting the Solaris Management Console

Complete the following steps:

1. Log in to your system, and reboot the system to establish a known starting condition for the system's operating environment.

```
# init 6
```

What is the current status of the Solaris Management Console server?

```
# /etc/init.d/init.wbem status
Solaris Management Console server not running on port 898.
```

2. Start the Solaris Management Console. Allow the toolboxes to launch completely before proceeding.

```
# smc &
1694
#
```

What is the current status of the Solaris Management Console server?

```
# /etc/init.d/init.wbem status
Solaris Management Console server version 2.1.0 running on port 898.
```

3. Exit the Solaris Management Console.

```
# /etc/init.d/init.wbem status
Solaris Management Console server version 2.1.0 running on port 898.
```

What is the current status of the Solaris Management Console server?

4. Stop the Solaris Management Console server.

```
# /etc/init.d/init.wbem stop
Shutting down Solaris Management Console server on port 898.
```

What is the current status of the Solaris Management Console server?

```
# /etc/init.d/init.wbem status
Solaris Management Console server not running on port 898.
```

5. Start the Solaris Management Console toolbox editor.

```
# smc edit &
1710
#Open Toolbox: http://server:898/toolboxes/smc.tbx failed
Open Toolbox: http://server:898/toolboxes/smc.tbx failed
```

This status message is generated when you stop `wbem` *services and subsequently attempt to launch the Solaris Management Console or Solaris Management Console toolbox editor.*

6. Start the Solaris Management Console server.

```
# /etc/init.d/init.wbem start
Starting Solaris Management Console server version 2.1.0.
endpoint created: :898
Solaris Management Console server is ready.
```

What is the current status of the Solaris Management Console server?

```
# /etc/init.d/init.wbem status
Solaris Management Console server version 2.1.0 running on port 898.
```

What happens to the Solaris Management Console server when you shut down either the Solaris Management Console or the Solaris Management Console toolbox editor?

Shutting down either the Solaris Management Console or the Solaris Management Console toolbox editor has no effect on the Solaris Management Console server.

What happens to the Solaris Management Console or the Solaris Management Console toolbox editor when you shut down the Solaris Management Console server?

Shutting down the Solaris Management Console server prevents the Solaris Management Console or the Solaris Management Console toolbox editor from starting because you cannot open the toolbox.

Task 2 – Opening a Toolbox

To open a toolbox, on *system1,* open the Management Tools (root) toolbox.

What is the URL for this toolbox?

`http://system1:898/toolboxes/smc.tbx.`

Task 3 – Adding a Toolbox URL

To add a toolbox URL, complete the following steps:

1. On *system1,* select the Add Toolbox URL from the Action menu.

   How does the server toolbox selection differ from the local toolbox selection?

   A server toolbox means a computer where the Solaris Management Console server is running, whereas a local toolbox means the computer from which you started the Solaris Management Console toolbox editor.

2. On *system1,* select Server Toolbox.
3. On *system1,* enter the name of the Solaris Management Console server (*system2)*, and click Next.

   What is the default port number used by the Solaris Management Console?

   The default Solaris Management Console port is 898.

4. On *system1,* from the Toolboxes list, select the toolbox that contains all of the management tools for managing the services and the configuration of *system1,* and click Next.

   What is the URL for this toolbox?

   The URL for this toolbox is
   `http://system1:898/toolboxes/this_computer.tbx`.
5. On *system1,* use the default toolbox name and description.
6. On *system1,* use the default toolbox icons.

7. On *system1*, override the management scope of the parent node.

   a. Select the file management scope from the Management Scope pull-down menu.

      After viewing the list of selections from the Management Scope pull-down menu, what is another term that can be used to describe management scope?

      Management scope refers to name service.

   b. Enter the name of the server where the file or name service resides (*system2*), and click Finish.

      How has the Solaris Management Console toolbox editor display changed?

      The Toolbox URL for the remote Solaris Management Console server has been added to the local Solaris Management Console server root toolbox.

Task 4 – Saving Toolbox

Complete the following steps:

1. On *system1*, select Management Tools in the Navigation pane.
2. Click Save As from the Toolbox menu.

Note – Prior to saving a Solaris Management Console Toolbox, you should make a backup of the toolbox.

3. On *system1*, select the directory and file location of the root toolbox, and click Save.

   What is directory location of the root toolbox?

   `/var/sadm/smc/toolboxes/smc/smc.tbx`

Task 5 – Opening the Toolbox

Complete the following steps:

1. On *system1,* select Open from the Toolbox menu.
2. On *system1,* select the default toolbox named This Computer (*system1*).

   What is the URL for the default toolbox?

The URL for the root toolbox *is*
http://system1:898/toolboxes/this_computer.tbx.

3. On *system1,* click Open.
4. On *system1,* double-click the Storage folder to select the folder and display its contents.

   What are the current contents of the Storage folder?

   The current contents of the Storage folder
   Tool (com.sun.admin.fsmgr.client.VFsMgr)
   Tool (com.sun.admin.diskmgr.client.VDiskMgr)
   Tool (com.sun.admin.volmgr.client.VVolMgr)

Task 6 – Adding a Tool

To run, complete the following steps:

1. On *system1,* select Add Tool from the Action Menu.
2. On *system1,* enter server *system2*.
3. On *system1,* select the Disks tool, and click Next.
4. On *system1,* select Override Tool Settings to override the name and description specified in the tool definition.
5. On *system1,* enter a tool name and description that will enable you to differentiate between the Disks' tools for the local system and those tools on the remote system.

   What are the name and description fields used?

   The name will be displayed in the Navigation pane of the Solaris Management Console. If the tool is selected in the Navigation pane, it is also displayed beneath the tool's icon in the Information pane. The description will be displayed in the Information pane if the tool is selected in the View pane.

6. On *system1,* click Use Tool Defaults, and click Next to use the default tool icons.

7. On *system1*, click Override.
8. On *system1*, select the appropriate management scope (file, in this example) from the Management Scope pull-down menu.
9. The management scope choices are `ldap`, `dns`, `nisplus`, `nis`, or `file`. What is another way to describe management scope?

   Another way to describe management scope is name service
10. On *system1*, enter the name of the server (*system2*).
11. On *system1*, select the item Load tool when selected option

    What is the alternative to the Load tool when selected option?

    The alternative to loading the tool when selected is to load the tool when the toolbox is opened.
12. On *system1*, click Finish.

Task 7 – Saving the Toolbox

Complete the following steps:

1. On *system1* and in the Solaris Management Console toolbox editor, select Save As from the Toolbox menu.
2. On *system1*, change your path to `/var/sadm/smc/toolboxes/this_computer/this_computer.tbx`, and click Save.

Caution – You must select the This Computer (default) toolbox during the save operation to prevent writing over the Management Tools (root) toolbox.

What are the current contents of the Storage folder?

The current contents of the Storage folder are:
Tool (`com.sun.admin.fsmgr.client.VFsMgr`)
Tool (`com.sun.admin.diskmgr.client.VDiskMgr`)*
Tool (`com.sun.admin.volmgr.client.VVolMgr`)*
Tool (`com.sun.admin.diskmgr.client.VDiskMgr`)*

Task 8 – Checking Tool Access

Complete the following steps:

1. On *system1,* to re-open the root toolbox, select Home Toolbox from the Go menu.

   What happens when you select the Home Toolbox?

   Clicking on the Home Toolbox re-opens the local system's root toolbox.

2. On *system1,* double-click the This Computer (*system1*) toolbox to open the toolbox.

   How does double-clicking the This Computer (*system1*) toolbox differ from using the Home Toolbox in the Go menu?

   Double-clicking the This Computer (system1) toolbox begins the process of drilling down through the local default toolbox, whereas the Home Toolbox in the Go menu opens the local root toolbox.

3. On *system1,* double-click the Storage folder to open the folder.

   What is the current contents of the Storage folder?

   The current contents of the Storage folder are:
   Storage
   Mounts and Shares
   Disks
   Enhanced Storage
   Disks (system2)

4. On *system1,* double-click the Disks tool for server *system1*.

5. On *system1,* log in because this is the first tool opened since re-opening the Home Toolbox.

6. On *system1,* to display the disks from system *system2,* double-click the Disks (*system2*) entry in the Navigation pane.

7. On *system1,* close the toolbox by clicking the turner icon next to the This Computer *system1* entry in the Navigation pane.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Managing Swap Configuration

Objectives

A system's virtual memory is a combination of the available random access memory (RAM) and disk space. Portions of the virtual memory are reserved as swap space. Swap space can be defined as a temporary storage location that is used when system's memory requirements exceed the size of available RAM.

Upon completion of this module, you should be able to:

- Describe virtual memory
- Configure swap space

The following course map shows how this module fits into the current instructional goal.

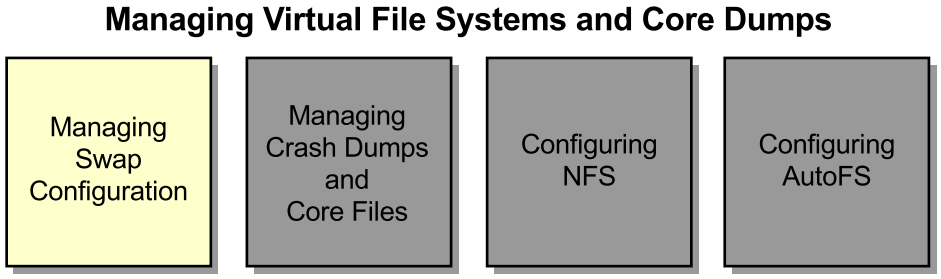


Figure 4-1 Course Map

Introducing Virtual Memory

Virtual memory combines RAM and dedicated disk storage areas known as swap space. Virtual memory management software maps copies of files on disk to virtual addresses. Programs use these virtual addresses rather than real addresses to store instructions and data. Virtual memory makes it possible for the operating environment (OE) to use a large range of memory. However, the `kernel` must translate the virtual memory addresses into real address in RAM before the actual program instruction is performed on a central processing unit (CPU).

Physical RAM

Physical memory refers to the actual RAM installed on a computer. When working with swap space, RAM is the most critical resource in your system. The amount of physical memory varies depending on the server model that hosts the Solaris™ 9 Operating Environment (Solaris 9 OE). The code for each active process and any data required by each process must be mapped into physical memory before execution can take place.

Virtual and Physical Addresses

The Solaris 9 OE virtual memory management system maps the files on disk to virtual addresses in virtual memory. The virtual memory management system then translates the virtual addresses into real, physical addresses in physical memory, because programs require instructions or data in these files. The CPU uses the data and instructions when they are placed in physical memory.

Anonymous Memory Pages

Physical memory pages associated with a running process can contain private data or stack information that does not exist in any file system on disk. Since these memory pages contain information that is not backed by a named file on the disk, these pages are known as anonymous memory pages. Anonymous memory pages are backed by swap space; in other words, swap space is used as a temporary storage location for data while it is swapped out of memory.

Swap Space

While the amount of physical memory in a system is constant, the requirements for using the memory vary. Often processes conflict over which one gets priority to use memory space. Sometimes a lower priority process must give up its memory space allocation to another process. The process with a lower priority has some of its pages in RAM paged out. Anonymous memory pages are placed in a swap area, but file systems are not placed in swap areas, because file system data exists as permanent storage on the disk.

Swap Slices

The primary swap space on the system is a disk slice. In the Solaris 9 OE, the default location for the primary partition is slice 1 of the boot disk. However, you can change the default location during a custom installation. Each time you reboot the system, an entry in the `/etc/vfstab` file configures the swap partition. As additional swap space becomes necessary, you can configure additional swap slices. Plan your swap slice location carefully. If you have additional storage space outside of the system disk, place the swap slice on the second drive to reduce the load on the system disk drive.

Swap Files

It is also possible to provide additional swap space on a system by using swap files. Swap files are files that reside on a file system, and that have been created using the `mkfile` command. These files might be useful in some cases. For example, swap files are useful when additional swap space is required, but there are no free disk slices and reslicing a disk to add more swap is not a practical solution. Swap files can be permanently included in the swap configuration by creating an entry for the swap file in the `/etc/vfstab` file.

The `swapfs` File System

When the `kernel` runs a process, swap space for any private data or stack space for the process must be reserved. The reservation occurs in case the stack information or private data might need to be paged out of physical memory; for example, if there are multiple processes contending for limited memory space.

On operating systems that do not provide virtual swap space, you must configure large amounts of physical swap space on systems to accommodate the reservations. You must always reserve swap space to accommodate for the possibility that a task gets paged out for a task with a higher priority.

Because of the virtual swap space provided by the `swapfs` file system in the Solaris 9 OE, there is less need for physical swap space on systems with a large available memory. The decreased need for physical swap space occurs because the `swapfs` file system provides virtual swap space addresses rather than real physical swap space addresses in response to swap space reservation requests. Therefore, you need physical swap space on disk, only in the event that the physical man pages containing private data need to be paged out.

Figure 4-2 shows that the swap space resides outside the physical RAM as a swap partition or as a swap file.

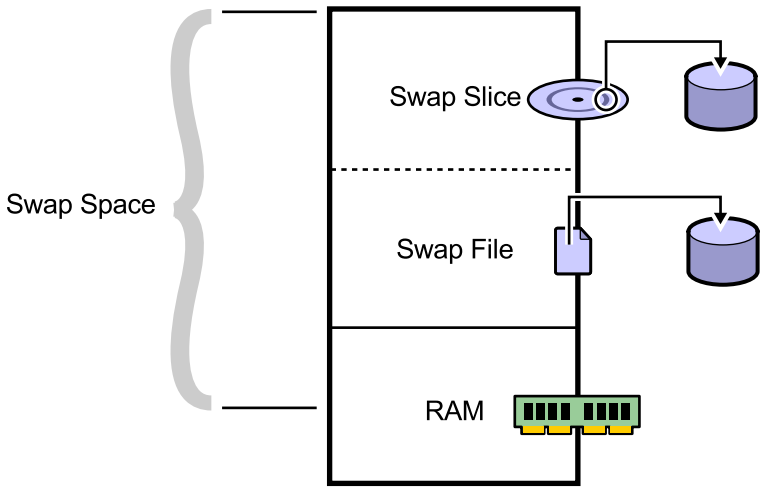


Figure 4-2 Swap Space

Paging

Paging is the transfer of selected memory pages between RAM and the swap areas. When you page private data to swap spaces, physical RAM is made available for other processes to use. If you need the pages that were paged out, you can retrieve them (page them in) from swap and map them back into physical memory. Moving these pages back into RAM might require more paging (page outs) of other process's pages to make room. Swapping is the movement of all memory pages associated with a process, between RAM and a disk.

Use the `pagesize` command to display the size of a memory page in bytes. The default page size for the Solaris 9 OE is 8192 bytes.

You can use the Multiple Page Size Support (MPSS) service to run legacy applications with larger memory page sizes. Using larger page sizes can significantly improve the performance of programs using large amounts of memory. Large pages must be mapped to addresses that are multiples of the page size.

Swapping does not *typically* occur in the Solaris OE. However, the requirement within the Solaris OE to reserve swap space prior to executing any process, makes it necessary that some amount of swap space is available. The required amount of swap space varies from system to system. The amount of available swap space must satisfy two criteria:

- It must be sufficient to supplement physical RAM to meet the needs of concurrently running processes.
- It must be sufficient to hold a crash dump (in a single slice).

Configuring Swap Space

The `swap` utility provides a method of adding, deleting, and monitoring the swap areas used by the `kernel`. Swap area changes made from the command line are not permanent and are lost after a reboot. To create permanent additions to the swap space, create an entry in the `/etc/vfstab` file. The entry in the `/etc/vfstab` file is added to the swap space at each reboot.

Displaying the Current Swap Configuration

Figure 4-3 shows the relationship between the used swap space, which consists of allocated and reserved swap spaces, and the available swap space.

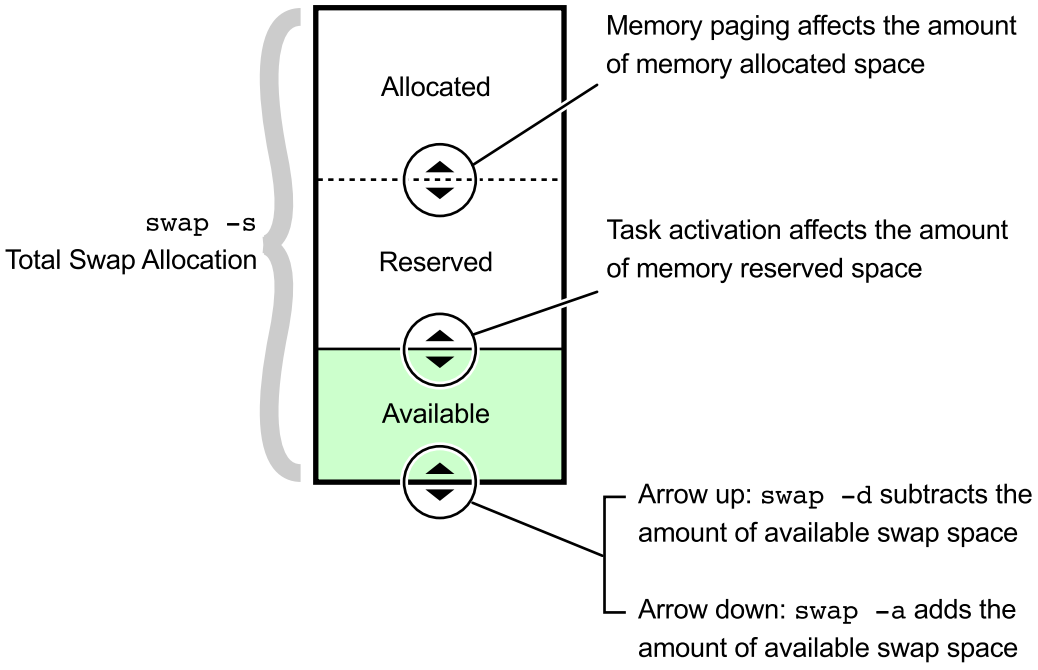


Figure 4-3 Swap Space Allocation

To view the current swap space allocation, complete the following steps:

1. List a summary of the system's virtual swap space.

```
# swap -s
total: 106240k bytes allocated + 8528k reserved = 114768k used, 566776k
available
```

2. List the details of the system's physical swap areas.

```
# swap -l
swapfile             dev     swaplo   blocks   free
/dev/dsk/c0t0d0s1   136,1    16       1206736  1084736
```

Adding Swap Space

When the swap space requirements of the system exceed the current swap space available, you can use the following procedures to add additional swap space to your system.

Adding Swap Slices

To add a swap slice, complete the following steps:

1. Edit the `/etc/vfstab` file to add information describing the swap slice.

```
# vi /etc/vfstab
#device             device     mount    FS       fsck   mount    mount
#to mount           to fsck    point    type     pass   at boot  opt
```

2. Add the following line to create the swap slice.

```
/dev/dsk/c1t0d0s3    -          -         swap     -      no         -
```

3. Use the `swap -a` command to add additional swap area.

```
# swap -a /dev/dsk/c1t0d0s3
```

Note – When the system is subsequently rebooted, the new swap slice `/dev/dsk/c1t0d0s3` will be automatically included as part of the swap space as a result of adding the entry to the `/etc/vfstab` file.

Adding Swap Files

To add a swap file, complete the following steps:

1. The `/export/data` file system appears to have adequate space to create an additional swap file. Create a 20-Mbyte swap file named `swapfile` in the `/export/data` directory.

```
# mkfile 20m /export/data/swapfile
```

2. Add the swap file to the system's swap space.

```
# swap -a /export/data/swapfile
```

3. List the details of the modified system swap space.

```
# swap -l
swapfile             dev     swaplo    blocks     free
/dev/dsk/c0t0d0s1    136,1   16        1206736    1084736
/export/data/swapfile -      16        40944      40944
```

4. List a summary of the modified system swap space.

```
# swap -s
total: 106256k bytes allocated + 8512k reserved = 114768k used, 587512k
available
```

5. To use a swap file when the system is subsequently rebooted, add an entry for the swap file in the `/etc/vfstab` file.

```
# vi /etc/vfstab
#device              device     mount   FS      fsck   mount    mount
#to mount            to fsck    point   type    pass   at boot  opt
/export/data/swapfile  -         -       swap    -      no       -
```

Removing Swap Space

If you no longer need the additional swap space, you can delete the swap space by removing the additional swap slices and swap files.

Removing Swap Slices

To remove a swap slice, complete the following steps:

1. Delete a swap slice from the current swap configuration.

```
# swap -d /dev/dsk/c1t0d0s3
```

2. To prevent the swap slice from being configured as part of the swap configuration during a reboot or change of run level, edit the `/etc/vfstab` file, and remove the swap slice entry from the file.

Removing Swap Files

To remove a swap file, complete the following steps:

1. Delete a swap file from the current swap configuration.

```
# swap -d /export/data/swapfile
```

2. Remove the file to free the disk space that it is occupying.

```
# rm /export/data/swapfile
```

3. To prevent the swap file from being configured as part of the swap configuration during a reboot or change of run level, edit the `/etc/vfstab` file, and remove the swap file entry.

Note – The output of the `df -h /export/data/swapfile` command shows the space in use until you remove the swap file.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Managing swap Utility Configuration (Level 1)

In this exercise, you add and remove a swap space.

Preparation

To prepare for this exercise:

- Each student will configure swap space on their assigned workstation.
- Each student should unconfigure the additional swap space before exiting the lab exercise.
- Make sure that the `/export` directory exists on your system.
- Each student uses disk slice 5 on their system for this exercise.

Note – The actual swap statistics will vary depending on the configuration of each system.

To support disk requirements for the remaining labs in this course, partition the second disk using the information in Table 4-1.

Table 4-1 Partition Information

Slice	Size	Use
0	5 Mbytes	State database replica
1	5 Mbytes	State database replica
3	5 Mbytes	State database replica
4	310 Mbytes	Root (/) mirror
5	512 Mbytes	Swap/dump
6	free	Flash

Tasks

Perform the following tasks:

- Obtain a report of the swap space usage on the system.
- List the swap areas that are configured on the system.
- Configure additional swap space using a swap file.
- Configure additional swap space using a disk partition.
- Unconfigure the additional swap space.

Exercise: Managing swap Utility Configuration (Level 2)

In this exercise, you add and remove a swap space.

Preparation

To prepare for this exercise:

- Each student will configure swap space on their assigned workstation.
- Each student should unconfigure the additional swap space before exiting the lab exercise.
- Make sure that the `/export` directory exists on your system.
- Each student uses disk slice 5 on their system for this exercise.

Note – The actual swap statistics will vary depending on the configuration of each system.

To support disk requirements for the remaining labs in this course, partition the second disk using the information in Table 4-2.

Table 4-2 Partition Information

Slice	Size	Use
0	5 Mbytes	State database replica
1	5 Mbytes	State database replica
3	5 Mbytes	State database replica
4	310 Mbytes	Root (/) mirror
5	512 Mbytes	Swap/dump
6	free	Flash

Task Summary

Perform the following tasks:

- Obtain a report of the swap space usage on the system.
- List the swap areas that are configured on the system.
- Configure additional swap space using a swap file.
- Configure additional swap space using a disk partition.
- Unconfigure the additional swap space.

Tasks

To determine the amount of disk space used by a `swapfs` file system, complete the following steps:

1. Run the `swap -s` command.

   What is the total number of bytes actually allocated and currently in use?

   _____

   What is the number of bytes allocated and not currently in use, but reserved by processes for possible future use?

   _____

   What is the total amount of swap space, both allocated and reserved?

   _____

   What is the total swap space currently available for future reservation and allocation?

   _____

2. Run the `swap -l` command.

   List the physical swap area configured on your system.

   _____

   How much total swap space is in the listed swap device?

   _____

   How much space is available for the listed device?

   _____

3. Run the `df -h` command.

   Does the `/export` directory have sufficient space to add 20 Mbytes of swap space?

   _____

4. Create a 20-Mbyte swap file in the `/export` directory, and add it to the system swap space.

   _____

5. Use the `swap -l` command to verify that the new swap space is available.

   _____

6. Use the `swap -s` command to verify that the new swap space is available.

   How does the output differ between the `swap -l` command and the `swap -s` command?

   _____

7. Remove the swap file created in Step 4.
8. Use the `swap` utility to verify that the swap space is no longer available.
9. Add a disk partition as a swap slice to your existing swap space.
10. Add the new swap partition to the `/etc/vfstab` file to make the partition permanent. To verify this change, you must reboot the system.
11. After the reboot, verify that the additional swap space exists by using the `swap` utility.

    Is the newly listed swap partition the same as the one you added to the `/etc/vfstab` file?

    _____

12. Verify the additional swap space exists using the `df -h` command.

    Why is the newly created swap space listed in the `/etc/vfstab` file not listed in the output of the `df -h` command?

    _____

13. To return the system to its initial swap configuration, remove the additional swap space entry from the `/etc/vfstab` file, and reboot the system.
14. Remove the additional swap slice using the `swap -d` command.

Exercise: Managing swap Utility Configuration (Level 3)

In this exercise you add and remove a swap space.

Preparation

To prepare for this exercise:

- Each student will configure swap space on their assigned workstation.
- Each student should unconfigure the additional swap space before exiting the lab exercise.
- Make sure that the `/export` directory exists on your system.
- Each student uses disk slice 5 on their system for this exercise.

Note – The actual swap statistics will vary depending on the configuration of each system.

To support disk requirements for the remaining labs in this course, partition the second disk using the information in Table 4-3.

Table 4-3 Partition Information

Slice	Size	Use
0	5 Mbytes	State database replica
1	5 Mbytes	State database replica
3	5 Mbytes	State database replica
4	310 Mbytes	Root (/) mirror
5	512 Mbytes	Swap/dump
6	free	Flash

Task Summary

Perform the following tasks:

- Obtain a report of the swap space usage on the system.
- List the swap areas that are configured on the system.
- Configure additional swap space using a swap file.
- Configure additional swap space using a disk partition.
- Unconfigure the additional swap space.

Tasks and Solutions

This section describes the tasks you must perform, and lists the solutions to these tasks. To determine the amount of disk space used by a `swapfs` file system, complete the following steps:

1. Run the `swap -s` command.

```
# swap -s
total: 106240k bytes allocated + 8528k reserved = 114768k used, 566776k
available
```

What is the total number of bytes actually allocated and currently in use?

106, 240 Kbytes

What is the number of bytes allocated and not currently in use but reserved by processes for possible future use?

8528 Kbytes

What is the total amount of swap space, both allocated and reserved?

114,768 Kbytes

What is the total swap space currently available for future reservation and allocation?

566,776 Kbytes

2. Run the `swap -l` command.

```
# swap -l
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1 16     1206736  1084736
```

List the physical swap area configured on your system.

```
/dev/dsk/c0t0d0s1
```

How much total swap space is in the listed swap device?

12,06,736 Kbytes

How much space is available for the listed device?

1,084,736 Kbytes

3. Run the `df -h` command.

```
# df -h
Filesystem             size     used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0       1.4G    876M    503M    64%    /
/proc                     0       0       0      0%    /proc
mnttab                    0       0       0     0%    /etc/mnttab
fd                        0       0       0     0%    /dev/fd
swap                    552M     24K    552M    1%    /var/run
swap                    554M      2M    552M    1%    /tmp
/dev/dsk/c0t0d0s7       2.0G      9K    1.9G    1%    /export
```

Does the `/export` directory have sufficient space to add 20 Mbytes of swap space?

Yes

4. Create a 20-Mbyte swap file in the `/export` directory, and add it to the system swap space.

```
# mkfile 20m /export/swapfile
# swap -a /export/swapfile
```

5. Use the `swap -l` command to verify that the new swap space is available.

```
# swap -l
swapfile             dev   swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1  16    1206736  1084736
/export/swapfile     -      16     40944    40944
```

6. Use the `swap -s` command to verify that the new swap space is available.

```
# swap -s
total: 106256k bytes allocated + 8512k reserved = 114768k used, 587512k
available
```

How does the output differ between the `swap -l` command and the `swap -s` command?

The `swap -l` *command output is a listing of each space, whereas the* `swap -s` *command output only produces a cumulative report.*

7. Remove the swap file created in Step 4.

```
# swap -d /export/swapfile
# rm /export/swapfile
```

8. Use the `swap` utility to verify that the swap space is no longer available.

```
# swap -l
swapfile             dev   swaplo   blocks   free
/dev/dsk/c0t0d0s1   136,1  16       1206736  1084736
# swap -s
total: 106240k bytes allocated + 8528k reserved = 114768k used, 566776k
available
```

9. Add a disk partition as a swap slice to your existing swap space.

```
# swap -a /dev/dsk/c#t#d#s5
```

10. Add the new swap partition to the `/etc/vfstab` file to make the partition permanent. To verify this change, you must reboot the system.

```
# vi /etc/vfstab
/dev/dsk/c#t#d#s5  -  -   swap  -  no  -
```

11. After the reboot, verify that the additional swap space exists by using the `swap` utility.

```
# swap -l
swapfile             dev     swaplo    blocks   free
/dev/dsk/c0t0d0s1   136,1    16        1206736  1206736
/dev/dsk/c1t0d0s3   32,3     16        614704   614704
```

Is the newly listed swap partition the same as the one you added to the `/etc/vfstab` file?

Yes

12. Verify the additional swap space exists using the `df -h` command.

    Why is the newly created swap space listed in the `/etc/vfstab` file not listed in the output of the `df -h` command?

    The `df -h` *output does not produce an entry for the additional* `swap` *utility devices, however the added swap space is reflected in the total swap space.*

13. To return the system to its initial swap configuration, remove the additional swap space entry from the `/etc/vfstab` file, and reboot the system.

```
# vi /etc/vfstab
```

14. Remove the additional swap space using the `swap -d` command.

```
# swap -d /dev/dsk/c#t#d#s5
```

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 5

Managing Crash Dumps and Core Files

Objectives

When an operating system has a fatal error, it generates a crash dump file (crash dump). When a process has a fatal error, it generates a core file. Upon completion of this module, you should be able to:

- Manage crash dump behavior
- Manage core file behavior

The following course map shows how this module fits into the current instructional goal.

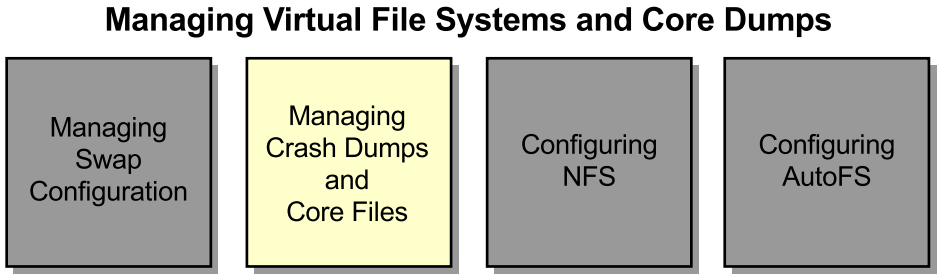


Figure 5-1 Course Map

Managing Crash Dump Behavior

If a fatal operating system error occurs, the operating system prints a message to the console, describing the error. The operating system then generates a crash dump by writing some of the contents of the physical memory to a predetermined dump device, which is typically a local disk slice. You can configure the dump device by using the `dumpadm` command. After the operating system has written the crash dump to the dump device, the system reboots. The crash dump is saved for future analysis to help you determine the cause of the fatal error.

The Crash Dump

If the Solaris OE `kernel` encounters a problem that might endanger the integrity of data or when the `kernel` encounters an unexpected hardware fault, the `panic` routine is executed. Despite its name, a system panic is a well-controlled event where memory contents are copied to a disk partition defined as a dump device. Whatever the cause, the crash dump itself provides valuable information to help your support engineer diagnose the problem.

When an operating system crashes, the `savecore` command is automatically executed during a boot. The `savecore` command retrieves the crash dump from the dump device and then writes the crash dump to a pair of files in your file system:

- The `savecore` command places `kernel` core information in the `/var/crash/`*`nodename`*`/vmcore.`*`X`* file, where *`nodename`* is the name returned by `uname -n`, and *`X`* is an integer identifying the dump.
- The `savecore` command places name list information and symbol table information in the `/var/crash/`*`nodename`*`/unix.`*`X`* file.

Note – Within the crash dump directory, a file named `bounds` is created. The `bounds` file holds a number that is used as a suffix for the next dump to be saved.

Together, these data files form the saved crash dump. You can use the `dumpadm` command to configure the location of the dump device and the `savecore` directory.

By default, the dump device is an appropriate swap partition. Swap partitions are disk partitions reserved as a virtual memory, backing store for the operating system. The swap partition contains only temporary data; therefore, permanent data is overwritten by the crash dump.

Displaying the Current Dump Configuration

To view the current dump configuration, enter the `dumpadm` command without arguments, as shown in the following example:

```
# dumpadm
Dump content: kernel pages
Dump device: /dev/dsk/c0t0d0s1 (swap)
Savecore directory: /var/crash/host1
Savecore enabled: yes
```

The previous example shows the set of default values:

- The dump content is set to `kernel` memory pages only
- The dump device is a swap disk partition
- The directory for `savecore` files is set to `/var/crash/host1`
- The `savecore` command is set to run automatically on reboot

The following example shows that the current configuration is located in the `/etc/dumpadm.conf` file:

```
# cat /etc/dumpadm.conf
# dumpadm.conf
#
# Configuration parameters for system crash dump.
# Do NOT edit this file by hand -- use dumpadm(1m) instead.
#
DUMPADM_DEVICE=/dev/dsk/c0t0d0s1
DUMPADM_SAVDIR=/var/crash/host1
DUMPADM_CONTENT=kernel
DUMPADM_ENABLE=yes
```

Changing the Crash Dump Configuration

The `dumpadm` command manages the configuration of the operating system crash dump facility.

Note – Perform all modifications to the crash dump configuration by using the `dumpadm` command, rather than attempting to edit the `/etc/dumpadm.conf` file. Editing the file might result in an inconsistent system dump configuration.

The syntax of the dumpadm command is:

```
/usr/sbin/dumpadm [-nuy][-c content-type][-d dump-device]
    [-m mink| minm| min%] [-r root-dir] [-s savecore-dir]
```

where:

`-n`	Modifies the dump configuration so it does not run the `savecore` command automatically on reboot.
`-u`	Forcibly updates the `kernel` dump configuration based on the contents of the `/etc/dumpadm.conf` file.
`-y`	Modifies the dump configuration so that the `savecore` command is run automatically on reboot. This is the default.
`-c` *content-type*	Specifies the contents of the crash dump. The *content-type* can be `kernel`, `all`, or `curproc`. The `curproc` content type includes the `kernel` memory pages and the memory pages of the currently executing process.
`-d` *dump-device*	Modifies the dump configuration to use the specified dump device. The dump device can be an absolute path name or `swap`.

`-m` *min*`k`
`-m` *min*`m`
`-m min%`
	Creates a `minfree` file in the current *savecore-dir* directory indicating that the `savecore` command should maintain at least the specified amount of free space in the file system in which the *savecore-dir* directory is located:
	• `k` – Indicates a positive integer suffixed with the unit `k`, specifying kilobytes.
	• `m` – Indicates a positive integer suffixed with the unit `m`, specifying megabytes.
	• `%` – Indicates a percent (%) symbol, indicating that the `minfree` value is computed as the specified percentage of the total, current size of the file system that contains the *savecore-dir* directory.

`-r` *root-dir*	Specifies an alternative root directory relative to which the `dumpadm` command should create files. If the `-r` argument is not specified, the default root directory "`/`" is used.
`-s` *savecore-dir*	Modifies the dump configuration to use the specified directory to save files written by the `savecore` command. The default *savecore-dir* directory is `/var/crash/`*hostname*, where *hostname* is the output of the `uname -n` command.

Managing Core File Behavior

When a process terminates abnormally, it typically produces a core file. You can use the `coreadm` command to specify the name or location of core files produced by abnormally terminating processes.

Core Files

A core file is a point-in-time copy (snapshot) of the RAM allocated to a process. The copy is written to a more permanent medium, such as a hard disk. A core file is useful in analyzing why a particular program crashed.

A core file is also a disk copy of the address space of a process, at a certain point-in-time. This information identifies items, such as the task name, task owner, priority, and instruction queue, in execution at the time that the core file was created.

When a core file occurs, the operating system generates two possible copies of the core files, one copy known as the global core file and the other copy known as the per-process core file. Depending on the system options in effect, one file, both files, or no files can be generated. When generated, a global core file is created in mode 600 and is owned by the superuser. Non-privileged users cannot examine files with these permissions.

Ordinary per-process core files are created in mode 600 under the credentials of the process. The owner of the process can examine files with these permissions.

Displaying the Current Core File Configuration

You use the `coreadm` command without options to display the current configuration.

```
# coreadm
1   global core file pattern:
2      init core file pattern: core
3           global core dumps: disabled
4      per-process core dumps: enabled
5      global setid core dumps: disabled
6 per-process setid core dumps: disabled
7      global core dump logging: disabled
```

Note – The line numbers in the example are not part of the configuration. They are part of the example only to assist with the following description of the file.

Line 1 of the output identifies the name to use for core files placed in a global directory. When generated, a global core file is created with mode 600 and is owned by the superuser. Non-privileged users cannot examine files with this permission.

Line 2 of the output identifies the default name that per-process core files must use. This name is set for the `init` process, meaning it is inherited by all other processes on the system.

Line 3 indicates that global core files are disabled.

Line 4 indicates that core file generation in the current working directory of a process is enabled. Per-process core files are created with mode 600 with the credentials of the process. Only the owner of the process can examine these files.

Lines 5 and 6 indicate that generation of per-process core files for processes with `setuid` or `setgid` permissions are disabled, and the generation of global core files for processes with `setuid` or `setgid` permissions is disabled. If core file generation for processes with `setuid` or `setgid` permissions is enabled, the core files generated are owned by the superuser and have their permissions set to 600.

Line 7 identifies whether global core dump logging is enabled.

Caution – A process that has a `setuid` mode presents security issues with respect to dumping core files. The files might contain sensitive information in its address space to which the current non-privileged owner of the process should not have access. Therefore, by default, `setuid` core files are not generated because of this security issue.

By viewing the `/etc/coreadm.conf` file, you can verify the same configuration parameters that were displayed with the `coreadm` command.

```
# cat /etc/coreadm.conf
# coreadm.conf
#
# Parameters for system core file configuration.
# Do NOT edit this file by hand -- use coreadm(1) instead.
COREADM_GLOB_PATTERN=
COREADM_INIT_PATTERN=core
COREADM_GLOB_ENABLED=no
COREADM_PROC_ENABLED=yes
COREADM_GLOB_SETID_ENABLED=no
COREADM_PROC_SETID_ENABLED=no
COREADM_GLOB_LOG_ENABLED=no
```

Changing the Core File Configuration

The `coreadm` command allows you to control core file generation behavior. For example, you can use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. The flexibility of this configuration makes it easier to track problems by examining the core files in a specific directory whenever a process or daemon terminates abnormally. This flexibility also makes it easy to locate and remove core files on a system.

Note – You should make all modifications to the `coreadm` configuration at the command line by using the `coreadm` command instead of editing the `/etc/coreadm.conf` file.

You can enable or disable two configurable `core` file paths, per-process and global, separately. If a global `core` file path is enabled and set to `/corefiles/core`, for example, then each process that terminates abnormally produces two core files: one in the current working directory, and one in the `/corefiles/core` directory.

Note – If the directory defined in the global core file path does not exist, you must create it.

Users can run the `coreadm` command with the `-p` option to specify the file name pattern for the operating system to use when generating a per-process core file.

```
coreadm [-p pattern] [pid]...
```

Only the `root` user can run the following `coreadm` command options to configure system-wide core file options.

```
coreadm [-g pattern] [-i pattern] [-d option ... ] [-e option ... ]
```

"The coreadm Command Options" on page 5-10 describes the `core` file options.

The coreadm Command Options

The following are some options to the coreadm command.

Note – Only the superuser can use all options, except for the -p option, which a regular user can use.

-i *pattern*	Sets the per-process core file name pattern from init to *pattern*. This option is the same as the coreadm -p *pattern* 1 command, except that the setting is persistent after a reboot.
-e *option*	Enables the specified core file option, where option is: • global – Enables core dumps by using the global core pattern. • process – Enables core dumps by using the per-process core pattern. • global-setid – Enables setid core dumps by using the global core pattern. • proc-setid – Enables setid core dumps by using the per-process core pattern. • log – Generates a syslog (3) message when a user attempts to generate a global core file.
-d *option*	Disables the specified core file option; see the -e *option* for descriptions of possible options. You can specify multiple -e and -d options by using the command line.
-u	Updates system-wide core file options from the contents of the configuration file /etc/coreadm.conf. If the configuration file is missing or contains invalid values, default values are substituted. Following the update, the configuration file is resynchronized with the system core file configuration.
-g *pattern*	Sets the global core file name pattern to *pattern*. The pattern must start with a forward slash (/), and can contain any of the special embedded variables described in Table 5-1 on page 5-11.

`-p` *`pattern`* Sets the per-process core file name pattern to *`pattern`* for each of the specified process IDs (PIDs). The pattern can contain any of the special embedded variables described in Table 5-1 and does not have to begin with a forward slash (`/`). If *`pattern`* does not begin with "`/`", it is evaluated relative to the current directory in effect when the process generates a core file.

A non-privileged user can only apply the `-p` option to processes owned by that user. The superuser can apply the `-p` option to any process.

A core file named *`pattern`* is a file system path name with embedded variables. The embedded variables are specified with a leading percent (%) character. The operating system expands these variables from values in effect when the operating system generates a core file. The possible variables are listed in Table 5-1.

Table 5-1 Pattern Options for the `coreadm` Command

Option	Meaning
`%p`	PID
`%u`	Effective user ID (EUID)
`%g`	Effective group ID (EGIC)
`%f`	Executable file name
`%n`	System node name (`uname -n`)
`%m`	Machine hardware name (`uname -m`)
`%t`	The time in seconds since midnight January 1, 1970
`%%`	Literal %

Examples of the coreadm Command

Example 1 – Setting the Core File Name Pattern as a Regular User

When executed from a user's `$HOME/.profile` or `$HOME/.login` file, the following entry sets the core file name pattern for all processes run during the login session:

```
coreadm -p core.%f.%p $$
```

Note – The $$ variable is the PID of the currently running shell. The per-process core file name pattern is inherited by all child processes.

Example 2 – Dumping a User's Core Files Into a Subdirectory

The following command places all of the user's core files into the `corefiles` subdirectory of the user's home directory, differentiated by the system node name. This example is useful for users who use many different systems, but share a single home directory across multiple systems.

```
$ coreadm -p $HOME/corefiles/%n.%f.%p $$
```

Example 3 – Enabling and Setting the Core File Global Name Pattern

The following is an example of setting *system-wide* parameters that add the executable file name and PID to the name of any core file that is created:

```
# coreadm -g /var/core/core.%f.%p -e global
```

For example, the core file name pattern `/var/core/core.%f.%p` causes the `xyz` program with PID `1234` to generate the core file `/var/core/core.xyz.1234`.

To verify that this parameter is now part of the core file configuration, run the `coreadm` command again:

```
# coreadm
     global core file pattern: /var/core/core.%f.%p
       init core file pattern: core
            global core dumps: enabled
       per-process core dumps: enabled
      global setid core dumps: disabled
 per-process setid core dumps: disabled
     global core dump logging: disabled
```

Example 4 – Checking the Core File Configuration for Specific PIDs

Running the `coreadm` command with a list of PIDs reports each process's per-process core file name pattern, for example:

```
# coreadm 278 5678
278: core.%f.%p
5678: /home/george/cores/%f.%p.%t
```

Only the owner of a process or the superuser can query a process by using the `coreadm` command with a list of PIDs.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Collecting the Crash Dump and Core Dump (Level 1)

In this exercise, you configure crash dumps and core files.

Preparation

To prepare for this exercise, refer to the material in the module.

Tasks

Perform the following tasks:

- Use the `dumpadm` command to view the current dump configuration.
- Use the `dumpadm` command to change the current dump configuration to a new swap partition.
- Collect a pair of crash dump files.
- Use the `coreadm` command to view the default configuration for potential core files.
- Configure the system to collect global and per-process core files.
- Collect a global and a per-process core file.

Exercise: Collecting the Crash Dump and Core Dump (Level 2)

In this exercise, you configure crash dumps and core files.

Preparation

To prepare for this exercise, refer to the material in the module.

Task Summary

In this exercise, you perform the following tasks:

- Use the `dumpadm` command to view the current dump configuration.
- Use the `dumpadm` command to change the current dump configuration to a new swap partition.
- Collect a pair of crash dump files.
- Use the `coreadm` command to view the default configuration for potential core files.
- Configure the system to collect global and per-process core files.
- Collect a global and a per-process core file.

Tasks

Perform the following tasks.

Task 1 – Using the dumpadm Command to Display the Core File Directory Location

Complete the following steps:

1. Use the `dumpadm` command without arguments to view the current dump configuration.
2. Fill in the configuration parameters from the output:

   Dump content: _____

   Dump device: _____

   The `savecore` directory: _____

   Is `savecore` enabled? _____
3. Use the `dumpadm` command to change the dump device to the external disk drive slice 5.
4. Run the `sync` command to flush all previously unwritten system buffers out to disk, ensuring that all file modifications up to that point will be saved.
5. Force the `kernel` to panic and save a crash dump by using the `savecore -L` command
6. When the system reboots, make sure the crash dump succeeded by using the `file` command on the files of the `savecore` directory.

Task 2 – Using the coreadm Command to Display Default Configuration for Potential Core Files

Complete the following steps:

1. Use the `coreadm` command to the display default initial configuration.
2. Create the core file directory, and enable a global core file path.
3. Turn on logging to generate a message when a global core file is attempted.
4. Display the configuration information to verify the changes.
5. In another terminal window, create a new directory named `/dir`, and change to that directory.

6. Run the `pwd` command to see the current working directory.
7. Run the `ps` command to get the PID of the new shell, and send a SIGFPE signal (Signal 8) to the new shell by using the `kill` command. (The SIGFPE signal forces a core file.)

Note – The `kill -8` command terminates the shell and the Common Desktop Environment (CDE) terminal window in which it is executed.

8. In the original terminal window, check to see if a core file exists in the current working directory of the old shell. Use the `file` command to verify that the core file is from the old shell.
9. Use the `ls` command to check for a core file in the `/var/core` directory.
10. Observe the messages generated in the console window and the `/var/adm/messages` file due to `coreadm` logging being enabled.

Exercise: Collecting the Crash Dump and Core Dump (Level 3)

In this exercise, you configure crash dumps and core files.

Preparation

To prepare for this exercise, refer to the material in the module.

Task Summary

Perform the following tasks:

- Use the `dumpadm` command to view the current dump configuration.
- Use the `dumpadm` command to change the current dump configuration to a new swap partition.
- Collect a pair of crash dumps.
- Use the `coreadm` command to view the default configuration for potential core files.
- Configure the system to collect global and per-process core files.
- Collect a global and a per-process core file.

Tasks and Solutions

This section describes the tasks you must perform and lists the solutions to these tasks.

Task 1 – Using the dumpadm Command to Display the Core File Directory Location

Complete the following steps:

1. Use the `dumpadm` command with no arguments to view the current dump configuration.

```
# dumpadm
```

2. Fill in the configuration parameters from the output:

   Dump content: `kernel` *pages*

   Dump device: `/dev/dsk/c0t0d0s1 (swap)`

   The `savecore` directory: `/var/crash/sys42`

   Is `savecore` enabled? *Yes*

3. Use the `dumpadm` command to change the dump device to the external disk drive slice 5.

```
# dumpadm -d /dev/dsk/c#t#d#s5
```

4. Run the `sync` command to flush all previously unwritten system buffers out to disk, ensuring that all file modifications up to that point will be saved.

```
# sync
```

5. Force the `kernel` to panic and save a crash dump by using the `savecore -L` command.

```
# savecore -L
```

6. When the system reboots, make sure the crash dump succeeded by using the `file` command on the files of the `savecore` directory.

   The output shown should be similar to the following:

```
# cd /var/crash/savecore_directory
# ls
bounds    unix.0    vmcore.0
# file vmcore.0
vmcore.0:       SunOS 5.9 Beta 64-bit SPARC crash dump from 'sys42'
```

Task 2 – Using the coreadm Command to Display Default Configuration for Potential Core Files

Complete the following steps:

1. Use the coreadm command to the display default initial configuration.

   The command and resulting output should be similar to the following:

```
# coreadm
     global core file pattern:
       init core file pattern: core
            global core dumps: disabled
       per-process core dumps: enabled
      global setid core dumps: disabled
 per-process setid core dumps: disabled
     global core dump logging: disabled
```

2. Create the core file directory, and enable a global core file path.

```
# mkdir /var/core
# coreadm -e global -g /var/core/core.%f.%p
```

3. Turn on logging to generate a message when a global core file is attempted.

```
# coreadm -e log
```

4. Display the configuration information to verify the changes.

```
# coreadm
     global core file pattern: /var/core/core.%f.%p
       init core file pattern: core
            global core dumps: enabled
       per-process core dumps: enabled
      global setid core dumps: enabled
 per-process setid core dumps: disabled
     global core dump logging: enabled
```

5. In another terminal window, create a new directory named /dir, and change to that directory.

```
# mkdir /dir
# cd /dir
```

6. Run the pwd command to see the current working directory.

```
# pwd
/dir
```

7. Run the `ps` command to get the PID of the new shell, and send a SIGFPE signal (Signal 8) to the new shell by using the `kill` command. (SIGFPE forces a core file.)

```
# ps
PID TTY      TIME CMD
   441 pts/3    0:00 ps
   430 pts/3    0:00 sh
# kill -8 PID
```

Note – The `kill -8` command terminates the shell and the CDE terminal window in which it is executed.

8. In the original terminal window, check to see if a core file exists in the current working directory of the old shell. Use the `file` command to verify that the core file is from the old shell.

```
# cd /dir
# ls
core
# file core
core:           ELF 32-bit MSB core file SPARC Version 1, from 'sh'
```

9. Use the `ls` command to check for a core file in the `/var/core` directory.

```
# ls /var/core
core.sh.430
```

10. Observe the messages generated in the console window and the `/var/adm/messages` file due to `coreadm` logging being enabled.

```
# tail /var/adm/messages
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Configuring NFS

Objectives

The Network File System (NFS) is a client-server application that lets users view, store, and update files on a remote computer as though they were on the their own local computer.

Upon completion of this module, you should be able to:

- Describe the benefits of NFS
- Describe the fundamentals of the NFS distributed file system
- Manage an NFS server
- Manage an NFS client
- Enable the NFS server logging
- Manage NFS with the Solaris™ Management Console storage folder tools
- Troubleshoot NFS errors

The following course map shows how this module fits into the current instructional goal.

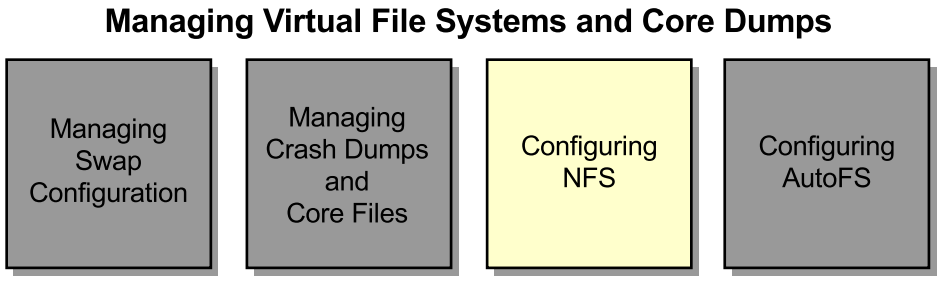


Figure 6-1 Course Map

Introducing the Benefits of NFS

The NFS service enables computers of different architectures running different operating systems to share file systems across a network.

You can implement the NFS environment on different operating environments (OEs) because NFS defines an abstract model of a file system. Each operating system applies the NFS model to its file system semantics. For example, NFS file system operations, such as reading and writing, work as if they were accessing a local file.

Some of the benefits of the NFS service are that it:

- Allows multiple computers to use the same files, because all users on the network can access the same data
- Reduces storage costs by sharing applications on computers instead of allocating local disk space for each user application
- Provides data consistency and reliability, because all users can read the same set of files
- Supports heterogeneous environments, including those found on a personal computer (PC)
- Reduces system administration overhead

Note – The NFS was developed by Sun Microsystems and is recognized as a file server standard. Its protocol uses the Remote Procedure Call (RPC) method of communication between computers on the Internet.

Benefits of Centralized File Access

The NFS service lets you share a whole or partial directory tree or a file hierarchy. Instead of placing copies of commonly used files on every system, the NFS service enables you to place one copy of the files on one computer's hard disk. All other systems can then access the files across the network. When using the NFS service, remote file systems are almost indistinguishable from local file systems.

Note – In most UNIX environments, a file hierarchy that can be shared corresponds to a file system. Because NFS functions across operating systems, and the concept of a file system might be meaningless in non-UNIX environments, the use of the term file system refers to a file hierarchy that can be shared and mounted over NFS environments.

The files are centrally located, making the same files accessible to many users and systems simultaneously. This accessibility feature is useful when giving a user access to a single home directory across multiple systems or when providing access to various applications.

Benefits of Common Software Access

Systems can share one or more centrally located software packages, reducing the disk space requirements for individual systems.

Remote file sharing is transparent to the user and to any application, because these resources appear as if they exist on the local system.

Introducing the Fundamentals of the NFS Distributed File System

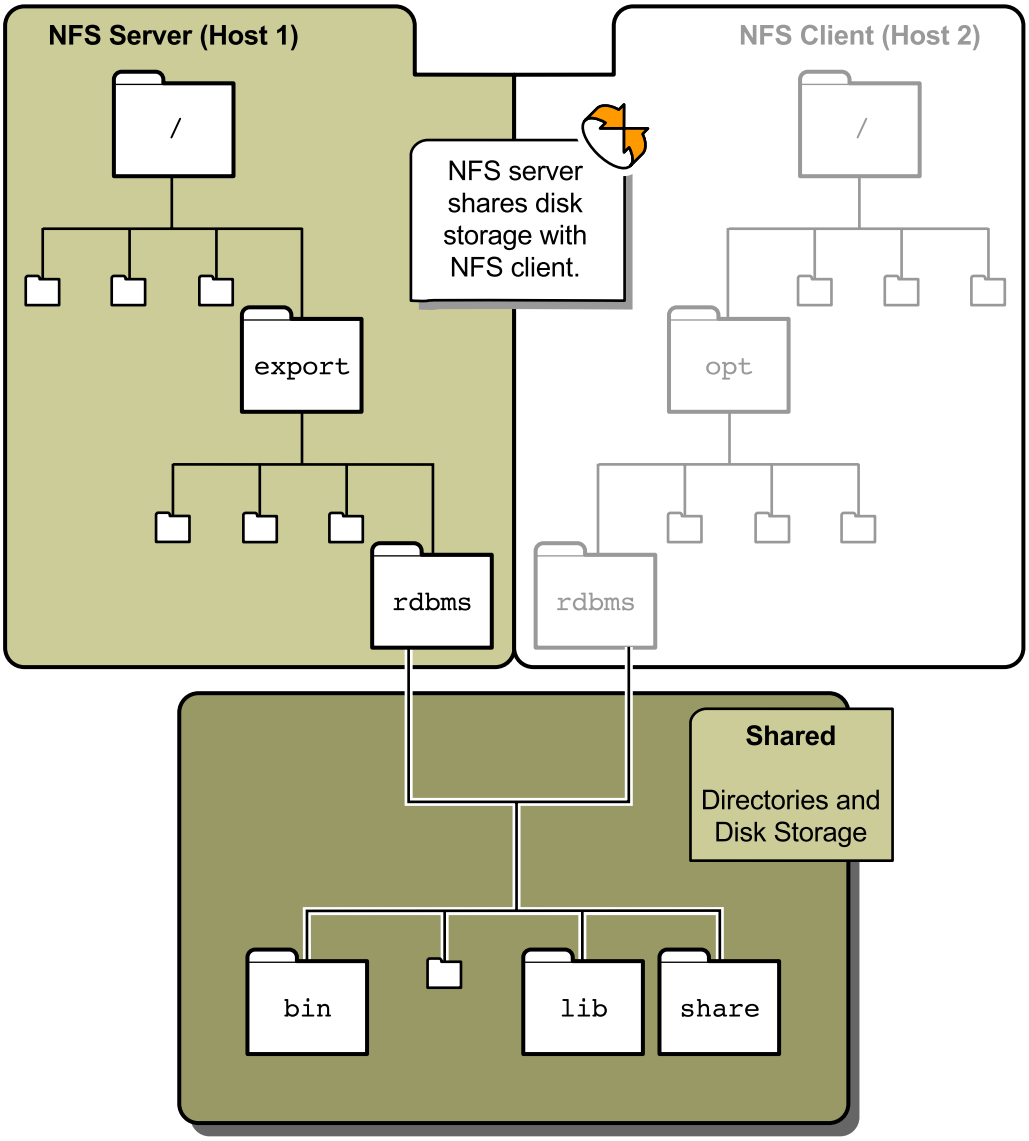
The Solaris 9 OE supports the sharing of remote file resources and presents them to users as if they were local files and directories. The primary distributed file system (DFS) type supported by the Solaris 9 OE is NFS.

The NFS environment contains the following components:

- NFS server
- NFS client

NFS Server

The NFS server contains file resources shared with other systems on the network. A computer acts as a server when it makes files and directories on its hard disk available to the other computers on the network.



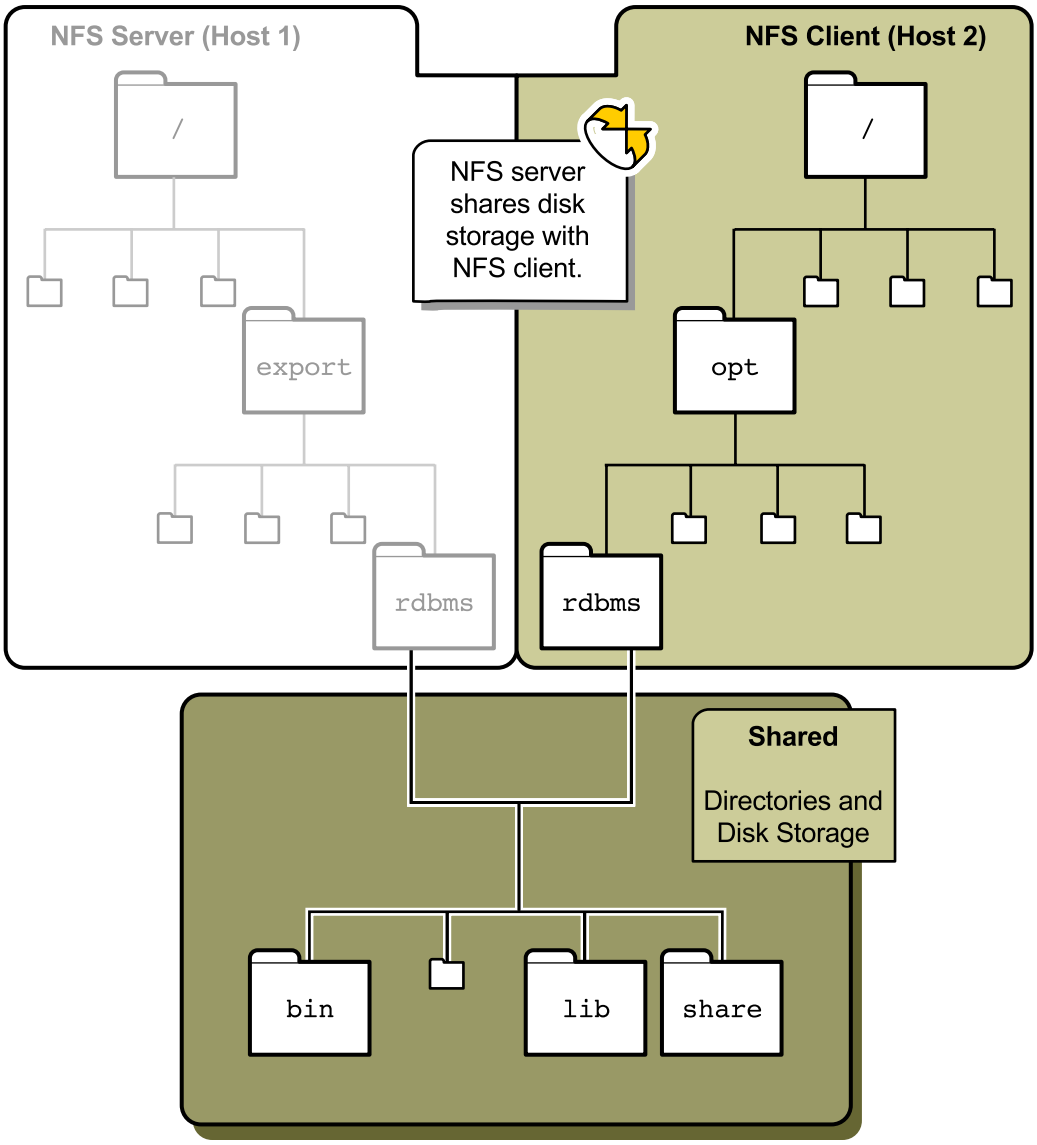

```
Host1# share /export/rdbms
```

Figure 6-2 NFS Server Configuration

Figure 6-2 shows how files and directories on an NFS server are made available to NFS clients. The NFS server is sharing the `/export/rdbms` directory over NFS.

NFS Client

The NFS client system mounts file resources shared over the network and presents the file resources to users as if they were local files.




```
Host2# mount Host1:/export/rdbms /opt/rdbms
```

Figure 6-3 NFS Client Configuration

Figure 6-3 shows how an NFS client uses the files and directories shared by an NFS server. The `/export/rdbms` directory, shared by the NFS server, is mounted on the NFS client on the `/opt/rdbms` mount point. The resource mount point exists on the NFS client, and the NFS server shares the file resources with other computers on the network.

Managing an NFS Server

You use NFS server files, NFS server daemons, and NFS server commands to configure and manage an NFS server.

The NFS Server Files

You need several files to support NFS server activities on any computer. Table 6-1 lists these files and their functions.

Table 6-1 NFS Server Files

File	Description
`/etc/dfs/dfstab`	Lists the local resources to share at boot time.
`/etc/dfs/sharetab`	Lists the local resources currently being shared by the NFS server. Do not edit this file.
`/etc/dfs/fstypes`	Lists the default file system types for remote file systems.
`/etc/rmtab`	Lists file systems remotely mounted by NFS clients. Do not edit this file.
`/etc/nfs/nfslog.conf`	Lists information defining the location of configuration logs used for NFS server logging.
`/etc/default/nfslogd`	Lists configuration information describing the behavior of the `nfslogd` daemon.

The /etc/dfs/dfstab File

The /etc/dfs/dfstab file contains the commands that share local directories. Each line of the dfstab file consists of a share command.

```
# cat /etc/dfs/dfstab

#       Place share(1M) commands here for automatic execution
#       on entering init state 3.
#
#       Issue the command '/etc/init.d/nfs.server start' to run the NFS
#       daemon processes and the share commands, after adding the very
#       first entry to this file.
#
#       share [-F fstype] [ -o options] [-d "<text>"] <pathname>
[resource]
#       .e.g,
#       share  -F nfs  -o rw=engineering  -d "home dirs"  /export/home2

share -o ro /export/sys44_data
```

Note – If the nfs.server script does not find any share commands in the /etc/dfs/dfstab file, it does not start the NFS daemons.

The contents of the /etc/dfs/dfstab file are read when:

- The system enters run level 3.
- The superuser runs the shareall command. The NFS daemons must be running to share directories.
- The superuser runs the /etc/init.d/nfs.server script with the start argument. This script starts the NFS server daemons.

The /etc/dfs/sharetab File

The /etc/dfs/sharetab file stores the results of the share commands. This file contains a table of local resources currently being shared. The following example shows that two nfs resources are shared in read-only mode.

```
# cat /etc/dfs/sharetab
/export/sys44_data      -       nfs     ro
```

The /etc/dfs/fstypes File

The /etc/dfs/fstypes file lists a system's DFS file system types. For each distributed file system type, there is a line beginning with the file system type, which is used with the -F option of the share and mount commands. The file system type listed on the first line of this file is the default file system type when entering DFS administration commands without the -F fstypes option.

```
# cat /etc/dfs/fstypes
nfs NFS Utilities
autofs AUTOFS Utilities
cachefs CACHEFS Utilities
```

The /etc/rmtab File

The /etc/rmtab file contains a table of file systems remotely mounted by NFS clients. After a client successfully completes an NFS mount request, the mountd daemon on the server makes an entry in the /etc/rmtab file. This file also contains a line entry for each remotely mounted directory that has been successfully unmounted, except that the mountd daemon replaces the first character in the entry with the hash character. For example:

```
# The format of this file follows the syntax
# hostname:fsname
sys42:/export/sys44_data
#ys41:/usr/share/man
#ys43:/export/sys44_data
```

The entries for unmounted directories (indicated with the hash mark in the first character of the system name) are removed by the mountd daemon during a system startup. Because stale entries accumulate in the /etc/rmtab file when a client crashes, you should periodically review the contents and delete entries as necessary.

The NFS Server Daemons

You need several daemons to support NFS activities. These daemons can support both NFS client and NFS server activity, NFS server activity alone, or logging of the NFS server activity.

To start the NFS server daemons or to specify the number of concurrent NFS requests that can be handled by the `nfsd` daemon, use the `/etc/rc3.d/S15nfs.server` script.

If a system has entries in its `/etc/dfs/dfstab` file, these server daemons start when the system enters run level 3. Table 6-2 lists the NFS server daemons.

Table 6-2 NFS Server Daemons

Daemon	Description
`mountd`	Handles file system mount requests from remote systems, and provides access control
`nfsd`	Handles client file system requests
`statd`	Works with the `lockd` daemon to provide crash recovery functions for the lock manager
`lockd`	Supports record locking operations on NFS files
`nfslogd`	Provides operational logging

The `mountd` Daemon

The `mountd` daemon handles NFS file system mount requests from remote systems and provides access control. The `mountd` daemon checks the `/etc/dfs/sharetab` file to determine whether a particular file or directory is being shared and whether the requesting client has permission to access the shared resource.

When an NFS client issues an NFS mount request, the `mount` command on the client contacts the `mountd` daemon on the server. The `mountd` daemon provides a file handle to the client.File handles are client references that uniquely identify a file or directory on the server. File handles encode a file's inode number, inode generation number, and disk device number.

The NFS client mount process writes the file handle (along with other information about the mounted resource) to the local `/etc/mnttab` file.

The nfsd Daemon

When a client process attempts to access a remote file resource, the nfsd daemon on the NFS server receives the request and the resource's file handle, and then performs the requested operation. This daemon returns any data to the requesting client process.

The nfsd daemon also handles file system data requests from clients. Only the superuser can start the nfsd daemon. The nfsd daemon is started when the system enters run level 3.

The statd Daemon

The statd daemon works with the lock manager lockd daemon to provide crash recovery functions for the lock manager. The server's statd daemon tracks the clients that are holding locks on an NFS server. When the NFS server reboots after a crash, the statd daemon on the server contacts the statd daemon on the client, which informs the lockd daemon to reclaim any locks on the server. When an NFS client reboots after a crash, the statd daemon on the client system contacts the statd daemon on the server, which invokes the lockd daemon to clear any previous client process locks on the server.

The lockd Daemon

The lockd daemon supports record locking operations for NFS files. The daemon sends locking requests from the NFS client to the NFS server. The server's lockd daemon enables local locking on the NFS server.

The nfslogd Daemon

The nfslogd daemon provides operational logging for an NFS server. NFS logging is enabled when the share is made available. For all file systems for which logging is enabled, the NFS kernel module records all operations in a buffer file. The nfslogd daemon periodically processes the contents of the buffer files to produce American Standard Code for Information Interchange (ASCII) log files, as defined by the contents of the /etc/default/nfslogd file.

The nfslogd daemon also handles the mapping of file handles to path names. The daemon keeps track of these mappings in a *file-handle-to-path* mapping table. After post-processing, the ASCII log files store the records.

Managing the NFS Server Daemons

The NFS daemons start conditionally when the system transitions through the run levels, or they start manually when running the scripts in the `/etc/init.d` directory.

Note – The `nfsd` and `mountd` daemons start if there is an uncommented share statement in the system's `/etc/dfs/dfstab` file.

Starting the NFS Server Daemons

The `/etc/rc3.d/S15nfs.server` script starts the NFS server daemons when the system enters run level 3.

To start the NFS server daemons manually, perform the command:

```
# /etc/init.d/nfs.server start
```

Stopping the NFS Server Daemons

The NFS server daemons are shut down when:

- The system enters run level 2 using the `/etc/rc2.d/K28nfs.server` script
- The system enters run level 1 using the `/etc/rc1.d/K28nfs.server` script
- The system enters run level S using the `/etc/rcS.d/K28nfs.server` script
- The system enters run level 0 using the `/etc/rc0.d/K28nfs.server` script

To stop the NFS server daemons manually, perform the command:

```
# /etc/init.d/nfs.server stop
```

NFS Server Commands

Table 6-3 lists the NFS server commands.

Table 6-3 NFS Server Commands

Commands	Description
`share`	Makes a local directory on an NFS server available for mounting. It also displays the contents of the `/etc/dfs/sharetab` file.
`unshare`	Makes a previously available directory unavailable for client side mount operations.
`shareall`	Reads and executes share statements in the `/etc/dfs/dfstab` file.
`unshareall`	Makes previously shared resources unavailable.
`dfshares`	Lists available shared resources from a remote or local NFS server.
`dfmounts`	Displays a list of NFS server directories that are currently mounted.

Configuringthe NFS Server for Sharing Resources

The following sections describe the basic functionality of the NFS server commands. These commands configure shared remote resources.

Making File Resources Available for NFS Mounting

When the `mountd` and `nfsd` daemons are running, you can use the `share` command to make file resources available:

```
share [ -F nfs ] [ -o options ] [ -d description ] [ pathname ]
```

where:

`-F nfs`	Specifies the file system type. This option is not typically required, because NFS is the default remote file system type.
`-o options`	Controls a client's access to an NFS shared resource.
`-d description`	Describes the shared file resource.
pathname	Specifies the absolute path name of the resource for sharing.

Note – If you do not use the `-F nfs` option with the `share` command, the system uses the file system type from the first line of the `/etc/dfs/fstypes` file.

To share a file resource from the command line, you can use the `share` command. For example, to share the `/export/sys44_data` directory as a read-only shared resource, perform the command:

```
# share -o ro /export/sys44_data
```

By default, NFS-mounted resources are available with read and write privileges based on standard Solaris OE file permissions. Access decisions are based on a comparison of the user ID (UID) of the client and the owner.

The following `share` command options shown in Table 6-4 restrict the read and write capabilities for NFS clients and enable superuser access to a mounted resource.

Table 6-4 The `share` Command Options

Options	Definitions
`ro`	Informs clients that the server accepts only read requests
`rw`	Allows the server to accept read and write requests from the client
`root=`*`client`*	Informs clients that the root user on the specified client system or systems can perform superuser-privileged requests on the shared resource
`ro=`*`access-list`*	Allows read requests from the specified access list
`rw=`*`access-list`*	Allows read and write requests from the specified access list, as shown in Table 6-5

Table 6-5 Access List Options

Option	Description
`access-list=client:client`	Allows access based on a colon-separated list of one or more clients.
`access-list=@network`	Allows access based on a network number (for example, `@192.168.100`) or a network name (for example, `@mynet.com`). The network name must be defined in the `/etc/networks` file.
`access-list=.domain`	Allows access based on a Domain Name System (DNS) domain; the dot (`.`) identifies the value as a DNS domain.
`access-list=netgroup_name`	Allows access based on a configured net group (Network Information Service [NIS] or Network Information Service Plus [NIS+] only).
`anon=`*`n`*	Sets *`n`* to be the effective user ID (EUID) of unknown users. By default, unknown users are given the EUID 60001 (`UID_NOBODY`). If *`n`* is set to `-1`, access is denied.

You can combine these options by separating each option with commas, which forms intricate access restrictions. The following examples show some of the more commonly used options:

```
# share -F nfs -o ro directory
```

This command restricts access to NFS-mounted resources to read-only access.

```
# share -F nfs -o ro,rw=client1 directory
```

This command restricts access to NFS-mounted resources to read-only access; however, the NFS server accepts both read and write requests from the client named `client1`.

```
# share -F nfs -o root=client2 directory
```

This command allows the `root` user on the client named `client2` to have superuser access to the NFS-mounted resources.

```
# share -F nfs -o anon=0 directory
```

By setting the option `anon=0`, the EUID for access to shared resources is set to the UID of the user who is accessing the shared resource.

The `share` command writes information for all shared file resources to the `/etc/dfs/sharetab` file. The file contains a table of the local shared resources.

Note – If no argument is specified, the `share` command displays a list of all the currently shared file resources.

```
# share
-               /export/sys44_data   ro   ""
```

Making File Resources Unavailable for Mounting

Use the `unshare` command to make file resources unavailable for mount operations. This command reads the `/etc/dfs/sharetab` file.

```
unshare [ -F nfs ] pathname
```

where:

`-F nfs`	Specifies NFS as the file system type. Because NFS is the default remote file system type, you do not have to specify this option.
`pathname`	Specifies the path name of the file resource to unshare.

For example, to make the `/export/sys44_data` directory unavailable for client-side mount operations, perform the command:

```
# unshare /export/sys44_data
```

Sharing and Unsharing All NFS Resources

Use the `shareall` and `unshareall` commands to share and unshare all NFS resources.

The `shareall` command, when used without arguments, shares all resources listed in the `/etc/dfs/dfstab` file.

```
shareall [ -F nfs ]
```

The `unshareall` command, when used without arguments, unshares currently shared file resources listed in the `/etc/dfs/sharetab` file.

```
unshareall [ -F nfs ]
```

Displaying Currently Shared NFS Resources

The `dfshares` command uses the NFS daemons, `mountd` and `nfsd`, to display currently shared NFS resources.

```
dfshares [ -F nfs ] [ host ]
```

The `dfshares` command displays resources currently being shared by the local server when used without a *`host`* argument.

```
# share -F nfs -o ro /export/sys44_data
# dfshares
RESOURCE                                  SERVER ACCESS    TRANSPORT
     sys44:/export/sys44_data              sys44  -          -
```

By specifying one or more server names as arguments, the `dfshares` command also displays file resources being shared by other servers. For example:

```
# dfshares sys42
RESOURCE                                  SERVER ACCESS    TRANSPORT
     sys42:/export/sys42_eng_data          sys42  -          -
```

Displaying NFS Mounted Resources

The `dfmounts` command displays remotely mounted NFS resource information.

```
dfmounts [ -F nfs ] [ server ]
```

The `dfmounts` command, when used without arguments, displays a list of directories on the local server that are currently mounted and also displays a list of the client systems that currently have the shared resource mounted.

```
# dfmounts
RESOURCE     SERVER PATHNAME                 CLIENTS
  -           sys44 /export/sys44_data       sys41,sys42
```

By specifying one or more server names on the command line, the `dfmounts` command can also display a list of clients currently mounted (shared directories) on other NFS servers. For example:

```
# dfmounts sys42
RESOURCE     SERVER PATHNAME                  CLIENTS
  -           sys42 /export/sys42_eng_data   sys41
```

Managing the NFS Client

NFS client files, NFS client daemons, and NFS client commands work together to manage the NFS client.

NFS Client Files

You need several files to support NFS client activities on any computer. Table 6-6 lists the files that support NFS client activities.

Table 6-6 NFS Client Files

File	Description
`/etc/vfstab`	Defines file systems to be mounted locally.
`/etc/mnttab`	Lists currently mounted file systems, including automounted directories. The contents of this file are maintained by the `kernel` and cannot be edited.
`/etc/dfs/fstypes`	Lists the default file system types for remote file systems.

The `/etc/vfstab` File

To mount remote file resources at boot time, enter the appropriate entries in the client's `/etc/vfstab` file. For example:

```
#device             device       mount            FS    fsck  mount    mount
#to mount           to fsck      point            type  pass  at boot  options
#
sys44:/export/sys44_data    -   /export/remote_data   nfs   -     yes    soft,bg
```

The /etc/mnttab File

The /etc/mnttab file system provides read-only access to the table of mounted file systems for the current host. Mounting a file system adds an entry to the table of mounted file systems. Unmounting a file system removes an entry from the table of mounted file systems.

Remounting a file system updates the information in the mounted file system table. The kernel maintains a chronological list in the order of the mount time. The first mounted file system is first on the list and the most recently mounted file system is last. Although the /etc/mnttab file is a mount point for the mntfs file system, it appears as a regular file containing the current mount table information. The /etc/rcS.d/s70buildmnttab.sh script establishes the mntfs file system during the boot process.

The /etc/dfs/fstypes File

As with an NFS server, NFS clients use the /etc/dfs/fstypes file to determine distributed file system support.

```
# cat /etc/dfs/fstypes
nfs NFS Utilities
autofs AUTOFS Utilities
cachefs CACHEFS Utilities
```

NFS Client Daemons

The NFS client daemons are started using the /etc/rc2.d/S73nfs.client script. Table 6-7 lists the NFS client daemons.

Table 6-7 NFS Client Daemons

Daemon	Description
statd	Works with the lockd daemon to provide crash recovery functions for the lock manager
lockd	Supports record-locking operations on NFS files

Managing the NFS Client Daemons

Two NFS daemons, the `statd` daemon and the `lockd` daemon, run both on the NFS servers and the NFS clients. These daemons start automatically when a system enters run level 2.

Both the `statd` and `lockd` daemons provide crash recovery and locking services for NFS. If a server crashes, clients can quickly re-establish connections with files they were using. Therefore, the server has a record of the clients that were using its NFS resources. It contacts each client for information about which files were in use, which helps to provide continuous operation. You can start both these daemons from the `/etc/init.d/nfs.client` script. Neither daemon requires administrative intervention.

Starting the NFS Client Daemons

The `/etc/rc2.d/S73nfs.client` script automatically starts the NFS client daemons when the system enters run level 2.

To manually start these daemons, perform the command:

```
# /etc/init.d/nfs.client start
```

Stopping the NFS Client Daemons

The `/etc/rc0.d/k41nfs.client` script automatically shuts down NFS client daemons when the system enters init run level 0.

To manually stop these daemons, perform the command:

```
# /etc/init.d/nfs.client stop
```

NFS Client Commands

Table 6-8 lists the NFS client commands.

Table 6-8 NFS Client Commands

Command	Description
`dfshares`	Lists available shared resources from a remote or local NFS server
`mount`	Attaches a file resource (local or remote) to a specified local mount point
`umount`	Unmounts a currently mounted file resource
`mountall`	Mounts all file resources or a specific group of file resources listed in the `/etc/vfstab` file with a `mount at boot` value of `yes`
`umountall`	Unmounts all non-critical local and remote file resources
`dfmounts`	Displays a list of currently mounted NFS server directories

Configuring the NFS Client for Mounting Resources

The following sections describe some of the functions of the NFS client utilities.

Displaying a Server's Available Resources

You can use the `dfshares` command to list resources made available by an NFS server. To verify the resources that an NFS server is currently making available, run the `dfshares` command with the server name as an argument.

```
# dfshares sys44
RESOURCE                                  SERVER ACCESS    TRANSPORT
     sys44:/export/sys44_data              sys44  -         -
```

Accessing the Remote File Resource

Enter the `/usr/sbin/mount` command to attach a local or remote file resource to the file system hierarchy.

```
mount [ -F nfs ] [ -o options ] server:pathname mount_point
```

where:

`-F nfs`	Specifies NFS as the file system type. The `-F nfs` option is not necessary, because NFS is the default remote file system type specified in the `/etc/dfs/fstypes` file.
`-o` *`options`*	Specifies a comma-separated list of file-system specific options, such as `rw`. The `rw` option mounts the file resource as read, write. The `ro` option mounts the file resource as read-only. (The default is `rw`.)
`server:pathname`	Specifies the name of the server and the path name of the remote file resource. The names of the server and the path name are separated by a colon (:).
`mount_point`	Specifies the path name of the mount point on the local system (which must already exist).

Use the `mount` command to access a remote file resource. For example:

```
# mount sys44:/export/sys44_data /export/remote_data
```

When mounting a read-only remote resource, you can specify a comma-separated list of sources for the remote resource, which are then used as a list of failover resources. This process works if the resource mounted from all of the servers in the list is the same. For example:

```
# mount -o ro sys45,sys43,sys41:/multi_homed_data /remote_shared_data
```

In this example, if the server `sys45` is unavailable, the request passes to the next server on the list (`sys43`) and then to the server `sys41`.

Unmounting the Remote File Resources From the Client

Use the `umount` command to detach local and remote file resources from the file system hierarchy. This command reads the `/etc/mnttab` file on the client.

```
umount server:pathname | mount_point
```

The command can specify either the *`server:pathname`* option or the *`mount_point`* option.

```
#  umount /export/remote_data
```

Mounting All File Resources

Without any arguments, the `/usr/sbin/mountall` command mounts all file resources listed in the `/etc/vfstab` file with a `mount at boot` value of `yes`.

To limit the action of this command to remote file resources, use the `-r` option.

```
mountall -r [ -F nfs ]
```

```
#  mountall -r
```

Unmounting All Currently Mounted File Resources

When you use the `umountall` command without any arguments, it unmounts all currently mounted file resources except for the root (`/`), `/usr`, `/var`, `/var/adm`, `/var/run`, `/proc`, and `/dev/fd` directories. To restrict the unmounting to only remote file systems, use the `-r` option.

```
umountall -r [ -F nfs ]
#  umountall -r
```

Note – Use the `-F FSType` with the `mountall` and `umountall` commands to specify `FSType` as the file system type. You do not have to specify the `-F nfs` option, because NFS is listed as the default remote file system type.

Mounting Remote Resources at Boot Time

To mount the remote file resources at boot time, enter the appropriate entries in the client's `/etc/vfstab` file. For example:

```
#device              device       mount             FS    fsck  mount    mount
#to mount            to fsck      point             type  pass  at boot  options
#
sys44:/export/sys44_data    -   /export/remote_data   nfs   -     yes      soft,bg
```

where the fields in the `/etc/vfstab` file are:

`device to mount`	The name of the server and the path name of the remote file resource. The server host name and share name are separated by a colon (`:`).
`device to fsck`	NFS resources are not checked by the client because the file system is not local to the client; therefore, this field is always dash (`-`) for NFS resources.
`mount point`	The mount point for the resource.
`FS type`	This field specifies the type of file system to be mounted.
`fsck pass`	NFS resources are not checked by the client, because the file system is not local to the client. Therefore, this field is always dash (`-`) for NFS resources.
`mount at boot`	This field can contain either of two values, `yes` or `no`. If the field is set to the value `yes`, the specified resource is mounted every time the `mountall` command is run.
`mount options`	A comma-separated list of mount options. See Table 6-9 on page 6-27 a description of each option.

Note – If the `/etc/vfstab` file contains the file resource, the superuser can specify either *`server`*`:`*`pathname`* or *`mount_point`* on the command line, because the `mount` command checks the `/etc/vfstab` file for more information.

Table 6-9 The `mount` Command Options

Option	Description
`rw|ro`	Specifies whether the resource is mounted as read/write or read-only. The default is read/write.
`bg|fg`	During an NFS mount request, if the first mount attempt fails, retry in the background or foreground. The default is to retry in the foreground.
`soft|hard`	When the number of retransmissions has reached the number specified in the `retrans=n` option, a file system mounted with the `soft` option reports an error on the request, and stops trying. A file system mounted with the `hard` option prints a warning message and continues to try to process the request. The default is a `hard` mount. Although the `soft` option and the `bg` option are not the default settings, combining them usually results in the fastest client boot when NFS mounting problems occur.
`intr|nointr`	Enables or disables the use of keyboard interrupts to kill a process that hangs while waiting for a response on a hard-mounted file system. The default is `intr`.
`suid|nosuid`	Indicates whether to enable `setuid` execution. The default enables `setuid` execution.
`timeo=`*n*	Sets the timeout to *n* tenths of a second. The default timeout is 11, measured in one-tenth of a second (0.1 second) for User Datagram Protocol (UDP) transports, and 600 tenths of a second for Transmission Control Protocol (TCP).
`retry=`*n*	Sets the number of times to retry the mount operation. The default is 10,000 times.
`retrans=`*n*	Sets the number of NFS retransmissions to *n*. The default is 5 for UDP. For the connection-oriented TCP, this option has no effect.

Enabling the NFS Server Logging

Maintain an NFS activity log to:

- Track remote file accesses on your network
- Assist in debugging NFS failures

Fundamentals of NFS Server Logging

The NFS server logging feature records NFS transactions on the file system. The `nfslogd` daemon provides operational logging.

When you enable NFS server logging, the NFS `kernel` module writes records of all NFS operations on the file system into a buffer file. The data includes a time stamp, the client IP address, the UID of the requester, the file handle of the resource being accessed, and the type of operation that occurs.

The `nfslogd` Daemon

The functions of the `nfslogd` daemon are that it:

- Converts the raw data from the logging operation into ASCII records, and stores the raw data in ASCII log files.
- Resolves IP addresses to host names and UIDs to login names.
- Maps the file handles to path names, and records the mappings in a file-handle-to-path mapping table. Each tag in the `/etc/nfs/nfslog.conf` file corresponds to one mapping table.

Note – Keep the `nfslogd` daemon running. If the `nfslogd` daemon is not running, changes are not tracked to the mappings in the file-handle-to-path table.

Configuring NFS Log Paths

The `/etc/nfs/nfslog.conf` file defines the path, file names, and type of logging that the `nfslogd` daemon must use. There is a tag corresponding to each definition.

To configure NFS server logging, identify or create the tag entries for each of the server's shared resources. The `global` tag defines the default values.

The following is an example an `nfslog.conf` file:

```
# cat /etc/nfs/nfslog.conf
#ident  "@(#)nfslog.conf        1.5     99/02/21 SMI"
#
# Copyright (c) 1999 by Sun Microsystems, Inc.
# All rights reserved.
#
# NFS server log configuration file.
#
# <tag> [ defaultdir=<dir_path> ] \
#       [ log=<logfile_path> ] [ fhtable=<table_path> ] \
#       [ buffer=<bufferfile_path> ] [ logformat=basic|extended ]
#

global  defaultdir=/var/nfs \
        log=nfslog fhtable=fhtable buffer=nfslog_workbuffer
```

Use the following parameters with each tag, as needed:

`defaultdir=`*`dir_path`*	Specifies the default parent directory. All relative path entries to this log can be seen.
`log=`*`logfile_path`*	Specifies the relative or absolute path and the file name for the ASCII log file.
`fhtable=`*`table_path`*	Specifies relative or absolute path and the file name for the file-handle-to-path database file.
`buffer=`*`bufferfile_path`*	Specifies the relative and absolute path and the file name for the raw buffer file.
`logformat=basic/`*`extended`*	Specifies the format when creating user-readable log files. The basic format produces a log file similar to the FTPdaemon. The extended format gives a more detailed view.

If you do not specify an absolute path in the parameters, the `nfslogd` daemon appends the name given to the path specified by the `defaultdir` parameter. To override the value specified by the `defaultdir` parameter, use an absolute path.

To easily identify the log files for different shared resources, place them in separate directories. For example:

```
# cat /etc/nfs/nfslog.conf
#ident  "@(#)nfslog.conf        1.5     99/02/21 SMI"
#
   .
   .
# NFS server log configuration file.
#
global defaultdir=/var/nfs \
         log=nfslog fhtable=fhtable buffer=nfslog_workbuffer
public defaultdir=/var/nfs/public \
         log=nfslog fhtable=fhtable buffer=nfslog_workbuffer
```

Create the `/var/nfs/public` directory before starting NFS server logging.

In the previous example, any file system shared with `log=public` uses the following values:

- The default directory is the `/var/nfs/public` directory.
- The log is stored in the `/var/nfs/public/nfslog` file.
- The `/var/nfs/public/fhtables` file stores the *`file-handle-to-path`* database.
- The `/var/nfs/public/nfslog_workbuffer` file stores the buffer.

Initiating NFS Logging

To initiate NFS server logging, complete the following steps:

1. Become superuser.
2. Optional: Change the file system configuration settings. In the `/etc/nfs/nfslog.conf` file, either:
   - Edit the default settings for all file systems by changing the data corresponding to the `global` tag.
   - Add a new tag for the specific file system.

   If you do not need these changes, do not edit this file.
3. To share file systems using NFS server logging, you must first enable NFS server logging. Edit the `/etc/dfs/dfstab` file to add an entry for file systems for which you want to enable NFS server logging. Either:
   - Specify a tag by entering the tag to use with the `log=`*`tag`* option in the `/etc/dfs/dfstab` file.
   - Use the `log` option without specifying a tag, which causes the option to use the `global` tag as a default. The following example uses the default settings in the `global` tag:

```
share -F nfs -o ro,log /export/sys44_data
```

4. Check that the NFS service is running on the server.

   To start or restart the `mountd`, `nfsd`, and `nfslogd` daemons if they are not running, perform the command:

```
# /etc/init.d/nfs.server start
```

   If the `/etc/nfs/nfslog.conf` file exists and you execute the `nfs.server` script, the `nfs.server` script starts the `nfslogd` daemon.

5. Run the `share` command to verify that the correct options are listed.

```
# share
-               /export/sys44_data   ro,log   ""
```

6. If you add the additional entries to the `/etc/dfs/dfstab` file, share the file system by rebooting the system or entering the `shareall` command.

```
# shareall
```

Configuring the nfslogd Daemon Behavior

The configuration information in the /etc/default/nfslogd file controls the logging behavior of the nfslogd daemon.

The /etc/default/nfslogd file defines default parameters used for NFS server logging. Table 6-10 describes some of the NFS logging parameters.

Table 6-10 NFS Logging Parameters

Parameter	Description
IDLE_TIME	Sets the amount of time that the nfslogd daemon sleeps before checking the buffer file for more information. It also determines how often the configuration file is checked. The default value is 300 seconds. Increasing this number can improve performance by reducing the number of checks.
MIN_PROCESSING_SIZE	Sets the minimum number of bytes that the buffer file must reach before processing and writing to the log file. The default value is 524,288 bytes. Increasing this number can improve performance by reducing the number of times that the buffer file is processed. The MIN_PROCESSING_SIZE and the IDLE_TIME parameters determine how often the buffer file is processed.
UMASK	Specifies the permissions for the log files set by the nfslogd daemon. The default value is 0137.
CYCLE_FREQUENCY	Determines the time that must pass before the log files are cleared. The default value is 24 hours. Use the CYCLE_FREQUENCY parameter to prevent the log files from becoming too large.
MAX_LOGS_PRESERVE	Determines the number of log files to save. The default value is 10.

Managing NFS With the Solaris Management Console Storage Folder Tools

You can manage the NFS system by using components of the storage folder tools from the default tool box of the Solaris Management Console. The Mounts and Shares tool lets you view, create, and manage several types of mounts and shares. This module uses the following terms:

- A share refers to making a directory on one computer available to other computers.
- A mount is the act of connecting a file or a directory to a shared directory.

Adding a Shared Directory on the NFS Server

Using the Solaris Management Console, you can share a directory to the network.

To add a shared directory on the NFS server, complete the following steps:

1. Open the Solaris Management Console on the NFS server.

Note – The following steps display the contents of the Shared folder within the Mounts and Shares tool.

2. Click the turner icon to display the default toolbox called This Computer (*nfs_servername*).
3. Click the turner icon to display the Storage folder.

Note – When you access a tool for the first time, after opening the Solaris Management Console, log in to the Solaris Management Console to authenticate your access rights.

4. Click the turner icon to display the Mounts and Shares tool.

5. Click the Shares icon to display the currently shared resources from the Shares folder.

   The Shared folder opens. The remaining steps add a shared directory to the list of shared resources.

6. To start the Add Shared Directory wizard, select Add Shared Directory from the Action menu.

7. To specify the directory name select one of the following options:

   - Enter the name of the shared resource in the Directory location.
   - Enter a description of the resource in the Description location.
   - Configure the sharing options as follows:

     1. Share this directory only, or share this directory and its subdirectories.
     2. Share this directory at each boot, or share this directory according to the current demand.

8. To specify how to access the directory, complete the following steps:

   a. Select Basic to set read or read/write permissions for all users and systems that access the shared directory.

   b. Select Advanced to further define authentication methods. Refer to the Help Index feature on the Solaris Management Console to define the authentication methods.

9. Specify the directory access as either read/write or read-only.

10. Review your shared directory selections:

    a. To make any changes in your selections, click Back to back up and modify an entry.

    b. If you are satisfied with your selections, click Finish to create the shared directory.

11. Return to the Solaris Management Console Shared directories folder, which displays the new shared directory.

    You can now access the shared directory through NFS mounts.

Mounting a Shared Directory on the NFS Client

To mount a shared directory on the NFS client, complete the following steps:

1. Open the Solaris Management Console on the NFS client.

Note – The following steps display the contents of the Mounts folder within the Mounts and Shares tool.

2. Click the turner icon to display the default toolbox that is labeled as This Computer (*nfs_clientname*).
3. Click the turner icon to display the Storage folder.
4. Click the Mount and Share icon to display the Mounts and Shares tool.
5. Click the turner icon to display the Mounts and Shares tool.
6. Click the Mount icon to display the currently mounted resources in the Mounts folder.

   The Mounts folder opens. The remaining steps add an NFS mounted directory to the list of mounted resources.
7. To start the Add NFS Mount wizard, select the Add NFS Mount field from the Action menu.
8. To identify the computer sharing the directory, enter the name of the NFS server in the Computer field.
9. To specify the mount point, enter the name of the NFS client directory that will contain the contents of the shared directory.

   If the mount point directory does not exist on the NFS client, you must create it.
10. Specify whether to mount the directory at boot time or to manually mount the directory before trying to access it.
11. Specify the kind of directory access as either read/write or read-only.

Note – Access rights for the NFS client mounts cannot exceed the access rights defined on the NFS server for that shared resource.

12. Review your NFS mount selections:
    a. To make any changes in your selections, click Back to back up and modify an entry.
    b. If you are satisfied with your selections, click Finish to add the NFS mount point.
13. Select the Solaris Management Console Mounts folder, which displays the newly created mount point.

    You can now access the NFS mounted directory in the same way as you would access the local file systems.

Troubleshooting NFS Errors

You can detect most NFS problems from console messages or from certain symptoms that appear on a client system. Some common errors are:

- The `rpcbind failure` error
- The `server not responding` error
- The `NFS client fails a reboot` error
- The `service not responding` error
- The `program not registered` error
- The `stale file handle` error
- The `unknown host` error
- The `mount point` error
- The `no such file` error

The `rpcbind failure` Error

The following example shows the message that appears on the client system during the boot process or in response to an explicit mount request.

```
nfs mount: server1:: RPC: Rpcbind failure
RPC: Timed Out
nfs mount: retrying: /mntpoint
```

The error in accessing the server is due to:

- The combination of an incorrect Internet address and a correct host or node name in the `hosts` database file supporting the client node.
- The `hosts` database file that supports the client has the correct server node, but the server node temporarily stops due to an overload.

To solve the `rpcbind failure` error condition when the server node is operational, determine if the server is out of critical resources (for example, memory, swap, or disk space).

The server not responding Error

The following message appears during the boot process or in response to an explicit mount request, and this message indicates a known server that is inaccessible.

```
NFS server server2 not responding, still trying
```

Possible causes for the server not responding error are:

- The network between the local system and the server is down. To verify that the network is down, enter the ping command (ping *server2*).
- The server (*server2*) is down.

The NFS client fails a reboot Error

If you attempt to boot an NFS client and the client-node stops, waits, and echoes the following message:

```
Setting default interface for multicast: add net 224.0.0.0: gateway:
client_node_name.
```

these symptoms might indicate that a client is requesting an NFS mount using an entry in the /etc/vfstab file, specifying a foreground mount from a non-operational NFS server.

To solve this error, complete the following steps:

1. To interrupt the failed client node press Stop-A, and boot the client into single-user mode.
2. Edit the /etc/vfstab file to comment out the NFS mounts.
3. To continue booting to the default run level (normally run level 3), press Control-D.
4. Determine if all the NFS servers are operational and functioning properly.
5. After you resolve problems with the NFS servers, remove the comments from the /etc/vfstab file.

Note – If the NFS server is not available, an alternative to commenting out the entry in the `/etc/vfstab` file is to use the `bg` mount option so that the boot sequence can proceed in parallel with the attempt to perform the NFS mount.

The `service not responding` Error

The following message appears during the boot process or in response to an explicit mount request, and indicates that an accessible server is not running the NFS server daemons.

```
nfs mount: dbserver: NFS: Service not responding
nfs mount: retrying: /mntpoint
```

To solve the `service not responding` error condition, complete the following steps:

1. Enter the `who -r` command on the server to see if it is at run level 3. If the server is not, change to run level 3 by entering the `init 3` command.
2. Enter the `ps -e` command on the server to check whether the NFS server daemons are running. If they are not, start them by using the `/etc/init.d/nfs.server start` script.

The `program not registered` Error

The following message appears during the boot process or in response to an explicit mount request and indicates that an accessible server is not running the `mountd` daemon.

```
nfs mount: dbserver: RPC: Program not registered
nfs mount: retrying: /mntpoint
```

To solve the `program not registered` error condition, complete the following steps:

1. Enter the `who -r` command on the server to check that it is at run level 3. If the server is not, change to run level 3 by performing the `init 3` command.
2. Enter the `pgrep -xl mountd` command. If the `mountd` daemon is not running, start it using the `/etc/init.d/nfs.server` script, first with the `stop` flag and then with the `start` flag.
3. Check the `/etc/dfs/dfstab` file entries.

The stale NFS file handle Error

The following message appears when a process attempts to access a remote file resource with an out-of-date file handle.

```
stale NFS file handle
```

A possible cause for the stale NFS file handle error is that the file resource on the server moved. To solve the stale NFS file handle error condition, unmount and mount the resource again on the client.

The unknown host Error

The following message indicates that the host name of the server on the client is missing from the hosts table.

```
nfs mount: sserver1:: RPC: Unknown host
```

To solve the unknown host error condition, verify the host name in the hosts database that supports the client node.

Note – The preceding example misspelled the node name server1 as sserver1.

The mount point Error

The following message appears during the boot process or in response to an explicit mount request and indicates a non-existent mount point.

```
mount: mount-point /DS9 does not exist.
```

To solve the mount point error condition, check that the mount point exists on the client. Check the spelling of the mount point on the command line or in the /etc/vfstab file on the client, or comment out the entry and reboot the system.

The no such file Error

The following message appears during the boot process or in response to an explicit mount request, which indicates that there is an unknown file resource name on the server.

```
No such file or directory
```

To solve the no such file error condition, check that the directory exists on the server. Check the spelling of the directory on the command line or in the /etc/vfstab file.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Configuring NFS (Level 1)

In this exercise, you configure an NFS server and client to share and mount the `/usr/share/man` file.

Preparation

Choose a partner for this lab. Determine which systems to configure as the NFS server and the NFS client. Verify that entries for both systems exist in the `/etc/hosts` file on both systems. Refer to your lecture notes as necessary to perform the following steps.

Tasks

Perform the following tasks:

- Select a system to act as an NFS server, and share the `/usr/share/man` directory. Perform the commands to verify that the directory is shared and that no NFS system mounts are present on the server:
  - `share`
  - `dfshares`
  - `dfmounts`
- On the NFS client system, rename the `/usr/share/man` directory to the `/usr/share/man.orig` directory. Make sure the man pages are not available. Create a `/usr/share/man` mount point. Mount the `/usr/share/man` directory from the NFS server. Verify that the man pages are available.
- On the NFS client, record the default options used for the NFS mount. Verify the list of mounts that the server provides. Unmount the `/usr/share/man` file, and verify the list of remote mounts the server is providing.
- On the NFS server, unshare the `/usr/share/man` directory. In the `/etc/dfs/dfstab` file, change the entry for this directory so that it uses the `-o rw=bogus` options. Share everything listed in the `dfstab` file.
- On the NFS client, attempt to mount the `/usr/share/man` directory from the NFS server. Record your observations.

- On the NFS server, unshare the `/usr/share/man` directory, and remove the entry for it from the `/etc/dfs/dfstab` file.
- On the NFS client, return the `/usr/share/man` directory to its original configuration.

Exercise: Configuring NFS (Level 2)

In this exercise, you configure an NFS server and client to share and mount the `/usr/share/man` file.

Preparation

Choose a partner for this lab. Determine which systems to configure as the NFS server and the NFS client. Verify that entries for both systems exist in the `/etc/hosts` file on both systems. Refer to your lecture notes as necessary to perform the following steps.

Task Summary

Perform the following tasks:

- Select a system to act as an NFS server, and share the `/usr/share/man` directory. Perform the commands to verify that the directory is shared and that no NFS system mounts are present on the server:
  - `share`
  - `dfshares`
  - `dfmounts`
- On the NFS client system, rename the `/usr/share/man` directory to the `/usr/share/man.orig` directory. Make sure the man pages are not available. Create a `/usr/share/man` mount point. Mount the `/usr/share/man` directory from the NFS server. Verify that the man pages are available.
- On the NFS client, record the default options used for the NFS mount. Verify the list of mounts that the server provides. Unmount the `/usr/share/man` file, and verify the list of remote mounts the server is providing.
- On the NFS server, unshare the `/usr/share/man` directory. In the `/etc/dfs/dfstab` file, change the entry for this directory so that it uses the `-o rw=bogus` options. Share everything listed in the `dfstab` file.
- On the NFS client, attempt to mount the `/usr/share/man` directory from the NFS server. Record your observations.

- On the NFS server, unshare the `/usr/share/man` directory, and remove the entry for it from the `/etc/dfs/dfstab` file.
- On the NFS client, return the `/usr/share/man` directory to its original configuration.

Tasks

Complete the following tasks.

Task 1– On the NFS Server

Complete the following steps:

1. Edit the `/etc/dfs/dfstab` file. Add an entry to share the directory that holds man pages.

   _____

2. Stop and start the NFS server daemons.

   _____

3. Verify that the `/usr/share/man` directory is shared and that no NFS mounts are present.

   _____

Task 2 – On the NFS Client

Complete the following steps:

1. Rename the `/usr/share/man` directory so that you can no longer access the man pages on the client system. Verify that the man pages are not available.

   _____

   What message does the `man` command report?

   _____

2. Create a new `man` directory (`/usr/share/man`) to use as a mount point.

   _____

3. Mount the `/usr/share/man` directory from the server.

4. Verify that the man pages are available.

   _____

   Are the man pages available?

   _____

5. Verify and record the default `ro | rw` options used for this mount.

   _____

6. Write a file into the NFS-mounted file system.

   _____

   What is the result of trying to write to the NFS-mounted file system?

   _____

   What conclusion can be reached by this exercise?

   _____

   _____

   _____

7. Verify the list of mounts that the server provides.

   _____

8. Unmount the `/usr/share/man` directory, and verify the list of remote mounts from the server.

Task 3 – On the NFS Server

Complete the following steps:

1. Unshare the `/usr/share/man` directory.

   _____

2. Change the `share` statement in the `/etc/dfs/dfstab` file for the `/usr/share/man` directory to read:

```
share -o ro=bogus /usr/share/man
```

3. Share the `/usr/share/man` directory.

   _____

Task 4 – On the NFS Client

Complete the following step:

Attempt to mount the `/usr/share/man` directory again.

_____

What happens?

_____

Task 5 – On the NFS Server

Complete the following steps:

1. Unshare the `/usr/share/man` directory.

   _____

2. Edit the `/etc/dfs/dfstab` file to remove the entry for the `/usr/share/man` directory.

   _____

Task 6 – On the NFS Client

Complete the following steps:

1. Return the `/usr/share/man` directory to its original configuration.

   _____

2. Verify that the man pages are now available.

   _____

Exercise: Configuring NFS (Level 3)

In this exercise, you configure an NFS server and client to share and mount the `/usr/share/man` file.

Preparation

Choose a partner for this lab. Determine which systems to configure as the NFS server and the NFS client. Verify that entries for both systems exist in the `/etc/hosts` file on both systems. Refer to your lecture notes as necessary to perform the following steps.

Task Summary

Perform the following tasks:

- Select a system to act as an NFS server, and share the `/usr/share/man` directory. Perform the commands to verify that the directory is shared and that no NFS system mounts are present on the server:
  - `share`
  - `dfshares`
  - `dfmounts`
- On the NFS client system, rename the `/usr/share/man` directory to the `/usr/share/man.orig` directory. Make sure the man pages are not available. Create a `/usr/share/man` mount point. Mount the `/usr/share/man` directory from the NFS server. Verify that the man pages are available.
- On the NFS client, record the default options used for the NFS mount. Verify the list of mounts that the server provides. Unmount the `/usr/share/man` file, and verify the list of remote mounts the server is providing.
- On the NFS server, unshare the `/usr/share/man` directory. In the `/etc/dfs/dfstab` file, change the entry for this directory so that it uses the `-o rw=bogus` options. Share everything listed in the `dfstab` file.
- On the NFS client, attempt to mount the `/usr/share/man` directory from the NFS server. Record your observations.

- On the NFS server, unshare the `/usr/share/man` directory, and remove the entry for it from the `/etc/dfs/dfstab` file.
- On the NFS client, return the `/usr/share/man` directory to its original configuration.

Tasks and Solutions

Complete the following tasks.

Task 1– On the NFS Server

Complete the following steps:

1. Edit the `/etc/dfs/dfstab` file. Add an entry to share the directory that holds man pages.

```
share -o ro /usr/share/man
```

2. Stop and start the NFS server daemons.

```
# /etc/init.d/nfs.server start
```

3. Verify that the `/usr/share/man` directory is shared and that no NFS mounts are present.

```
# share
-  /usr/share/man  ro  ""

# dfshares
RESOURCE                        SERVER  ACCESS    TRANSPORT
server:/usr/share/man           server   -           -

# dfmounts
```

There is no output for the `dfmounts` *command.*

Task 2 – On the NFS Client

Complete the following steps:

1. Rename the `/usr/share/man` directory so that you can no longer access the man pages on the client system. Verify that the man pages are not available.

```
# mv /usr/share/man /usr/share/man.orig
# man ls
```

What message does the `man` command report?

No manual entry for `ls`.

2. Create a new `man` directory (`/usr/share/man`) to use as a mount point.

```
# cd /usr/share
# mkdir man
```

3. Mount the `/usr/share/man` directory from the server.

```
# mount server:/usr/share/man /usr/share/man
```

4. Verify that the man pages are available.

```
# man ls
```

Are the man pages available?

Yes

5. Verify and record the default `ro | rw` options used for this mount.

```
# mount
```

The `ro | rw` *option for the* `mount` *command is read/write (rw) by default.*

6. Write a file into the NFS-mounted file system.

```
# touch /usr/share/man/test
touch: /usr/share/man/test cannot create
```

What is the result of trying to write to the NFS-mounted file system?

You cannot write to the file system.

What conclusion can be reached by this exercise?

Even though the file system mount is read/write, by default, the actual `ro | rw` *permission is read-only, as defined when the directory was shared on the NFS server.*

7. Verify the list of mounts that the server provides.

```
# dfmounts server
RESOURCE      SERVER   PATHNAME             CLIENTS
-             server  /usr/share/man        client
```

8. Unmount the /usr/share/man directory, and verify the list of remote mounts from the server.

```
# umount /usr/share/man
# dfmounts server
```

No output from the dfmounts *command indicates that there are no clients mounting the file systems from the server. (This output still shows the mount.)*

Task 3 – On the NFS Server

Complete the following steps:

1. Unshare the /usr/share/man directory.

```
# unshareall
```

2. Change the share statement in the /etc/dfs/dfstab file for the /usr/share/man directory to read:

```
share -o ro=bogus /usr/share/man
```

3. Share the /usr/share/man directory.

```
# shareall
```

Task 4 – On the NFS Client

Complete the following step:

Attempt to mount the `/usr/share/man` directory again.

What happens?

The client reports the error message:

```
nfs mount: server:/usr/share/man: Permission denied
```

Task 5 – On the NFS Server

Complete the following steps:

1. Unshare the `/usr/share/man` directory.

```
# unshareall
```

2. Edit the `/etc/dfs/dfstab` file to remove the entry for the `/usr/share/man` directory.

Task 6 – On the NFS Client

Complete the following steps:

1. Return the `/usr/share/man` directory to its original configuration.

```
# cd /usr/share
# rmdir man
# mv man.orig man
```

2. Verify that the man pages are now available.

```
# man ls
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 7

Configuring AutoFS

Objectives

The AutoFS file system provides a mechanism for automatically mounting NFS file systems on demand and for automatically unmounting these file systems after a predetermined period of inactivity. The mount points are specified using local or distributed automount maps.

Upon completion of this module, you should be able to:

- Describe the fundamentals of the AutoFS file system
- Use automount maps

The following course map shows how this module fits into the current instructional goal.

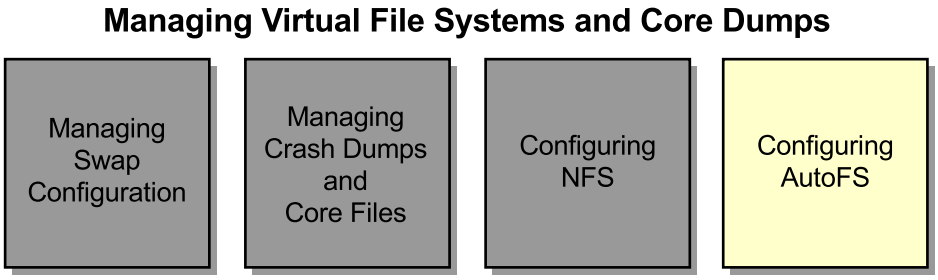


Figure 7-1 Course Map

Introducing the Fundamentals of AutoFS

AutoFS is a file system mechanism that provides automatic mounting using the NFS protocol. AutoFS is a client-side service. The AutoFS file system is initialized by the `/etc/rc2.d/S74autofs` automount script, which runs automatically when a system is booted. This script runs the `automount` command, which reads the AutoFS configuration files and also starts the automount daemon `automountd`. The `automountd` daemon runs continuously, mounting and unmounting remote directories on an as-needed basis.

Whenever a user on a client computer running the `automountd` daemon tries to access a remote file or directory, the daemon mounts the remote file system to which that file or directory belongs. This remote file system remains mounted for as long as it is needed. If the remote file system is not accessed for a defined period of time, the `automountd` daemon automatically unmounts the file system.

The AutoFS service mounts and unmounts file systems as required without any user intervention. The user does not need to use the `mount` and `umount` commands and does not need to know the superuser password.

The AutoFS file system enables you to do the following:

- Mount file systems on demand
- Unmount file systems automatically
- Centralize the administration of AutoFS mounts through the use of a name service, which can dramatically reduce administration overhead time
- Create multiple mount resources for read/write or read-only file systems

The automount facility contains three components:

- The AutoFS file system
- The `automountd` daemon
- The `automount` command

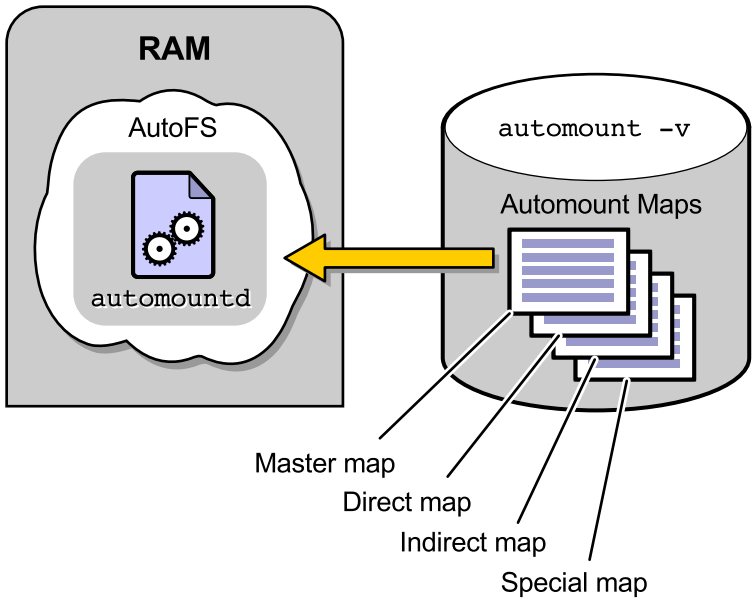


Figure 7-2 The AutoFS Features

AutoFS File System

An AutoFS file system's mount points are defined in the automount maps on the client system. After the AutoFS mount points are set up, activity under the mount points can trigger file systems to be mounted under the mount points. If the automount maps are configured, the AutoFS `kernel` module monitors mount requests made on the client. If a mount request is made for an AutoFS resource not currently mounted, the AutoFS service calls the `automountd` daemon, which mounts the requested resource.

The automountd Daemon

The /etc/rc2.d/S74autofs script starts the automountd daemon at boot time. The automountd daemon mounts file systems on demand and unmounts idle mount points.

Note – The automountd daemon is completely independent from the automount command. Because of this separation, you can add, delete, or change map information without having to stop and start the automountd daemon process.

The automount Command

The automount command, called at system startup time, reads the master map to create the initial set of AutoFS mounts. These AutoFS mounts are not automatically mounted at startup time, they are the points under which file systems are mounted on demand.

Using Automount Maps

The file system resources for automatic mounting are defined in automount maps. Figure 7-3 shows maps defined in the `/etc` directory.

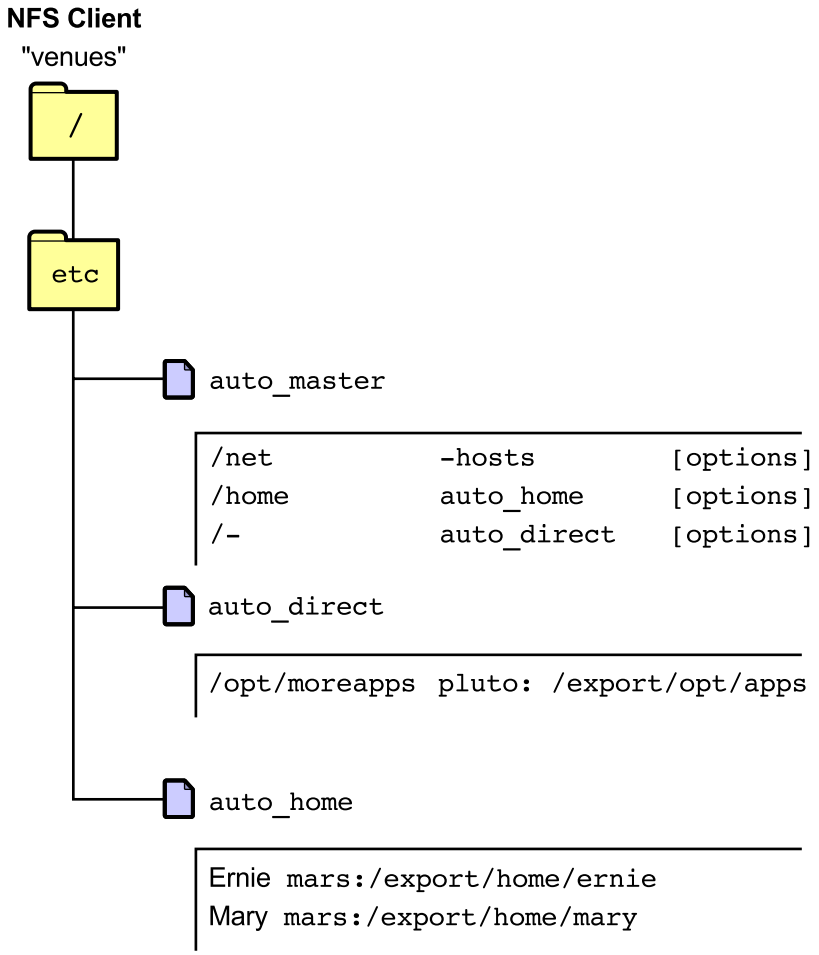


Figure 7-3 Configuring AutoFS Mount Points

The AutoFS map types are:

- Master map – Lists the other maps used for establishing the AutoFS file system. The `automount` command reads this map at boot time.
- Direct map – Lists the mount points as *absolute* path names. This map explicitly indicates the mount point on the client.
- Indirect map – Lists the mount points as *relative* path names. This map uses a relative path to establish the mount point on the client.
- Special – Provides access to NFS servers by using their host names.

The automount maps can be obtained from ASCII data files, NIS maps, NIS+ tables, or from an LDAP database. Together, these maps describe information similar to the information specified in the `/etc/vfstab` file for remote file resources.

The source for automount maps is determined by the `automount` entry in the `/etc/nsswitch.conf` file. For example, the entry:

`automount: files`

tells the `automount` command that it should look in the `/etc` directory for its configuration information. Using `nis` instead of *`files`* tells `automount` to check the NIS maps for its configuration information.

Configuring the Master Map

The `auto_master` map associates a directory, also called a mount point, with a map. The `auto_master` map is a master list specifying all the maps that the AutoFS service should check. Names of direct and indirect maps listed in this map refer to files in the `/etc` directory or to name service databases.

Associating a Mount Point With a Map

The following example shows an `/etc/auto_master` file.

```
# cat /etc/auto_master
# Master map for automounter
#
+auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
/xfn            -xfn
```

The general syntax for each entry in the `auto_master` map is:

`mount point  map name  mount options`

where:

`mount point`	The full path name of a directory. If the directory does not exist, the AutoFS service creates one, if possible.
`map name`	The name of a direct or indirect map. These maps provide mounting information. A relative path name in this field requires AutoFS to consult the `/etc/nsswitch.conf` file for the location of the map.
`mount options`	The general options for the map. The mount options are similar to those used for standard NFS mounts. However, the `nobrowse` option is an AutoFS-specific mount option.

Note – The plus (+) symbol at the beginning of the `+auto_master` line in this file directs the `automountd` daemon to look at the NIS, NIS+, or LDAP databases before it reads the rest of the map. If this line is commented out, only the local files are searched unless the `/etc/nsswitch.conf` file specifies that NIS, NIS+, or LDAP should be searched.

Identifying Mount Points for Special Maps

There are two mount points for special maps listed in the default `/etc/auto_master` file.

```
# cat /etc/auto_master
# Master map for automounter
#
+auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
/xfn            -xfn
```

The two mount points for special maps are:

The `-hosts` map	Provides access to all resources shared by NFS servers. The resources being shared by a server are mounted below the `/net/`*`hostname`* directory, or, if only the server's IP address is known, below the `/net/IPaddress` directory. The server does not have to be listed in the hosts database for this mechanism to work.
The `-xfn` map	Provides access to resources available through the Federated Naming Service (FNS). Resources associated with FNS mount below the `/xfn` directory.

Note – The `-xfn` map provides access to legacy FNS resources. Support for FNS is scheduled to cease with this release of the Solaris OE.

Using the /net Directory

Shared resources associated with the hosts map entry are mounted below the /net/*hostname* directory. For example, a shared resource named /documentation on host sys42 is mounted by the command:

```
# cd /net/sys42/documentation
```

Using the cd command to trigger the automounting of sys42's resource eliminates the need to log in to the system. Any user can mount the resource by executing the command to change to the directory that contains the shared resource. The resource remains mounted until a predetermined time period of inactivity has occurred.

Adding Direct Map Entries

The /- entry in the example master map defines a mount point for direct maps.

```
# cat /etc/auto_master
# Master map for automounter
#
+auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
/xfn            -xfn
/-              auto_direct     -ro
```

The /- mount point is a pointer that informs the automount facility that the full path names are defined in the file specified by *map_name* (the /etc/auto_direct file in this example).

Note – The /- entry is not an entry in the default master map. This entry has been added here as an example. The other entries in this example already exist in the auto_master file.

Even though the map_name entry is specified as auto_direct, the automount facility automatically searches for all map-related files in the /etc directory; therefore, based upon the automount entry in the /etc/nsswitch.conf file, the auto_direct file is the /etc/auto_direct file.

Note – An NIS or NIS+ master map can have only one direct map entry. A master map that is a local file can have any number of entries.

Creating a Direct Map

Direct maps specify the absolute path name of the mount point, the specific options for this mount, and the shared resource to mount. For example:

```
# cat /etc/auto_direct
# Superuser-created direct map for automounter
#
/apps/frame      -ro,soft   server1:/export/framemaker,v5.5.6
/opt/local       -ro,soft   server2:/export/unbundled
/usr/share/man   -ro,soft   server3,server4,server5:/usr/share/man
```

The syntax for direct maps is:

key [*mount-options*] *location*

where:

key	The full path name of the mount point for the direct maps.
mount-options	The specific options for a given entry.
location	The location of the file resource specified in *server:pathname* notation.

The following direct map entry specifies that the client mounts the /usr/share/man directory as read-only from the servers server3, server4, or server5, as available.

```
/usr/share/man    -ro        server3,server4,server5:/usr/share/man
```

This entry uses a special notation, a comma-separated list of servers, to specify a powerful automount feature—multiple locations for a file resource. The automountd daemon automatically mounts the /usr/share/man directory as needed, from servers server3, server4, or server5, with server proximity and administrator-defined weights determining server selection. If the nearest server fails to respond within the specified time-out period, the next server with the nearest proximity is selected.

Note – Selection criteria for multiple servers, such as server proximity and administrator-defined weights, is defined in the "Replicated File Systems" section of the automount man page.

Adding Indirect Map Entries

The /home entries define mount points for indirect maps. The map auto_home lists relative path names only. Indirect maps obtain the initial path of the mount point from the master map.

```
# cat /etc/auto_master
# Master map for automounter
#
+auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
/xfn            -xfn
```

The Solaris 2.6 through Solaris 9 OE releases support browsing of indirect maps and special maps with the -browse option. This support allows all of the potential mount points to be visible, regardless of whether they are mounted. The -nobrowse option disables the browsing of indirect maps. Therefore, in this example, the /home automount point does not provide browser functions for any directory other than those that are currently mounted. The default for this option is -browse.

Creating an Indirect Map

Use the `auto_home` indirect map to list the location of home directories across the network. For example,

```
# cat /etc/auto_home
# Home directory map for automounter
#
+auto_home
stevenu        host5:/export/home/stevenu
johnnyd        host6:/export/home/johnnyd
wkd            server1:/export/home/wkd
mary           mars:/export/home/mary
```

Note – The plus (+) symbol at the beginning of the `+auto_master` line in this file directs the `automountd` daemon to look at the NIS, NIS+, or LDAP databases before it reads the rest of the map. If this line is commented out, only the local files are searched unless the `/etc/nsswitch.conf` file specifies that NIS, NIS+, or LDAP should be searched.

The following describes the syntax for indirect maps:

key [*mount-options*] *location*

where:

key	Specifies the path name of the mount point relative to the beginning of the path name specified in the master map.
mount-options	Specifies the options for a given entry.
location	Specifies the location of the file resource specified in *server:pathname* notation.

The example /etc/auto_home file implies the following mount points: /home/stevenu, /home/johnnyd, /home/wkd, and /home/mary. Figure 7-4 shows the /home/mary mount point.

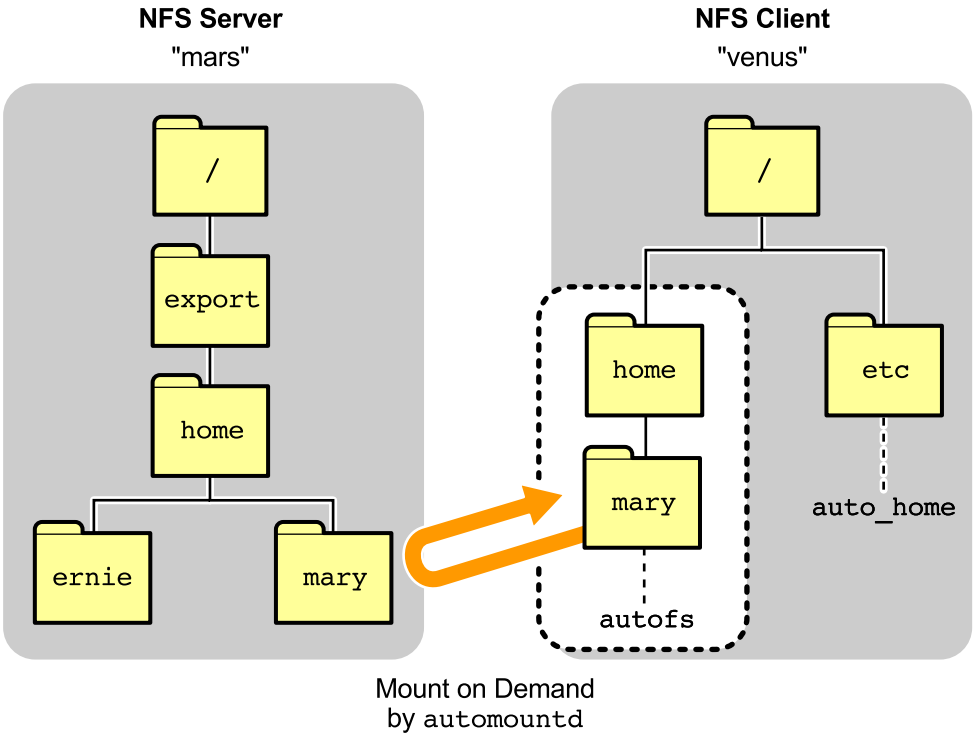


Figure 7-4 The Mount Points

Reducing the auto_home Map to a Single Line

The following entry reduces the auto_home file to a single line. The use of substitution characters specifies that for every login ID, the client remotely mounts the /export/home/*loginID* directory from the NFS server server1 onto the local mount point /home/*loginID*.

```
*       server1:/export/home/&
```

Figure 7-5 Mounting a Directory on a Local Mount Point

Figure 7-5 shows that this entry uses the wildcard character (*) to match any key. The substitution character (&) at the end of the location is replaced with the matched key field. Using wildcard and substitution characters works only when all home directories are on a single server (in this example, server1).

Updating the Automount Maps

When making changes to the master map or creating a direct map, run the `automount` command to make the changes effective.

Running the `automount` Command

The syntax of the command is:

```
automount [-t duration] [-v]
```

where:

`-t` *`duration`*	Specifies a time, in seconds, that the file system remains mounted when not in use. The default is 600 seconds (10 minutes).
`-v`	Specifies verbose mode, which displays output as the `automount` command executes.

You can modify the master map entries or add entries for new maps. However, you must run the `automount` command to make these changes effective.

You do not have to stop and restart the `automountd` daemon after making changes to existing entries in a direct map, because the daemon is stateless. You can modify existing entries in the direct map at any time. The new information is used when the `automountd` daemon next accesses the map entry to perform a mount.

Any modifications to indirect maps are automatically used by the `automountd` daemon.

A modification is a change to options or resources. A change to the key (the mount point) or a completely new line is an added entry, a deleted entry, or both.

Use Table 7-1 to determine whether you should run (or rerun) the `automount` command.

Table 7-1 When to Run the `automount` Command

Automount Map	Run if the Entry is Added or Deleted	Run if the Entry is Modified
`master map`	Yes	Yes
Direct map	Yes	No
Indirect map	No	No

Note – You can run the `automount` command at any time to rescan the maps, even if running the command is not required.

Verifying AutoFS Entries in the `/etc/mnttab` File

The `/etc/mnttab` file is a file system that provides read-only access to the table of mounted file systems for the current host. Mounting a file system adds an entry to this table. Unmounting a file system removes the entry from this table. Each entry in the table is a line of fields separated by spaces in the form of:

```
special   mount_point  fstype   options   time
```

where:

special	The name of the resource to be mounted
mount_point	The path name of the directory on which the file system is mounted
fstype	The type of file system
options	The mount options
time	The time at which the file system was mounted

You can display the /etc/mnttab file to obtain a snapshot of the mounted file systems, including those mounted as an AutoFS file system type.

```
# cat /etc/mnttab
/dev/dsk/c0t0d0s0       /       ufs
rw,intr,largefiles,onerror=panic,suid,dev=2200000       1008255791
/proc   /proc   proc    dev=4080000     1008255790
mnttab  /etc/mnttab     mntfs   dev=4140000     1008255790
fd      /dev/fd fd      rw,suid,dev=4180000     1008255794
swap    /var/run        tmpfs   dev=1   1008255797
swap    /tmp    tmpfs   dev=2   1008255802
/dev/dsk/c0t0d0s7       /export/home    ufs
rw,intr,largefiles,onerror=panic,suid,dev=2200007       1008255802
-hosts  /net    autofs  indirect,nosuid,ignore,nobrowse,dev=4300001
1008255810
auto_home       /home   autofs  indirect,ignore,nobrowse,dev=4300002
1008255810
-xfn    /xfn    autofs  indirect,ignore,dev=4300003     1008255810
sys44:vold(pid264)      /vol    nfs     ignore,dev=42c0001
1008255827
```

Stopping and Starting the Automount System

The /etc/init.d/autofs script executes automatically as the system transitions between run levels, or you can run the script manually from the command line.

Stopping the Automount System

When the autofs script runs with the stop argument, it performs a forced unmount of all AutoFS file systems, and it then kills the automountd daemon.

The autofs script runs with the stop argument automatically when transitioning to:

- Run level S using the /etc/rcS.d/K41autofs script
- Run level 1 using the /etc/rc1.d/K41autofs script
- Run level 0 using /etc/rc0.d/K41autofs scripts script

To run the script on demand, become superuser, and kill the `automountd` daemon by typing the following command:

```
# /etc/init.d/autofs stop
```

Starting the Automount System

When the `autofs` script is run with the `start` argument, the script starts the `automountd` daemon, and then it runs the `automount` utility as a background task.

The script runs with the `start` argument automatically when transitioning to run level 2 using the `/etc/rc2.d/S74autofs` script.

To run the script on demand, become superuser, and start the `automountd` daemon by performing the command:

```
# /etc/init.d/autofs start
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which option to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using the Automount Facility (Level 1)

In this exercise, you use the automount facility to automatically mount man pages and to mount a user's home directory.

Preparation

Choose a partner for this lab, and determine which system will be configured as the NFS server and which will serve as the NFS client. Verify that entries for both systems exist in the `/etc/hosts` file of each system. Refer to the lecture notes as necessary to perform the steps listed.

Tasks

Perform the following tasks:

- On the server, perform the steps required to share the `/usr/share/man` directory.
- On the client, rename the `/usr/share/man` directory to `/usr/share/man.orig` directory, and create a new mount point for the `/usr/share/man` directory. Edit the master map so that it calls a direct map. Create the direct map to mount the `/usr/share/man` directory from the server. Use the `automount` command to update the `automountd` daemon. Test that the man pages work, and verify the mount that occurs.
- Create a new, identical user on both the server and client that uses `/export/home/`*`username`* for the user's home directory. On both systems, make the changes required in the `/etc/passwd` file to set the home directory for this new user to the `/home/`*`username`* directory.
- On the server, perform the steps required to share the `/export/home` directory.
- On both systems, make the changes required in the `/etc/auto_home` file to allow both systems to automatically mount the `/export/home/`*`username`* directory when the new user calls for the `/home/`*`username`* directory. Test the new user login on both systems, and verify that the mounts take place. Log in as `root` when finished.
- On the server, unshare the `/export/home` and `/usr/share/man` directories, and remove entries for these directories from the `/etc/dfs/dfstab` file. Stop the NFS server daemons.
- On the client, remove the direct map entry from the `/etc/auto_master` file, and update the `automountd` daemon with the change. Return the `/usr/share/man` directory to its original configuration.

Exercise: Using the Automount Facility (Level 2)

In this exercise, you use the automount facility to automatically mount man pages and to mount a user's home directory.

Preparation

Choose a partner for this lab, and determine which system will be configured as the NFS server and which will serve as the NFS client. Verify that entries for both systems exist in the `/etc/hosts` file of each system. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

Perform the following tasks:

- On the server, perform the steps required to share the `/usr/share/man` directory.
- On the client, rename the `/usr/share/man` directory to `/usr/share/man.orig` directory, and create a new mount point for the `/usr/share/man` directory. Edit the master map so that it calls a direct map. Create the direct map to mount `/usr/share/man` directory from the server. Use the `automount` command to update the `automountd` daemon. Test that the man pages work, and verify the mount that occurs.
- Create a new, identical user on both the server and client that uses `/export/home/`*`username`* for the user's home directory. On both systems, make the changes required in the `/etc/passwd` file to set the home directory for this new user to the `/home/`*`username`* directory.
- On the server, perform the steps required to share the `/export/home` directory.
- On both systems, make the changes required in the `/etc/auto_home` file to allow both systems to automatically mount the `/export/home/`*`username`* directory when the new user calls for the `/home/`*`username`* directory. Test the new user login on both systems, and verify that the mounts take place. Log in as `root` when finished.
- On the server, unshare the `/export/home` and `/usr/share/man` directories and remove entries for these directories from the `/etc/dfs/dfstab` file. Stop the NFS server daemons.
- On the client, remove the direct map entry from the `/etc/auto_master` file, and update the `automountd` daemon with the change. Return the `/usr/share/man` directory to its original configuration.

Tasks

Complete the following tasks.

Task 1– On the Server Host

Complete the following steps:

1. Edit the `/etc/dfs/dfstab` file, and add a line to share the man pages.
2. Use the `pgrep` command to check if the `mountd` daemon is running.
   - If the `mountd` daemon is not running, start it.
   - If the `mountd` daemon is running, share the new directory.

Task 2 – On the Client Host

Complete the following steps:

1. Rename the `/usr/share/man` directory so that you cannot view the man pages installed on the client system.

   _____

2. Edit the `/etc/auto_master` file to add an entry for a direct map.

   _____

3. Use the `vi` editor to create a new file called `/etc/auto_direct`, and add an entry to the file to share the man pages.

   _____

4. Run the `automount` command to update the list of directories managed by the `automountd` daemon.

   _____

5. Test the configuration, and verify that a mount for the `/usr/share/man` directory exists after accessing the man pages.

   _____

   What did you observe to indicate that the `automount` operation was successful?

   _____

Task 3 – On the Server Host

Complete the following steps:

1. Verify that the `/export/home` directory exists. If it does not exist, create it.
2. Add a user account with the following characteristics:
   - User ID: `3001`
   - Primary group: `10`
   - Home directory: `/export/home/usera`
   - Login shell:`/bin/ksh`
   - User name: `usera`
3. Remove the lock string from the new user's `/etc/shadow` file entry.

Task 4 – On the Client Host

Complete the following steps:

1. Verify that the `/export/home` directory exists. If it does not exist, create it.
2. Add a user account with the following characteristics:
   - User ID: `3001`
   - Primary group: `10`
   - Home directory: `/export/home/usera`
   - Login shell: `/bin/ksh`
   - User name: `usera`
3. Remove the lock string from the new user's `/etc/shadow` file entry.

Task 5 – On Both Systems

Complete the following steps:

1. Edit the `/etc/passwd` file, and change the home directory for the new user from the `/export/home/`*`username`* directory to `/home/`*`username`*, where *`username`* is the name of your new user.
2. Edit the `/etc/auto_home` file. Add the following line, and replace *`username`* with the name of your new user:

`username` *`server`*`:/export/home/`*`username`*

Task 6 – On the Server Host

Complete the following steps:

1. Edit the `/etc/dfs/dfstab` file, and add a line to share the `/export/home` directory.
2. Use the `pgrep` command to check if the `mountd` daemon is running.
   - If the `mountd` daemon is not running, start it.
   - If the `mountd` daemon is running, share the new directory.

Task 7 – On Both Systems

Complete the following step:

Log in as the new user.

Do both systems automatically mount the new user's home directory?

_____

Which directory is mounted, and what is the mount point:

- On the server?

  _____

- On the client?

  _____

Task 8 – On the Client Host

Complete the following steps:

1. On the client, log off as `usera`.

   _____

2. Remove the entry for `usera` from the `/etc/auto_home` map.

   _____

3. Remove the entry for the `auto_home` map from the `/etc/auto_master` map.

   _____

4. Reboot the client.

   _____

5. Remove the `/usr/share/man` directory.

   _____

6. Rename the `/usr/share/man.orig` directory to `/usr/share/man`.

   _____

Task 9 – On the Server Host

Complete the following steps:

1. Log off as `usera`.

   _____

2. After the client reboots as described in Step 4 of "Task 8 – On the Client Host" section on page 7-24, remove the entry for `usera` from the `/etc/auto_home` map.

   _____

3. Remove the entries from `/etc/dfs/dfstab` file.

   _____

4. Unshare mounted directories.

   _____

Exercise: Using the Automount Facility (Level 3)

In this exercise, you use the automount facility to automatically mount man pages and to mount a user's home directory.

Preparation

Choose a partner for this lab, and determine which system will be configured as the NFS server and which will serve as the NFS client. Verify that entries for both systems exist in the `/etc/hosts` file of each system. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

Perform the following tasks:

- On the server, perform the steps required to share the `/usr/share/man` directory.
- On the client, rename the `/usr/share/man` directory to `/usr/share/man.orig` directory, and create a new mount point for the `/usr/share/man` directory. Edit the master map so that it calls a direct map. Create the direct map to mount `/usr/share/man` directory from the server. Use the `automount` command to update the `automountd` daemon. Test that the man pages work, and verify the mount that occurs.
- Create a new, identical user on both the server and client that uses `/export/home/`*`username`* for the user's home directory. On both systems, make the changes required in the `/etc/passwd` file to set the home directory for this new user to the `/home/`*`username`* directory.
- On the server, perform the steps required to share the `/export/home` directory.
- On both systems, make the changes required in the `/etc/auto_home` file to allow both systems to automatically mount the `/export/home/`*`username`* directory when the new user calls for the `/home/`*`username`* directory. Test the new user log in on both systems, and verify that the mounts that happen. Log in as `root` when finished.

- On the server, unshare the `/export/home` and `/usr/share/man` directories and remove entries for these directories from the `/etc/dfs/dfstab` file. Stop the NFS server daemons.
- On the client, remove the direct map entry from the `/etc/auto_master` file, and update the `automountd` daemon with the change. Return the `/usr/share/man` directory to its original configuration.

Tasks and Solutions

The following section provides the tasks with their solutions.

Task 1 – On the Server Host

Complete the following steps:

1. Edit the `/etc/dfs/dfstab` file, and add a line to share the man pages.

```
share -o ro /usr/share/man
```

2. Use the `pgrep` command to check if the `mountd` daemon is running.

```
# pgrep -xl mountd
400 mountd
```

- If the `mountd` daemon is not running, start it.

```
# /etc/init.d/nfs.server start
```

- If the `mountd` daemon is running, share the new directory.

```
# shareall
```

Task 2 – On the Client Host

Complete the following steps:

1. Rename the `/usr/share/man` directory so that you cannot view the man pages installed on the client system.

```
# cd /usr/share/
# mv man man.orig
```

2. Edit the /etc/auto_master file to add an entry for a direct map.

```
# vi /etc/auto_master
/-    auto_direct
```

3. Use the vi editor to create a new file called /etc/auto_direct, and add an entry to the file to share the man pages.

```
# vi /etc/auto_direct
/usr/share/man      server:/usr/share/man
```

4. Run the automount command to update the list of directories managed by the automountd daemon.

```
# automount -v
```

5. Test the configuration, and verify that a mount for the /usr/share/man directory exists after accessing the man pages.

```
# man ls
<-- output from man command -- >
# mount | grep man
/usr/share/man on sys44:/usr/share/man
remote/read/write/setuid/dev=42c0003 on Thu Dec 13 08:07:26 2001
```

What did you observe to indicate that the automount operation was successful?

This operation should automatically mount the directory in which the manuals are stored. In other words, the man *command should work.*

Task 3 – On the Server Host

Complete the following steps:

1. Verify that the /export/home directory exists. If it does not exist, create it.

```
# ls /export/home
```

Note – Perform the next command if the /export/home directory does not exist.

```
# mkdir /export/home
```

2. Add a user account with the following characteristics:
   - User ID: `3001`
   - Primary group: `10`
   - Home directory: `/export/home/usera`
   - Login shell:`/bin/ksh`
   - User name: `usera`

```
# useradd -u 3001 -g 10 -m -d /export/home/usera -s /bin/ksh usera
```

3. Remove the lock string from the new user's `/etc/shadow` file entry.

Task 4 – On the Client Host

Complete the following steps:

1. Verify that the `/export/home` directory exists. If it does not, create it.

```
# ls /export
# mkdir /export/home
```

2. Add a user account with the following characteristics:
   - User ID: `3001`
   - Primary group: `10`
   - Home directory: `/export/home/usera`
   - Login shell: `/bin/ksh`
   - User name: `usera`

```
# useradd -u 3001 -g 10 -m -d /home/usera -s /bin/ksh usera
```

3. Remove the lock string from the new user's `/etc/shadow` file entry.

Task 5 – On Both Systems

Complete the following steps:

1. Edit the `/etc/passwd` file, and change the home directory for the new user from the `/export/home/`*`username`* directory to `/home/`*`username`*, where *`username`* is the name of your new user.

```
# vi /etc/passwd
```

2. Edit the `/etc/auto_home` file. Add the following line, and replace *`username`* with the name of your new user:

`username` *`server`*`:/export/home/`*`username`*

Task 6 – On the Server Host

Complete the following steps:

1. Edit the `/etc/dfs/dfstab` file, and add a line to share the `/export/home` directory.

```
share /export/home
```

2. Use the `pgrep` command to check if the `mountd` daemon is running.

```
# pgrep -xl mountd
    391 mountd
```

- If the `mountd` daemon is not running, start it.

```
# /etc/init.d/nfs.server start
```

- If the `mountd` daemon is running, share the new directory.

```
# shareall
```

Task 7 – On Both Systems

Complete the following step:

Log in as the new user.

```
# su - usera
```

Do both systems automatically mount the new user's home directory?

Yes, this should work.

Which directory is mounted, and what is the mount point:

- On the server?

  The `/home/username` directory *is mounted on the* `/export/home/username` *directory.*

- On the client?

  The `/home/username` directory *is mounted on the* `server:/export/home/username` *directory.*

Task 8 – On the Client Host

Complete the following steps:

1. On the client, log off as `usera`.
2. Remove the entry for `usera` from the `/etc/auto_home` map.
3. Remove the entry for the `auto_home` map from the `/etc/auto_master` map.
4. Reboot the client.

```
# init 6
```

5. Remove the `/usr/share/man` directory.

```
# rmdir /usr/share/man
```

6. Rename the `/usr/share/man.orig` directory to `/usr/share/man`.

```
# mv /usr/share/man.orig /usr/share/man
```

Task 9 – On the Server Host

Complete the following steps:

1. Log off as `usera`.
2. After the client reboots as described in Step 4 of "Task 8 – On the Client Host" section on page 7-30, remove the entry for `usera` from the `/etc/auto_home` map.
3. Remove the entries from `/etc/dfs/dfstab` file.
4. Unshare mounted directories.

```
# unshareall
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Describing RAID and the Solaris™ Volume Manager Software

Objectives

A redundant array of independent disks (RAID) configuration enables you to expand the characteristics of a storage volume beyond the physical limitations of a single disk. You can use a RAID configuration to increase disk capacity as well as to improve disk performance and fault tolerance. The Solaris™ Volume Manager software provides a graphical user interface (GUI) tool to simplify system administration tasks on storage devices. Upon completion of this module, you should be able to:

- Describe RAID
- Describe Solaris Volume Manager software concepts

The following course map shows how this module fits into the current instructional goal.

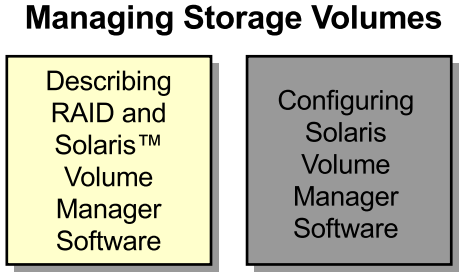


Figure 8-1 Course Map

Introducing RAID

RAID is a classification of methods to back up and to store data on multiple disk drives. There are six levels of RAID as well as a non-redundant array of independent disks (RAID 0). The Solaris Volume Manager software uses metadevices, which are product-specific definitions of logical storage volumes, to implement RAID 0, RAID 1, and RAID 5:

- RAID 0: Non-redundant disk array (concatenation and striping)
- RAID 1: Mirrored disk array
- RAID 5: Block-interleaved distributed-parity

RAID 0

RAID-0 volumes, including both stripes and concatenations, are composed of slices and let you expand disk storage capacity. You can either use RAID-0 volumes directly or use the volumes as the building blocks for RAID-1 volumes (mirrors). There are two types of RAID-0 volumes:

- Concatenated volumes (or concatenations)

  A concatenated volume writes data to the first available slice. When the first slice is full, the volume writes data to the next available slice.

- Striped volumes (or stripes)

  A stripe distributes data equally across all slices in the stripe.

RAID-0 volumes allow you to expand disk storage capacity efficiently. These volumes do not provide data redundancy. If a single slice fails on a RAID-0 volume, there is a loss of data.

Concatenated Volumes

Figure 8-2 shows that in a concatenated RAID 0 volume, data is organized serially and adjacently across disk slices, forming one logical storage unit.

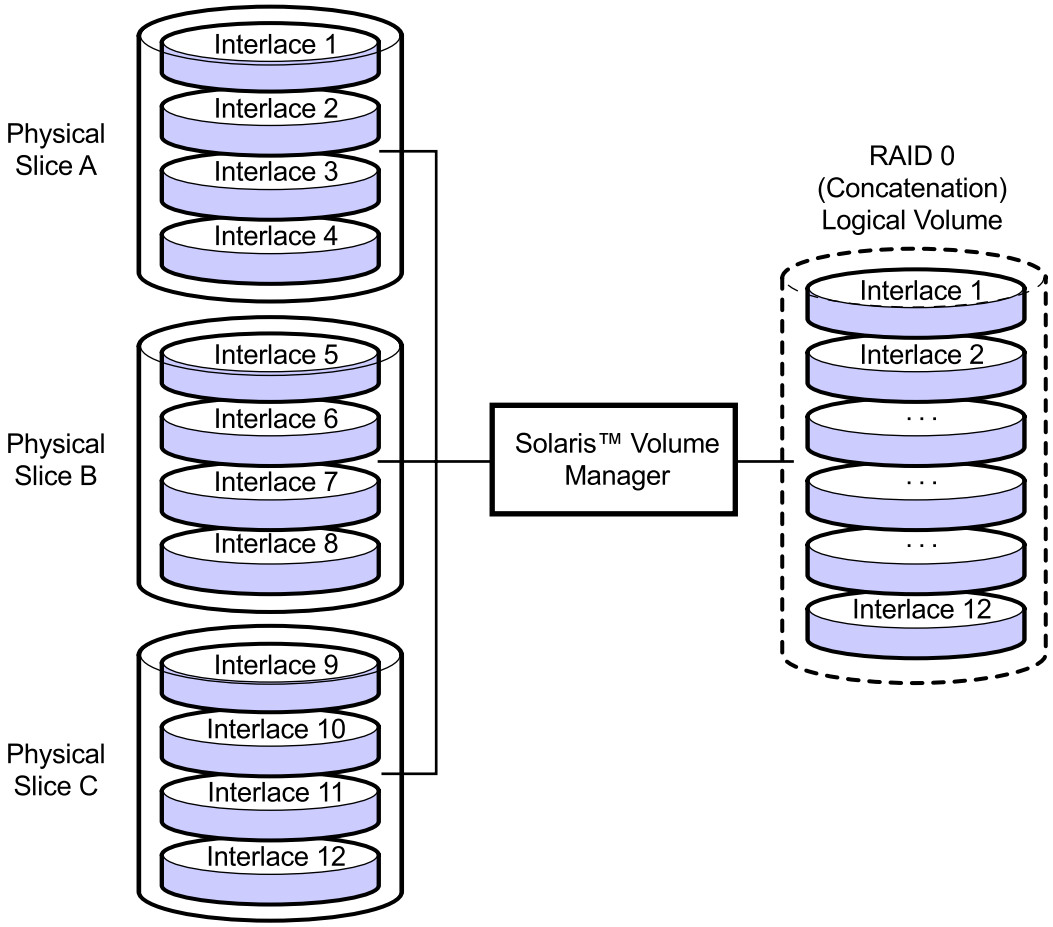


Figure 8-2 RAID-0 Concatenation

A concatenation combines the capacities of several slices to get a larger storage capacity. You can add more slices to the concatenation as the demand for storage increases. You can add slices at anytime, even if other slices are currently active.

Note – An interlace is a grouped segment of blocks on a particular slice.

The default behavior of concatenated RAID-0 volumes is to fill a physical component within the volume before beginning to store data on subsequent components within the concatenated volume. However, the default behavior of UFS file systems within the Solaris OE is to distribute the load across devices assigned to the volume containing a file system. This anomaly can make it seem that concatenated RAID-0 volumes distribute data across the components of the volume in a round-robin method. The data distribution is a function of the UFS file system that is mounted in the concatenated volume and is not a function of the concatenated volume itself.

You can also use a concatenation to expand any active and mounted UFS file system without having to bring down the system. Usually, the capacity of a concatenation is the total size of all the slices in the concatenation.

Striped Volumes

Figure 8-3 shows the arrangement of a RAID-0 volume. A RAID 0 volume configured as a stripe arranges data across two or more slices. Striping alternates equally-sized segments of data across two or more slices, forming one logical storage unit. These segments are interleaved round-robin, so that the combined space is created alternately from each slice.

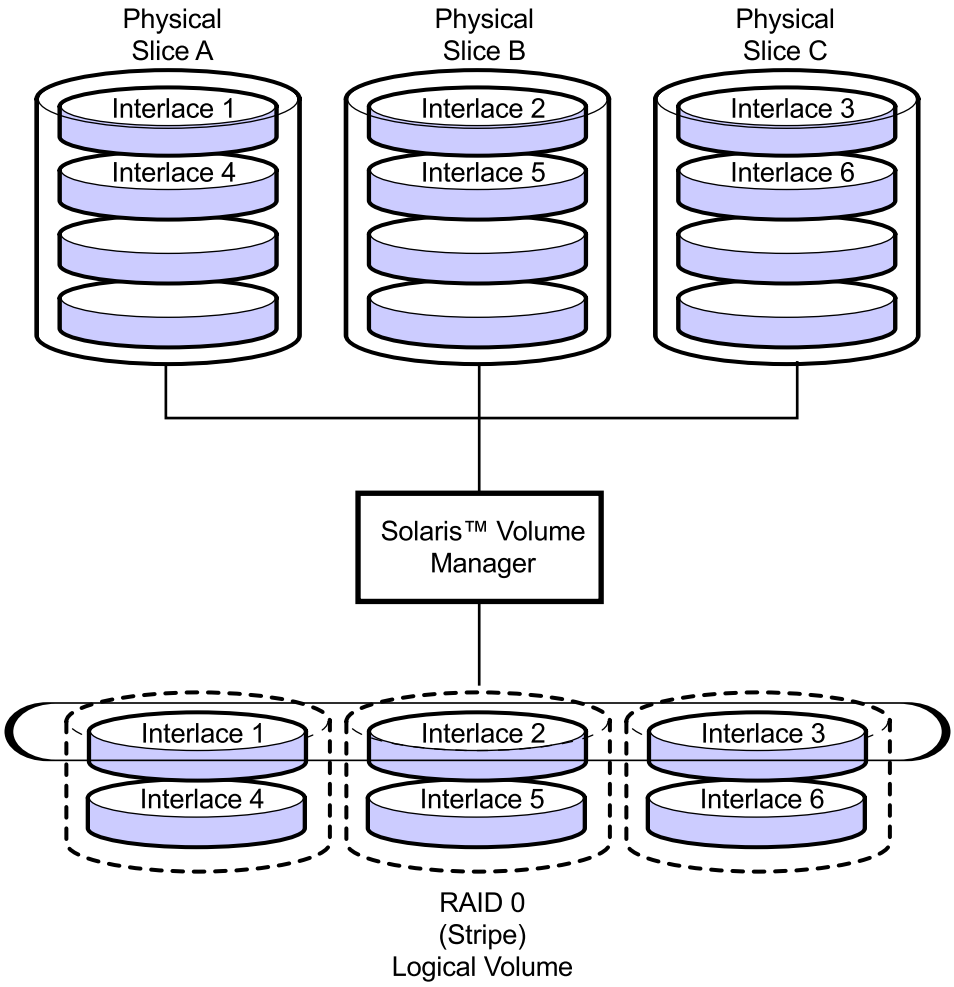


Figure 8-3 RAID-0 Stripe

Striping enables parallel data access because multiple controllers can access the data at the same time. Parallel access increases Input/Output (I/O) throughput because multiple disks in the volume are busy servicing I/O requests simultaneously.

You cannot convert an existing file system directly to a stripe. You must first back up the file system, create the stripe, and then restore the file system to the stripe.

For sequential I/O operations on a stripe, the Solaris Volume Manager software reads all the blocks in an *interlace.* An interlace is a grouped segment of blocks on a particular slice. The Solaris Volume Manager software then reads all the blocks in the interlace on the second slice, and so on.

An interlace is the size, in Kbytes, Mbytes, or blocks, of the logical data chunks on a stripe. Depending on the application, different interlace values can increase performance for your configuration. The performance increase comes from several disk head-arm assemblies (HDAs) concurrently executing I/O operations. When the I/O request is larger than the interlace size, you might get better performance.

When you create a stripe, you can set the interlace value or use the Solaris Volume Manager software's default interlace value of 16 Kbytes. After you create the stripe, you cannot change the interlace value (although you could back up the data on it, delete the stripe, create a new stripe with a new interlace value, and then restore the data).

RAID 1

RAID-1 volumes, also known as mirror volumes, are typically composed of RAID-0 volumes and provide the advantage of data redundancy. The disadvantage is the higher cost incurred by requiring two RAID-1 devices wherever a single RAID-0 device is mirrored. Typical topics to be considered when configuring mirrors are:

- Trade-offs when using mirrors
- Uses of multiple submirrors
- RAID 0+1
- RAID 1+0
- Mirror read, write, and synchronization options
- Mirror configuration guidelines

Trade-Offs When Using Mirrors

A RAID-1 (mirror) volume maintains identical copies of the data in RAID-0 volumes. Mirroring requires more disks. You need at least twice as much disk space as the amount of data to be mirrored.

After configuring a mirror, you can use it as if it were a physical slice. With multiple copies of data available, data access time is reduced if the mirror read and write policies are properly configured. You then use read and write policies to distribute the access to the submirrors evenly across the mirror. The mirror read and write policies are described in detail later in this module.

You can mirror any file system, including existing file systems. You can also use a mirror for any application, such as a database.

Using Multiple Submirrors

A mirror is made of two or more RAID-0 volumes configured as either stripes or concatenations. The mirrored RAID-0 volumes are called submirrors. A mirror consisting of two submirrors is known as a two-way mirror, while a mirror consisting of three submirrors is known as a three-way mirror.

Creating a two-way mirror is usually sufficient for data redundancy. A third submirror lets you maintain redundancy with one of the other two submirrors offline.

When a submirror is offline, it is in a read-only mode. The Solaris Volume Manager software tracks all the changes written to the online submirror. When the submirror is brought back online, only the newly written portions are resynchronized. Other reasons for taking the submirror offline include troubleshooting and repair.

You can attach or detach a submirror from a mirror at any time, though at least one submirror must remain attached to the mirror at all times. Usually, you begin the creation of a mirror with only a single submirror, after which you can attach additional submirrors.

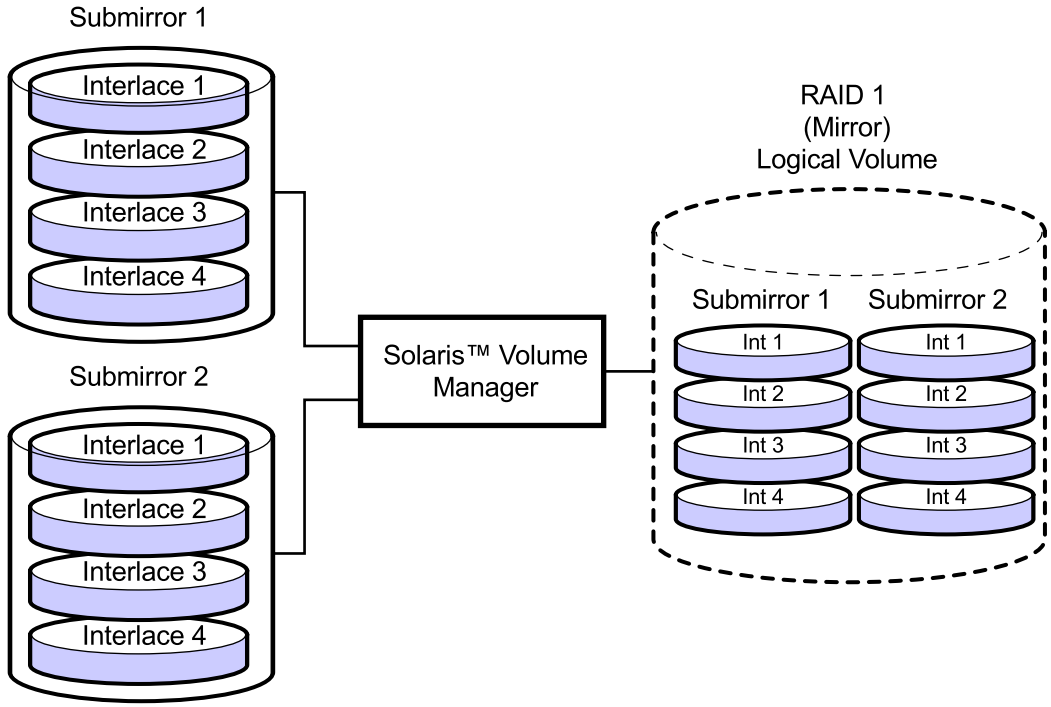


Figure 8-4 RAID-1 Mirror

The Solaris Volume Manager software makes duplicate copies of the data located on multiple physical disks. The Solaris Volume Manager software presents one virtual disk to the application. All disk writes are duplicated, and disk reads come from one of the underlying submirrors. If the submirrors are not of equal size, the total capacity of the mirror is limited by the size of the smallest submirror.

RAID 0+1

In RAID-0+1 volumes, stripes are mirrored to each other. In a pure RAID-0+1 configuration, the failure of one slice would cause the failure of the whole submirror.

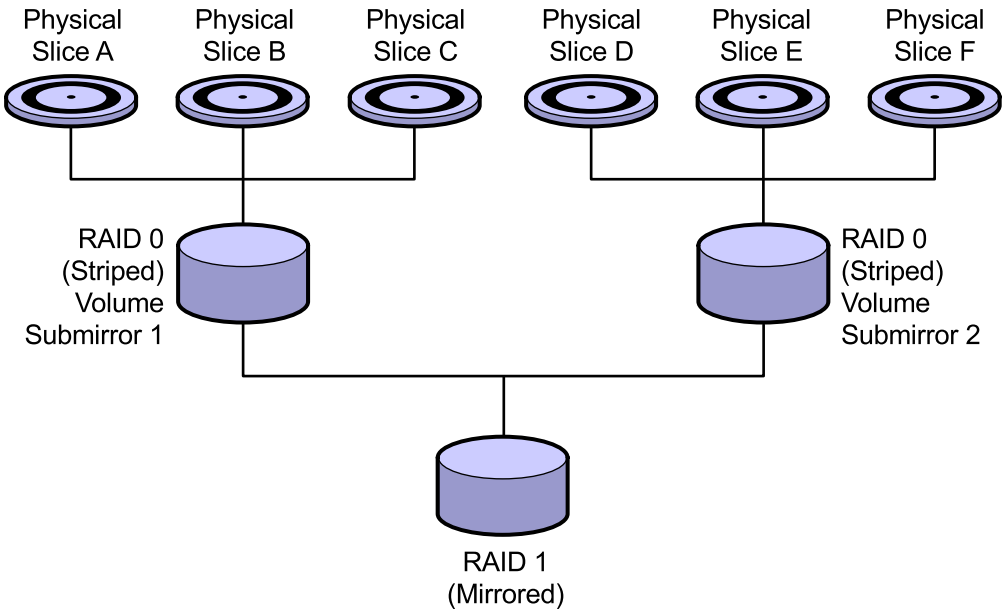


Figure 8-5 RAID-0+1 Mirror of Stripes

Figure 8-5 shows an example of a RAID-0+1 configuration. A failure in slice A, B, or C causes a failure of the entire Submirror 1. A failure in slice D, E, or F causes a failure of the entire Submirror 2. One failure in each submirror of the RAID 0+1 mirror causes a failure of the entire mirror.

RAID 1+0

RAID-1+0 volumes consist of multiple mirrors striped together. RAID 1+0 provides greater data security, because a failure of a single physical disk slice causes a failure for only one half of one of the submirrors, leaving most of the configuration's redundancy intact.

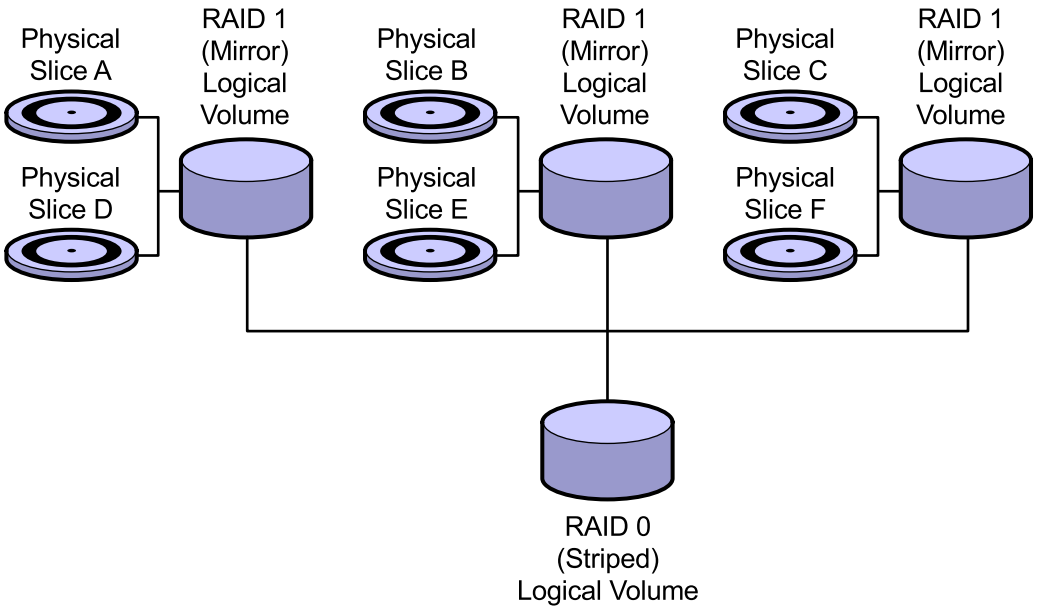


Figure 8-6 RAID 1+0 Stripe of Mirrors

Figure 8-6 shows an example of a RAID-1+0 volume configuration. RAID-1+0 consists of three slices. Each of these three slices mirrors itself. The RAID 0 stripe can tolerate three simultaneous physical slice failures, one in each RAID-1 mirror, before the entire RAID-0 stripe is considered to have failed. This is a more fault-tolerant configuration, as compared with the RAID-0+1 mirror. If both submirrors in any one of the mirrors fail, one third of the data is lost, and the RAID-1+0 volume is also considered failed.

Mirror Options

To optimize mirror performance, use the following options:

- Mirror read policy
- Mirror write policy

Note – The mirror options listed here are representative of the options presented when configuring RAID-1 mirrors using the Solaris Volume Manager software.

You can define mirror options when you initially create the mirror or after you set up the mirror. You can distribute the load across the submirrors to improve read performance. Table 8-1 describes the configurable mirror read policies.

Table 8-1 Mirror Read Policies

Read Policy	Description
Round Robin (default)	Balances the load across the submirrors
Geometric	Enables the system to divide reads among submirrors on the basis of a logical disk block address
First	Directs all reads to the first submirror

You can improve write performance by replicating all submirrors simultaneously. If a failure occurs during this write, all submirrors will be in an unknown state. Table 8-2 describes the configurable mirror write policies.

Table 8-2 Mirror Write Policies

Write Policy	Description
Parallel (Default)	Replicates a write to a mirror, and dispatches the write to all of the submirrors simultaneously
Serial	Specifies that writes to one submirror must complete before initiating writes to the next submirror

When a submirror is offline, any writes to the mirror are tracked in a dirty region log. When the submirror is brought back online, those regions must be updated or resynchronized.

Mirror Configuration Guidelines

The general configuration guidelines for configuring Solaris Volume Manager software mirrors are:

- Keep the slices of different submirrors on different disks and on different controllers for the best data protection. Organizing submirrors across separate controllers reduces the impact of a single controller failure and also improves mirror performance.
- Use the same type of disks and controllers in a single mirror. Particularly in old sMall Computer System Interface (SCSI) or storage module drive (SMD) storage devices, different models or brands of disks or controllers can vary in performance. Different performance levels can lead to a decrease in overall performance.
- Use submirrors of the same size to reduce unused disk space.
- Mount the mirror device directly. Do not try and mount a submirror directly, unless it is offline and mounted as read-only. Do not mount a slice that is part of a submirror, or you might destroy data and crash the system.
- Mirroring improves read performance, but reduces write performance. Mirroring improves read performance only in threaded or asynchronous I/O situations. There is no performance gain if there is only a single thread reading from the volume.
- Experiment with the mirror read policies to improve performance. For example, using the Solaris Volume Manager software, the default read mode is to alternate reads using a round-robin method among the disks. This mode is the default because it works best for UFS multiuser, multiprocess activity.
- In some cases, the geometric read option improves performance by minimizing head motion and access time. This option is most effective when there is only one slice per disk, when only one process at a time is using the file system, when I/O patterns are sequential, or when all accesses are read.

- Use the `swap -l` command to check for all swap devices. Mirror the slices specified as swap separately.
- Use only similarly configured submirrors within a mirror. In particular, if you create a mirror with an unlabeled submirror, you cannot attach any submirrors that contain disk labels.

RAID 5

RAID-5 volumes are striped volumes that use a distributed parity scheme for data protection. To fully understand RAID-5 volumes, you must understand each of the following:

- Standard RAID-5 volume
- Requirements for RAID-5 volumes
- Suggestions for RAID-5 volumes

Standard RAID-5 Volume

RAID level 5 is similar to striping in that data is distributed across a set of disks. The difference between a RAID level 5 and striping is that in the RAID level 5, parity data is also distributed across the same set of disks. When a disk fails, lost data from the failing disk is rebuilt on the failed volume from the other disks using the distributed data and parity information stored on the remaining (unfailed) disks in the RAID-5 volume.

A RAID-5 volume uses a storage capacity equivalent to one slice to store parity information from the remainder of the RAID-5 volume's slices. The parity information is distributed across all slices in the volume. Like a mirror, a RAID-5 volume increases data availability, but minimizes hardware cost. You cannot use a RAID-5 volume for the root (`/`) directory, the `/usr` directory, swap space, or existing file systems because the metadevice software is not loaded early enough in the Solaris OE boot process.

Figure 8-7 shows that the first three data interlaces are written to slices A, B, and C. The next item written is parity to Drive D. The pattern of writing data and parity results in both data and parity spread across all disks in the RAID-5 volume. You can read each drive independently. The parity protects against a single disk failure. In Figure 8-7, if each disk were 2 Gbytes, the total capacity of the RAID-5 volume would be 6 Gbytes. Parity information occupies the space equivalent to one drive.

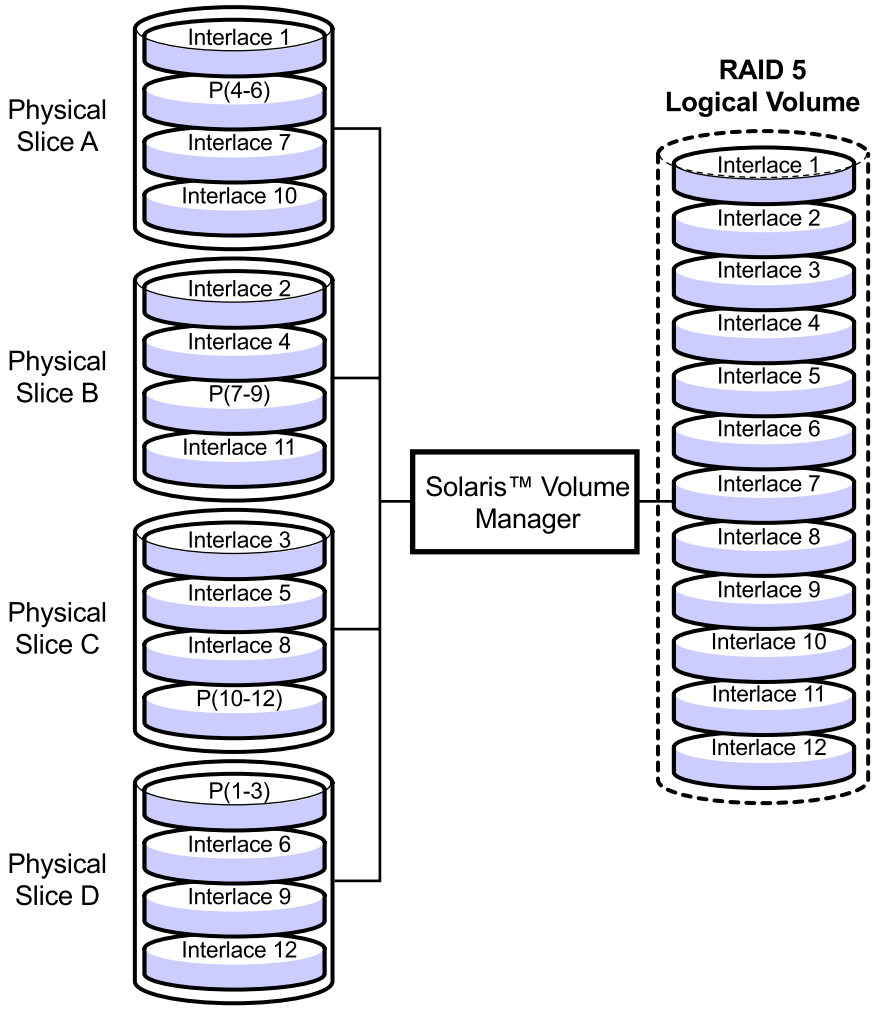


Figure 8-7 RAID-5 Distributed Parity

Requirements for RAID-5 Volumes

The general configuration guidelines for configuring RAID-5 volumes are:

- Create a RAID-5 volume with a minimum of three slices. The more slices a RAID-5 volume contains, the longer read and write operations take when a slice fails.
- Do not stripe, concatenate, or mirror RAID-5 volumes.
- Do not create a RAID-5 volume from a slice that contains an existing file system, because you will erase the data during the RAID-5 initialization process.
- When you create a RAID-5 volume, you can define the interlace value. If you do not specify a value, a default value of 16 Kbytes is assigned.
- A RAID-5 volume (with no hot spares) can only handle a single slice failure.
- To optimize performance, use slices across separate controllers when creating RAID-5 volumes.
- Use disk slices of the same size. Creating a RAID-5 volume of different-sized slices results in unused disk space on the larger slices.

Suggestions for RAID 5 Volumes

The following general suggestions can help avoid common performance problems when using RAID-5 volumes:

- Because of the complexity of parity calculations, volumes with greater than about 20 percent writes should probably not be RAID-5 volumes. If data redundancy on a write-heavy volume is needed, consider mirroring.
- If the slices in the RAID-5 volume reside on different controllers and the accesses to the volume are primarily large sequential accesses, then setting the interlace value to 32 Kbytes might improve performance.

Hardware Considerations

When planning your storage management configuration, keep in mind that for any given application there are trade-offs in performance, availability, and hardware costs. You might need to experiment with the different variables to determine what works best for your configuration. A few categories of information that you must address during the storage planning phase are:

- General storage guidelines
- Determining storage characteristics
- Storage performance guidelines

Storage Characteristics

When you classify storage characteristics, you provide guidelines for working with the Solaris Volume Manager software RAID-0 (concatenation and stripe) volumes, RAID-1 (mirror) volumes, and RAID-5 (striping with distributed parity) volumes.

While building your storage management plan, decide what types of storage devices to use. The storage characteristics guidelines help you compare and contrast the various storage mechanisms and also help you choose the best storage device.

Note – The storage mechanisms listed in Table 8-3 are not mutually exclusive. You can use them in combination to meet multiple goals. For example, you could create a RAID-1 volume for redundancy, and then create soft partitions on it to increase the number of possible discrete file systems.

Table 8-3 Choosing Storage Mechanisms

Feature	RAID-0 Concatenation	RAID-0 Stripe	RAID-1 Mirror	RAID-5 Stripe With Parity
Redundant data	No	No	Yes	Yes
Improved read performance	No	Yes	Depends on the underlying device	Yes
Improved write performance	No	Yes	No	No

You must consider many factors when optimizing redundant storage. Table 8-4 compares RAID-1 and RAID-5 volumes for the speed of write operations, random read operations, and the overall cost of the underlying hardware.

Table 8-4 Optimizing Redundant Storage

Factors	**RAID 1 (Mirror)**	**RAID 5**	**Non-Redundant**
Write operations	Faster	Slower	Neutral
Random read	Slower	Faster	Neutral
Hardware cost	Highest	Higher	Lowest
Performance during failure	Best	Poor	Data loss

General Storage Guidelines

The general configuration guidelines for planning your storage configuration are:

- RAID-0 devices (stripes and concatenations) do not provide data redundancy.
- Concatenation works well for small, random I/O.
- Striping performs well for large, sequential I/O and for random I/O distributions.
- Mirroring improves read performance
- Because of the read-modify-write property of RAID-5 volumes, volumes with greater than about 20-percent writes should probably not be RAID 5. In these write intensive situations, consider mirroring if data protection is required.
- RAID 5 writes are not as fast as mirrored writes, and mirrored writes are not as fast as unprotected writes.

Performance Guidelines

When designing your storage configuration, consider the following performance guidelines:

- Whenever possible, distribute storage devices across multiple I/O controllers, cables, and devices.
- Striping generally has the best performance, but it offers no data protection. For write-intensive applications, RAID 1 performs better than RAID 5.
- RAID-1 and RAID-5 volumes both increase data availability. Mirroring improves random read performance.
- RAID-5 volumes have a lower hardware cost than RAID-1 volumes, while RAID-0 volumes have no additional hardware cost.
- Identify the most frequently accessed data, and increase the access bandwidth for that data with mirroring or striping.
- Both stripes and RAID-5 volumes distribute data across multiple disk drives and help balance the I/O load. You can also use RAID-1 volumes to help balance the I/O load.
- Use available performance monitoring capabilities and generic tools, such as the `iostat` command, to identify the most frequently accessed data. Then increase the “access bandwidth” to the frequently accessed data, by striping RAID-1 volumes or RAID-5 volumes.
- A stripe’s performance is better than that of a RAID 5 volume, but stripes do not provide data redundancy.
- RAID 5 volume performance is lower than stripe performance for write operations, because the RAID-5 volume requires multiple I/O operations to calculate and store the parity.
- For raw random I/O reads, the stripe and the RAID-5 volume are comparable. Both the stripe and RAID-5 volume split the data across multiple disks, and the RAID-5 volume parity calculations are not a factor in reads, except after a component failure.

Introducing Solaris Volume Manager Software Concepts

The Solaris Volume Manager software lets you manage large numbers of disks and the data on those disks. Although there are many ways to use the Solaris Volume Manager software, most tasks include:

- Increasing storage capacity
- Increasing data availability
- Making the administration of large storage devices easier

In some instances, the Solaris Volume Manager software can also improve I/O performance.

Logical Volume

The Solaris Volume Manager software uses virtual disks called logical volumes to manage physical disks and their associated data. Historically, a logical volume is functionally identical to a physical slice. However, a logical volume can span multiple disk *members*. The Solaris Volume Manager software converts I/O requests directed at a volume into I/O requests to the underlying member disks.

You can create the Solaris Volume Manager software volumes from slices (disk partitions) or from other Solaris Volume Manager software volumes. An easy way to create volumes is to use the GUI built into the Solaris™ Management Console. The Enhanced Storage tool within the Solaris Management Console lists all the existing volumes. By following the steps in the tool wizard, you can create any type of Solaris Volume Manager software volumes or components. You can also build and modify volumes using command-line utilities in the Solaris Volume Manager software.

To create more storage capacity as a single volume, you can use the Solaris Volume Manager software to make the system treat a collection of many small slices as one large slice or device. After creating a large volume from these slices, you can immediately begin by using it just as any other slice or device.

The Solaris Volume Manager software can increase the reliability and availability of data by using RAID-1 volumes and RAID-5 volumes. Solaris Volume Manager software hot spares provide another level of data availability for RAID-1 volumes and RAID-5 volumes.

Note – In earlier versions of the Solaris OE, the Solaris Volume Manager software was known as Solstice DiskSuite™ software, and logical volumes were known as metadevices. Most of the associated command-line tools begin with the prefix *meta*. Logical devices are located under the `/dev/md` directory.

After setting up your configuration, use the Enhanced Storage tool within the Solaris Management Console to display information about the volume.

Soft Partitions

As disks become larger, and disk arrays present larger logical devices to Solaris OEs, users must be able to subdivide disks or logical volumes into more than eight sections, often to create manageable file systems or partition sizes.

Soft partitions provide a mechanism for dividing large storage spaces into smaller, more manageable, sizes. For example, large storage aggregations provide redundant storage of many gigabytes, but many scenarios would not require as much space. Soft partitions allow you to subdivide that storage space into more manageable sections, each of which can have a complete file system.

For example, you could create 1000 soft partitions on top of a RAID-1 volume or RAID-5 volume so that each of your users can have a home directory on a separate file system. If a user needs more space at a later date, you can grow the soft partition.

Note – The Solaris Volume Manager software can support up to 8192 logical volumes per disk set, but is configured for 128 (d0–d127) by default. For instructions on increasing the number of logical volumes, refer to *Solaris Volume Manager Administration Guide*, Part Number 806-6111-10

Use soft partitioning to divide a slice or volume into as many divisions as needed. Assign a name for each division or soft partition, just like you would do for other storage volumes, such as stripes or mirrors. A soft partition, once named, can be directly accessed by applications, including file systems, as long as it is not included in another volume.

When you partition a disk and build a file system on the resulting slices, you cannot later extend a slice without modifying or destroying the disk format. With soft partitions, you can extend portions up to the amount of space on the underlying device without moving or destroying data on other soft partitions.

Suggestions for Soft Partitioning

Consider the following factors when implementing soft partitions in your storage environment:

- You can build soft partitions on any slice. Creating a single slice that occupies the entire disk and then creating soft partitions on that slice is the most efficient way to use soft partitions at the disk level.
- To expand and manage storage space, build stripes on top of your disk slices, and then build soft partitions on the stripes.
- You can grow soft partitions to use any available space on a volume.
- Create a RAID-1 volume or a RAID-5 volume, and then create soft partitions on the RAID 1 volume or RAID-5 volume for maximum flexibility and higher availability.

Introducing the State Database

Before creating volumes using the Solaris Volume Manager software, state database replicas must exist on the Solaris Volume Manager software system. The state database stores information on disk about the state of your Solaris Volume Manager software configuration. The state database records and tracks changes made to your configuration. The Solaris Volume Manager software automatically updates the state database when a configuration or state change occurs. For example, creating a new volume is a configuration change, while a submirror failure is a state change. This section addresses the following:

- The Solaris Volume Manager software state database
- Recommendations for state database replicas
- Suggestions for state database replicas

The Solaris Volume Manager Software State Database

The state database is a collection of *multiple*, *replicated* database copies. Each copy, called a state database replica, ensures that the data in the database is always valid. Having copies of the state database protects against data loss from single points-of-failure. The state database tracks the location and status of all known state database replicas. During a state database update, each replica state database is updated. The updates take place one at a time to protect against corrupting all updates if the system crashes.

The Solaris Volume Manager software state database contains configuration and status information for all volumes and hot spares. The Solaris Volume Manager software maintains replicas (copies) of the state database to provide redundancy and to prevent database corruption during a system crash.

If your system loses a state database replica, Solaris Volume Manager software must determine which state database replicas still contain non-corrupted data. The Solaris Volume Manager software determines this information by a majority consensus algorithm. This algorithm requires that a majority (half + 1) of the state database replicas be available and in agreement with each other before any of them are considered non-corrupt. Because of the majority consensus algorithm, you should create at least three state database replicas when you set up your disk configuration. A consensus can be reached as long as at least two of the three state database replicas are available.

During booting, the Solaris Volume Manager software ignores corrupted state database replicas. In some cases, Solaris Volume Manager software tries to rewrite state database replicas that are corrupted. Otherwise the databases are ignored until you repair them. If a state database replica becomes corrupt because its underlying slice encountered an error, you must repair or replace the slice, and then recreate the replica.

If all state database replicas are lost, you could lose all data that is stored on your Solaris Volume Manager software volumes. You should create enough state database replicas on separate drives and across controllers to prevent complete data loss. You should also save your initial configuration information, as well as your disk partition information.

To protect data, the Solaris Volume Manager software will not function unless half of all state database replicas are available. The main functions of the majority consensus algorithm are:

- The system will stay running if at least half of the state database replicas are available.
- The system will panic if fewer than half the state database replicas are available.
- The system will not start the Solaris Volume Manager software unless a majority (half + 1) of the total number of state database replicas are available.

Recommendations for State Database Replicas

To avoid single points-of-failure, you should distribute state database replicas across slices, drives, and controllers. A majority of replicas must survive a single component failure. The Solaris Volume Manager software requires that half the replicas be available to run, and that a majority (half + 1) be available to boot. If you lose a replica (for example, due to a device failure), you might run into problems when running Solaris Volume Manager software or when rebooting the system. When working with state database replicas, consider the following:

- You should create state database replicas on a dedicated slice of at least 4 Mbytes per replica.
- You can put replicas on unused slices, and then use them on RAID-0, RAID-1, or RAID-5 volumes.
- You can only create state database replicas on slices that are not in use.
- You cannot create state database replicas on any slices in use.
- A minimum of three state database replicas are recommended. The following guidelines are recommended:
  - For a system with only a single drive: put all three replicas in one slice.
  - For a system with two to four drives: put two replicas on each drive.
  - For a system with five or more drives: put one replica on each drive.
- Make sure that you have at least two extra replicas per mirror.
- You can add additional state database replicas to the system at any time. The additional state database replicas help to ensure the Solaris Volume Manager software's availability.

Caution – If you upgraded from Solstice DiskSuite software to Solaris Volume Manager software and have state database replicas at the beginning of slices (as opposed to on separate slices), do not delete existing replicas and replace them with new ones in the same location. The default Solaris Volume Manager software state database replica size is 8192 blocks, while the default size in Solstice DiskSuite software was 1034 blocks. If you delete a default-size state database replica from Solstice DiskSuite software, and add a new default-size replica with the Solaris Volume Manager software, you will overwrite the first 7158 blocks of any file system occupying the rest of the shared slice, which destroys the data.

Introducing Hot Spares and Hot Spare Pools

Hot spares and hot spare pools provide additional physical slices for automatic recovery from RAID-1 mirror or RAID-5 volume failures.

Hot Spares

A hot spare is a slice (not a volume) that is functional and available, but not in use. A hot spare is on reserve to substitute for a failed slice in a submirror or RAID-5 volume. You cannot use a hot spare to hold data or state database replicas until the hot spare is assigned as a member. A hot spare must be ready for immediate use in the event of a slice failure in the volume with which it is associated. To use hot spares, invest in additional disks beyond those that the system requires to function.

Hot Spare Pools

A hot spare pool is a collection of slices. The Solaris Volume Manager software uses hot spare pools to provide increased data availability for RAID-1 volumes and RAID-5 volumes. The Solaris Volume Manager software reserves a hot spare for automatic substitution when a slice failure occurs in either a submirror or a RAID-5 volume.

Note – Hot spares do not apply to RAID-0 volumes or to one-way mirrors. For automatic substitution to work, redundant data must be available.

Module 9

Configuring Solaris Volume Manager Software

Objectives

The Solaris Volume Manager software provides mechanisms to configure physical slices of data into logical volumes. Logical volumes can be configured to provide data redundancy or to produce performance enhancements. The Solaris Volume Manager software also maintains a state database used to track the configuration and status of the volumes being used on the system. The Solaris Volume Manager software provides a GUI called the Enhanced Storage Tool to perform volume management tasks. You can also use the command line to perform volume management tasks.

Upon completion of this module, you should be able to:

- Distribute the state database replicas
- Build a mirror of the root (/) file system

The following course map shows how this module fits into the current instructional goal.

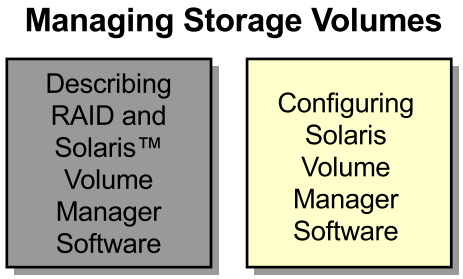


Figure 9-1 Course Map

Distributing the State Database Replicas

The state database contains configuration and status information for all volumes and hot spares. Multiple copies of the database, called replicas, provide redundancy and protect against data loss if a copy of the database is corrupted due to the system crashing or other types of failure. When configuring the state database, configure multiple copies of the state database, and distribute the state database replicas across multiple disks so that failure of a single disk only causes the loss of a single state database replica.

If the system loses a state database replica, Solaris Volume Manager software uses a majority consensus algorithm to determine which state database replicas still contain valid data. The algorithm requires that a majority (half +1) of the state database replicas are available and in agreement with each other, before any of them are considered valid. The majority consensus algorithm requires that you create at least three state database replicas before you build or commit any volumes. To reach a consensus, at least two of the three replicas must be available.

The majority consensus algorithm:

- Makes sure that the system stays running if at least half of the state database replicas are available.
- Causes the system to panic if fewer than half of the state database replicas are available.
- Prevents the system from starting the Solaris Volume Manager software unless a majority of the total number of state database replicas are available.

If insufficient state database replicas are available, you must boot into single-user mode and delete enough of the corrupt replicas to achieve a majority consensus.

Creating the State Database

You can create state database replicas by using:

- The `metadb -a` command
- The Solaris Volume Manager software GUI

Creating the State Database Using the Command Line

To create state database replicas by using the command line, use the `metadb` command. The syntax of the command is:

```
metadb -a [-f] [-c n] [-l nnnn] disk_slice
```

where:

`-a`	Adds a state database replica.
`-f`	Forces the operation, even if no replicas exist. Use this flag to force the creation of the initial replicas.
`-c n`	Specifies the number of replicas to add to the slice.
`-l nnnn`	Specifies the size of the new replicas, in blocks.
`disk_slice`	Specifies the name of the *`disk_slice`* that will hold the replica.

Note – The `metadb` command without options reports the status of all replicas.

The following example shows the creation of state database replicas:

```
# metadb -a -f c1t1d0s0 c1t1d0s1 c1t1d0s3
# metadb
        flags           first blk       block count
     a        u         16              8192            /dev/dsk/c1t1d0s0
     a        u         16              8192            /dev/dsk/c1t1d0s1
     a        u         16              8192            /dev/dsk/c1t1d0s3
```

This example lists the three replicas that were just created. Each replica begins at block 16 of the assigned disk slice. Each replica is 8192 blocks (or 4 Mbytes in size). The flags indicate that the replica is active and up to date.

Note – The previous example places the state database replicas on different slices within the same disk. In a production environment, you should distribute the replicas across multiple disks.

Creating the State Database Using the Solaris Management Console

The Enhanced Storage Tool within the Solaris Management Console software in Solaris Management Console Welcome Screen provides a GUI that guides you through Solaris Volume Manager software tasks.

Complete the following steps to create the state database replicas:

1. To start the Solaris Management Console, perform the command:

```
# smc &
```

The Solaris Management Console appears, as shown in Figure 9-2.

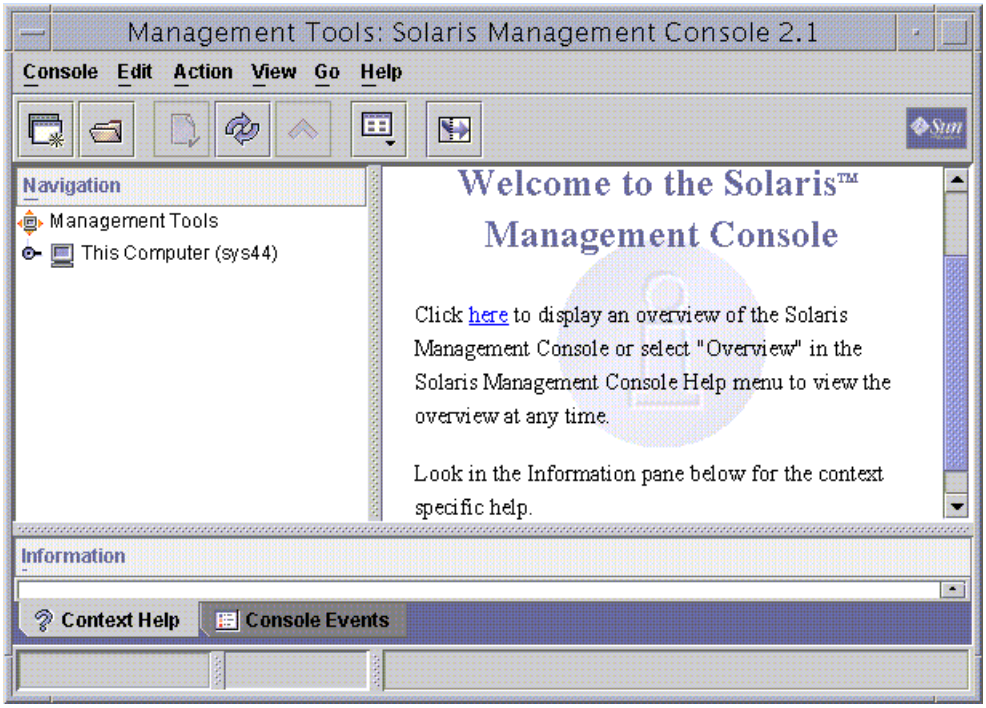


Figure 9-2 Solaris Management Console Welcome Screen

2. Use the Navigation pane to traverse the Solaris Management Console structure until you reach the Enhanced Storage Tool.
3. Click This Computer.
4. Select Storage.

5. Click Enhanced Storage, as shown in Figure 9-3, to display the contents of the Enhanced Storage Tool.

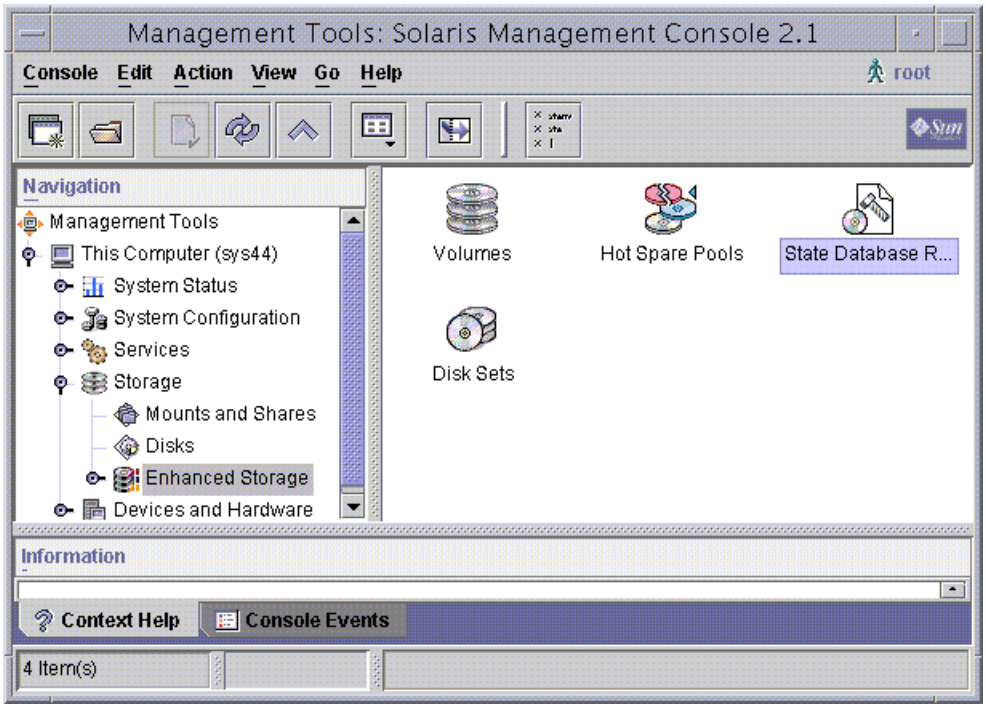


Figure 9-3 Solaris Management Console: Storage Tool

Note – After you start the Solaris Management Console, you must log in after you open the first tool.

6. Click the State Database Replica icon.

If the state database currently contains replicas, these replicas appear in the View pane. If no state database replicas exist, the View pane is empty, as shown in Figure 9-4.

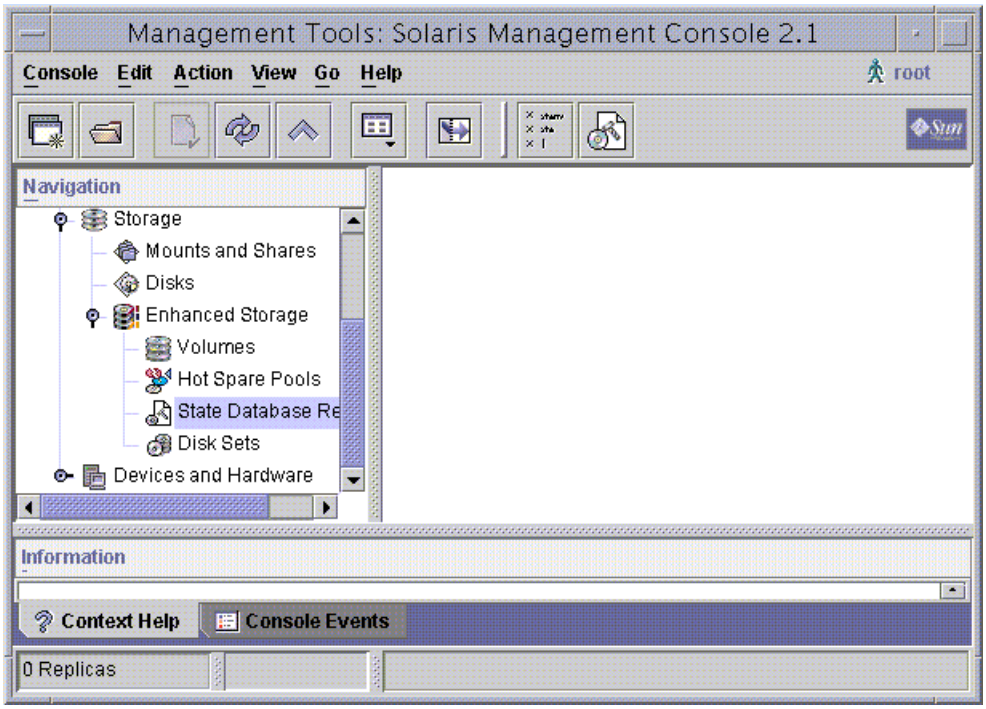


Figure 9-4 Solaris Management Console: View Pane

7. To create a replica, select Create Replicas from the Action menu, as shown in Figure 9-5, and follow the instructions.

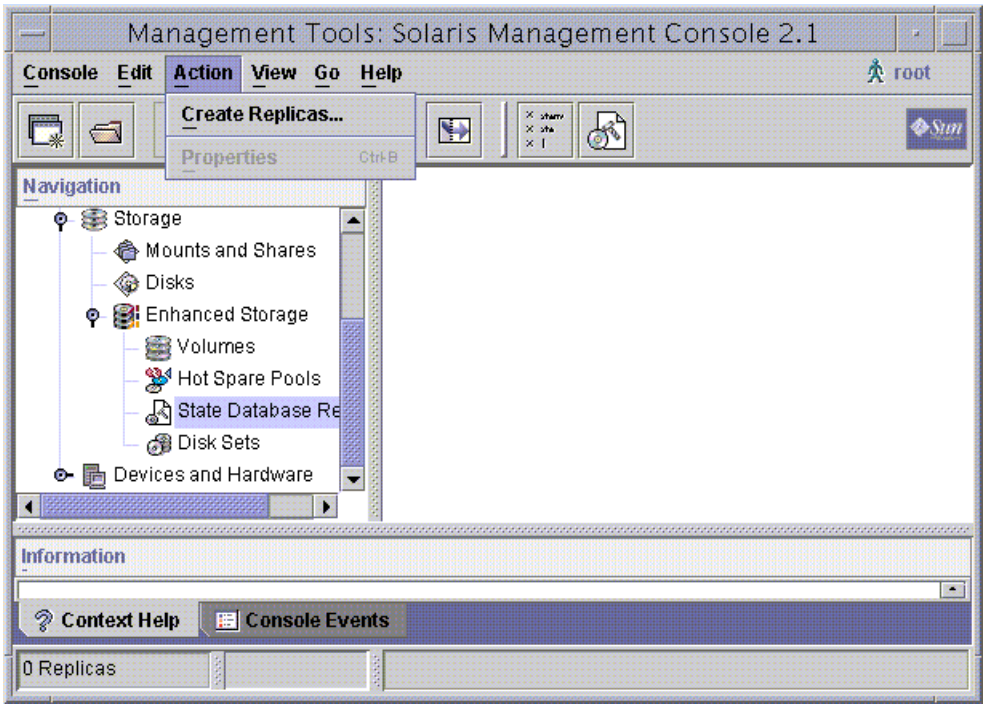


Figure 9-5 Solaris Management Console Window – Action Menu

A series of windows guide you through the creation of the state database.

8. Select alternate disk sets when additional disk sets are available, as shown in Create Replicas: Select Disk Sets Window (Figure 9-6). In this configuration, no additional disk sets have been configured, so choose the default selection of <none>.

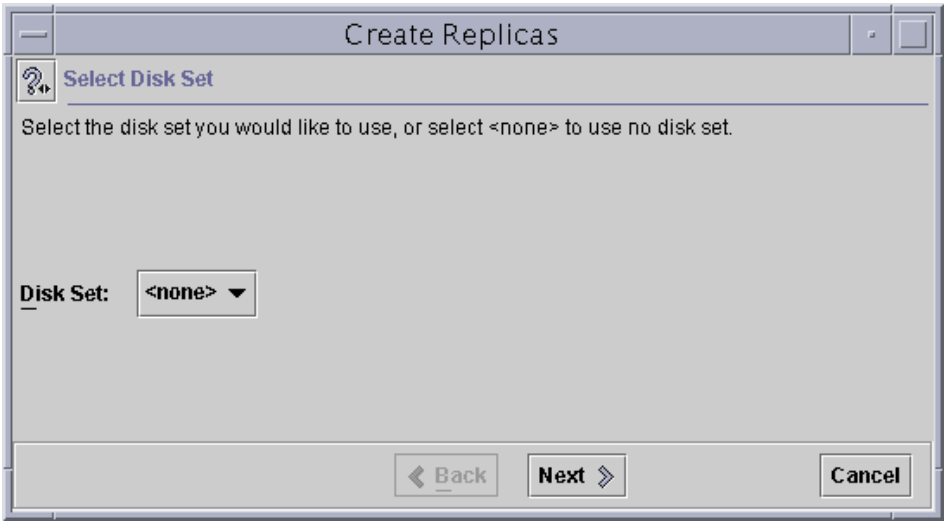


Figure 9-6 Create Replicas: Select Disk Sets Window

Note – A disk set is a set of shared disk drives that contain logical Volume Manager objects that can be shared exclusively but not concurrently by one or two hosts. Disk sets are enablers for host fail-over scenarios.

9. Click Next to continue.

Note – Disk sets are described in ES-220: *Disk Management With DiskSuite*.

When you choose disk slices on which to store the state database replicas, select at least three slices. The Create Replicas: Select Components Window (Figure 9-7) shows that you can choose to configure as many slices as are required by the size of your system's disk configuration. The size of these disk slices are pre-set using the partitioning mechanism of the `format` utility.

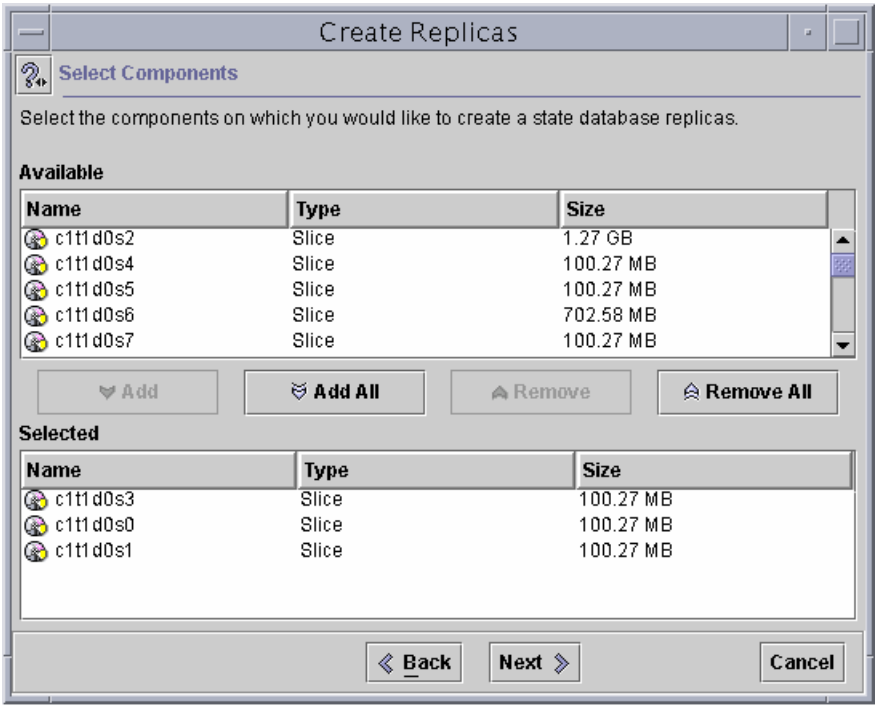


Figure 9-7 Create Replicas: Select Components Window

10. Select a slice.
11. Click Add.
12. Continue adding slices until all the necessary slices are selected.

Note – Alternatively, to select multiple slices, hold down the Control key while you make your selections.

13. Click Next to continue.

The default size of each replica is 8192 blocks or 4 Mbytes. The Create Replicas: Select Components Window, as shown in Figure 9-8, enables you to increase the size of the replicas and the number of replicas per slice.

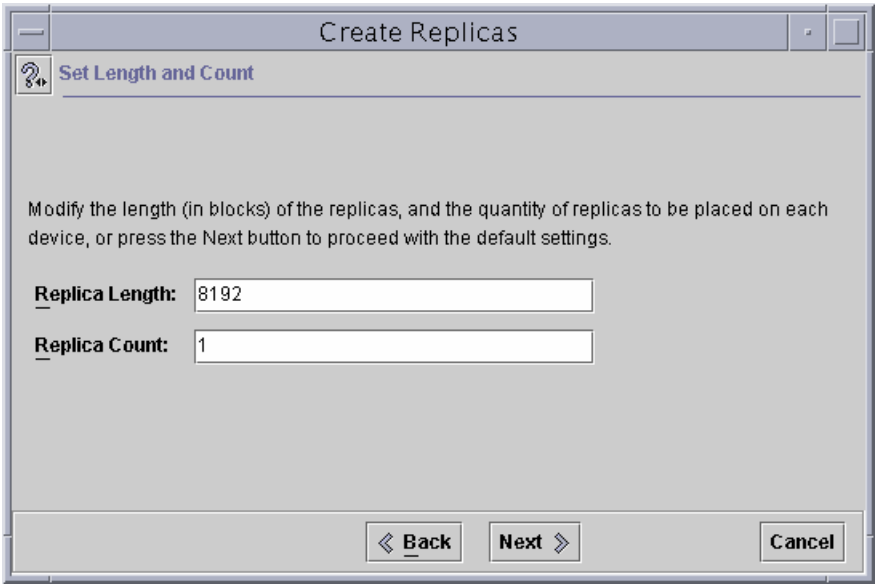


Figure 9-8 Create Replicas: Set Length and Count Window

14. Unless equipment limitations force you to assign multiple replicas to a device, accept the default replica count of 1.
15. Click Next to continue.

The Create Replicas: Review Window window shows the selections you have chosen for your state database replicas, as shown in Figure 9-9. Additionally, this window shows the commands that the Storage Volume Manager uses to build your selected configuration.

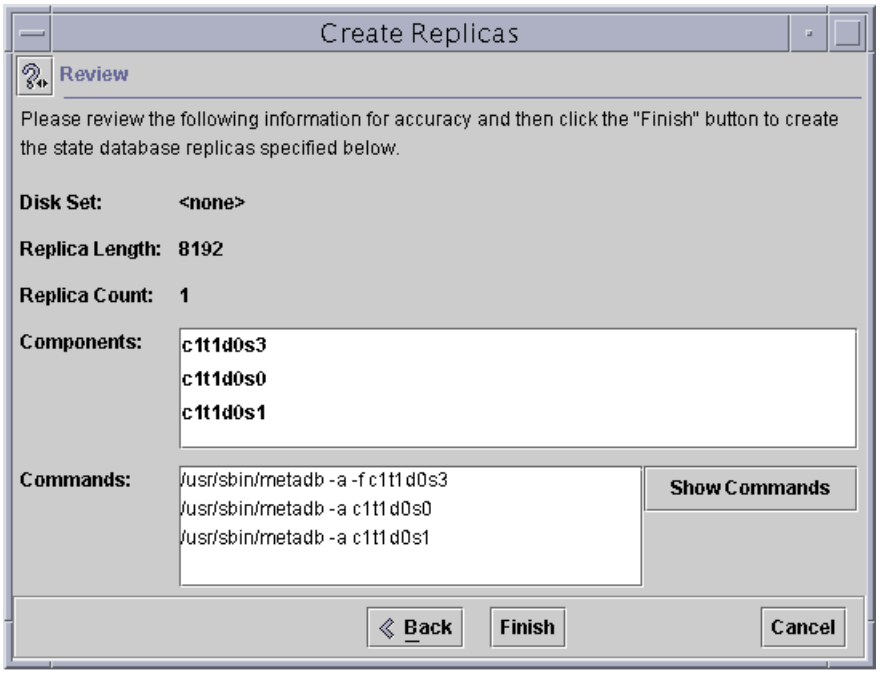


Figure 9-9 Create Replicas: Review Window

16. Double-check your selections to ensure that they meet the criteria of your state database replicas.

Note – Before you click Finish, click Show Commands to view and, optionally, log the commands used to accomplish the specified Enhanced Storage Tool operations.

17. Click Finish to complete the operation.

The Solaris Management Console: New State Database Replicas Window shows that the newly configured state database replicas appear in the View pane of the Solaris Management Console.

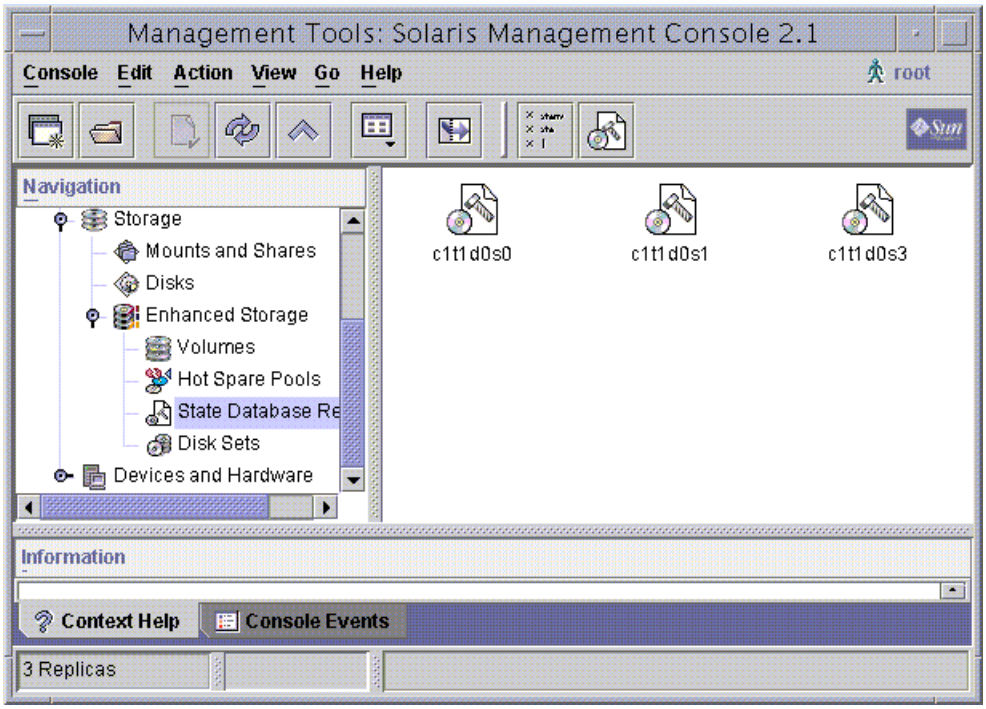


Figure 9-10 Solaris Management Console: New State Database Replicas Window

If the three replicas are configured on separate disks, the three replicas configure the system to tolerate a single disk failure and still maintain the majority consensus algorithm. The majority consensus algorithm is necessary for the system to remain running or for it to reboot to multiuser mode when required.

Note – The configuration represented in this example does not follow industry best practices. State database replicas should be distributed across multiple devices and disk controllers wherever possible.

Building a Mirror of the Root (/) File System

RAID-1 volumes, also known as mirrors, are a way to construct redundant volumes, in which a partial or complete failure of one of the underlying RAID-0 volumes does not cause data loss or interruption of access to the file systems. This section describes how to create a RAID-1 volume from the root (/) file system, which cannot be unmounted. To create a mirror, perform the following steps:

1. Create a RAID-0 volume on the file system you want to mirror.
2. Create a second RAID-0 volume that will contain the second submirror of the RAID-1 volume.
3. Create a one-way mirror using the RAID-0 volume that contains the file system to be mirrored.
4. Use the `metaroot` command to update the system's configuration.
5. Reboot your system.
6. Attach the second submirror to the file system mirror.
7. Record the alternate boot path that will be used in the event of a failure of the primary submirror, because this is a mirror of the root (/) file system.

The following scenario assumes the root (/) file system is initially stored on disk slice `c0t0d0s0`. A RAID-0 volume is created named `d11` on slice `c0t0d0s0`. A second RAID-0 volume is created as metadevice `d12` from a spare disk slice at `c1t2d0s1`. A RAID 1 volume is created named `d10` using the RAID-0 volumes named `d11` and `d12`, as shown in Figure 9-11.

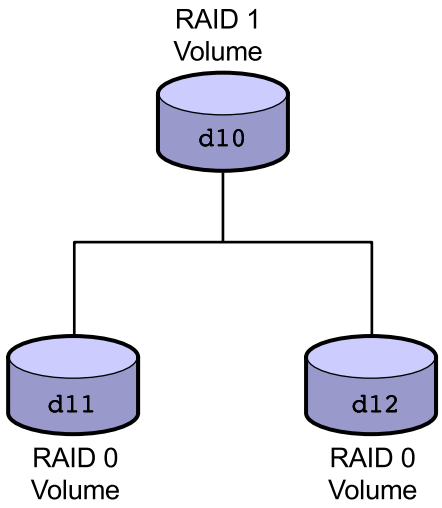


Figure 9-11 Mirror of Root (/) Partition

Creating a RAID 0 Volume

The first step when building a mirror of the root (/) file system is to create RAID-0 volumes, which you will later combine to form the mirror. Each RAID-0 volume becomes a submirror to the mirror. Use the `metainit` command to force the creation of the RAID-0 volume. The force (`-f`) option must be used because this is the root (/) file system, which cannot be unmounted.

The syntax of the `metainit` command is:

```
metainit -f concat/stripe numstripes width component...
```

where:

`-f`	Forces the `metainit` command to continue, even if one of the slices contains a mounted file system or is being used as swap space. This option is useful when configuring mirrors or concatenations on root (/), `swap`, and `/usr` file systems.
`concat/stripe`	Specifies the volume name of the concatenation or stripe being defined.
`numstripes`	Specifies the number of individual stripes in the metadevice. For a simple stripe, *`numstripes`* is always 1. For a concatenation, *`numstripes`* is equal to the number of slices.
`width`	Specifies the number of slices that make up a stripe. When the *`width`* is greater than 1, the slices are striped.
`component`	Specifies the logical name for the physical slice (partition) on a disk drive, such as `/dev/dsk/c0t0d0s1`.

The following example shows how to use the `metainit` command to create a RAID-0 volume:

```
# /usr/sbin/metainit -f d11 1 1 c0t0d0s0
d11: Concat/Stripe is setup
```

Caution – If encapsulating an existing file system in a RAID-0 volume, both the *numstripes* and *width* arguments must be `1`, or the data will be lost.

The command line forces the creation of volume `d11`. Volume `d11` creates a concatenation composed of a single stripe, one slice wide, and it is stored on the `/dev/dsk/c0t0d0s0` disk slice.

Note – In this example, the root (/) file system is stored on the disk slice `/dev/dsk/c0t0d0s0`. Because the root (/) file system is stored at that location, you must use of the `-f` option to force the creation of a volume on the mounted partition.

To create an additional RAID-0 volume, for the secondary submirror of the root file system, use the Enhanced Storage Tool within the Solaris Management Console.

To create additional volumes, complete the following steps:

1. Click the Volumes icon

   Any configured metadevice volumes appear on the View pane, as shown in Figure 9-12. If there are no metadevice volumes currently configured, the View pane remains empty. .

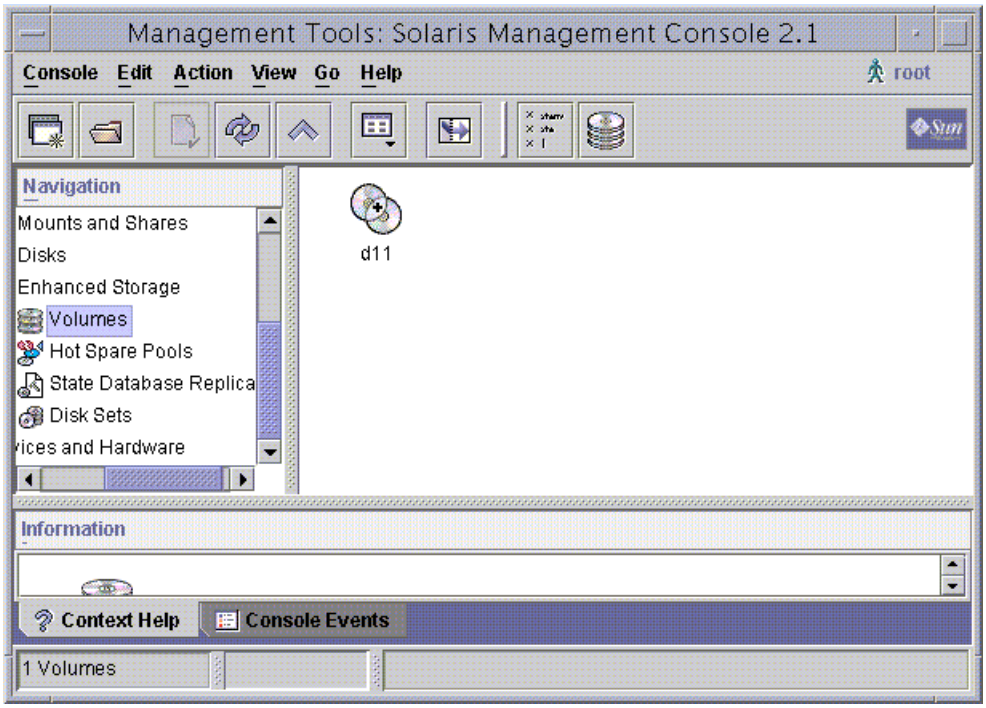


Figure 9-12 Volumes Icon

2. Select Create Volume from the Action menu, as shown in Figure 9-13.

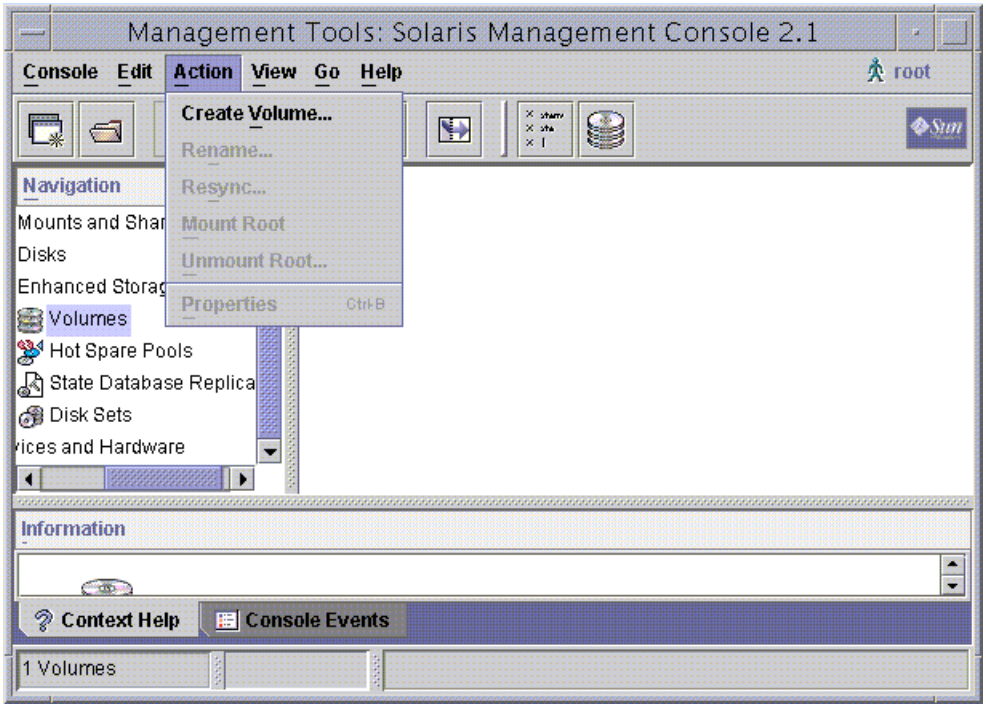


Figure 9-13 Solaris Management Console: Action Menu

3. Answer the prompts in the Create Volume Wizard window.

Every time you create a new volume, you can create additional state database replicas. When creating RAID-0 volumes, it is usually unnecessary to create additional state database replicas.

4. Select Don't Create State Database Replicas in the Create Volume window, as shown in Figure 9-14.

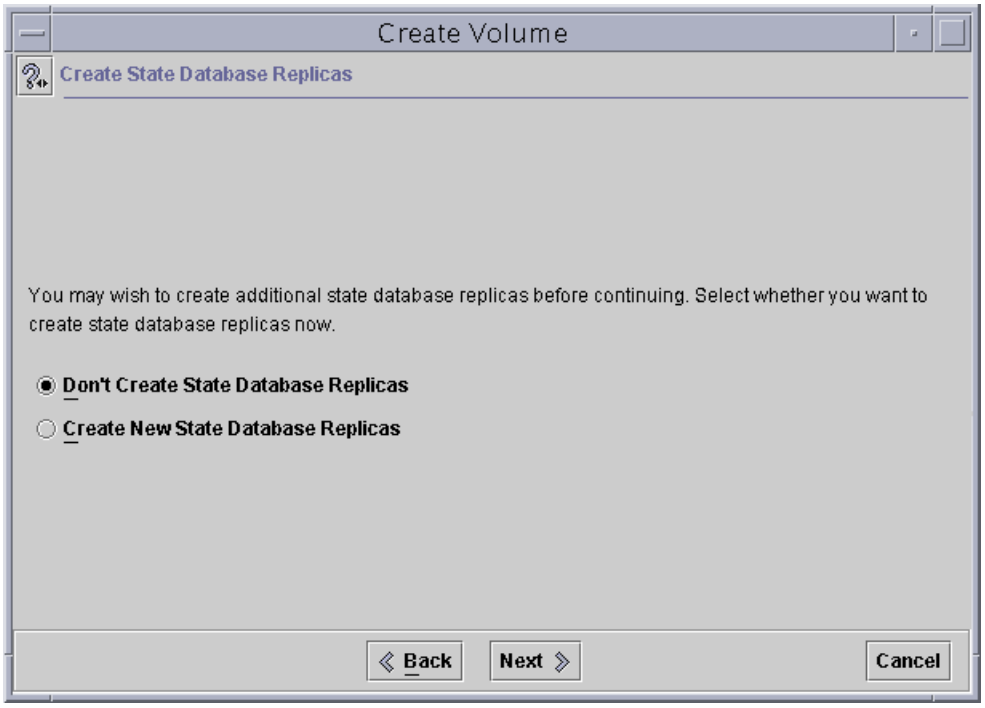


Figure 9-14 Create Volume Window

5. Click Next to continue.

Every time you create a new volume, as shown in Figure 9-15, you can relocate it on alternate disk sets.

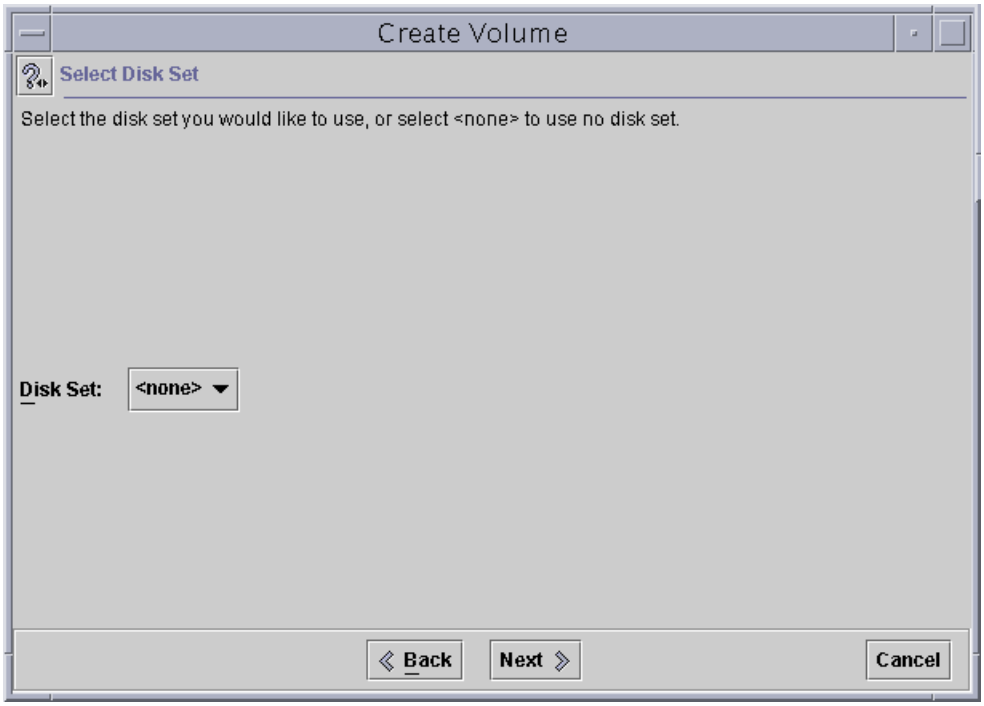


Figure 9-15 Create Volume: Select Disk Set Window

6. If only one disk set exists on the system, select the default of <none>.
7. Click Next to continue.

Figure 9-16 shows a selection of volume configurations that you can create.

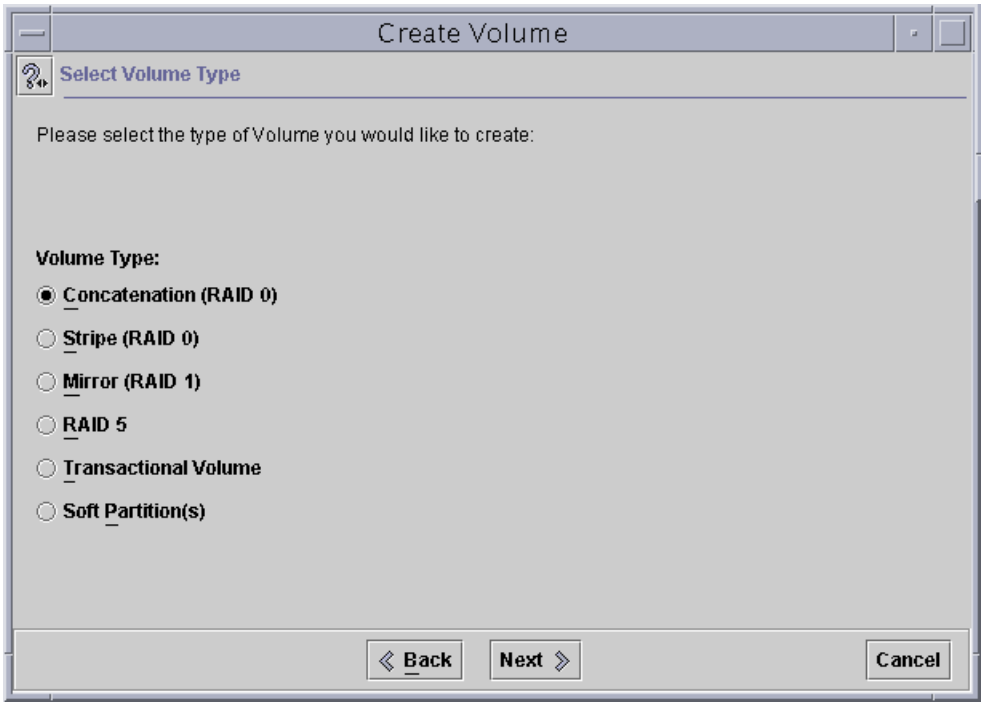


Figure 9-16 Create Volume: Select Volume Type Window

8. Select Concatenation (RAID 0).
9. Click Next to continue.

You can name the volume, as shown in Figure 9-17. By default, volume names fall within the range of `d0` through `d127`. In this procedure, build a mirror named `d10`. The two submirrors that will comprise the mirror are `d11` (for the first submirror) and `d12` (for the second submirror). You have already created volume `d11` from the slice that contains the root (/) file system, so this one is volume `d12`, which will contain the mirror of the root (/) file system.

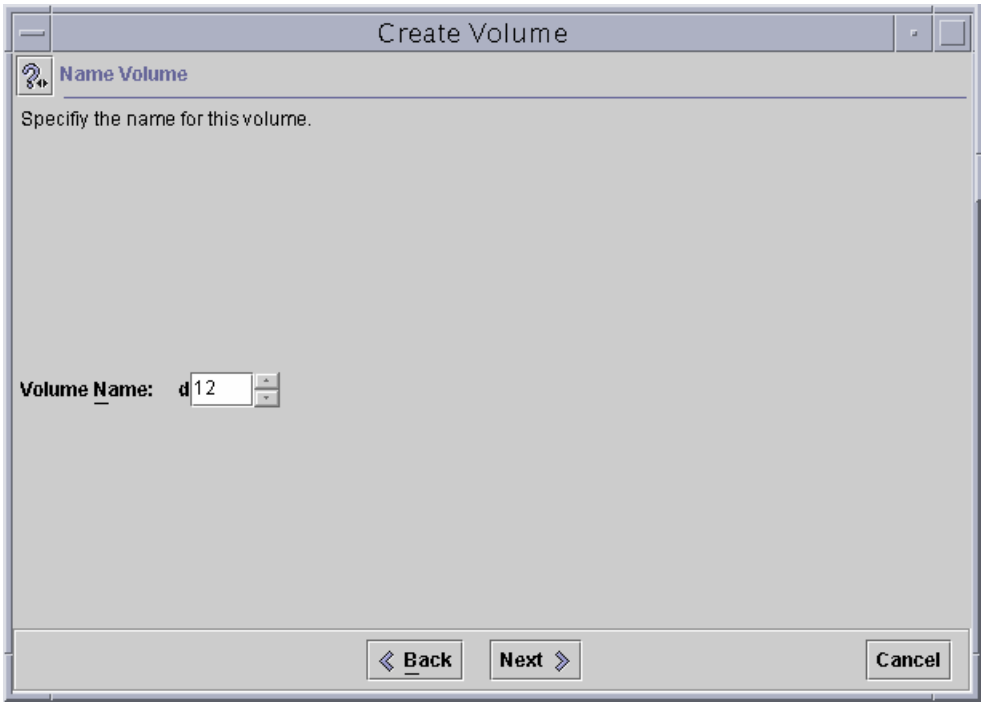


Figure 9-17 Create Volume: Name Volume Window

10. Name the volume `d12`.
11. Click Next to continue.

You can also select a slice that the new volume will occupy, as shown in Figure 9-18. This volume is the secondary submirror of a mirror, therefore the size of this slice must be equal to or greater than the size of the primary submirror of the mirror.

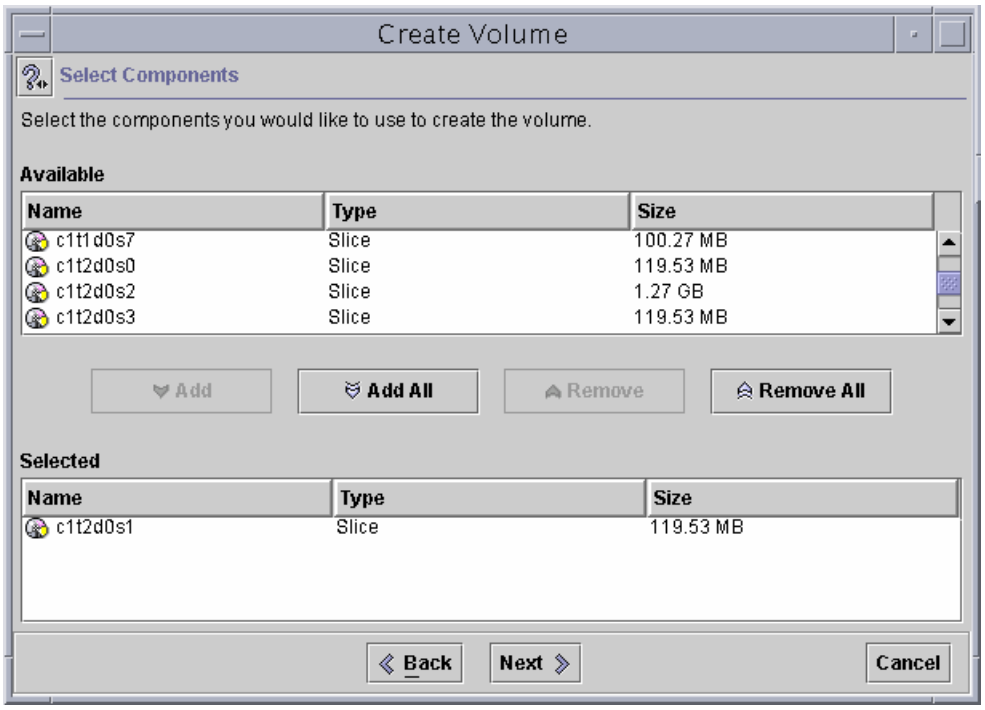


Figure 9-18 Create Volume: Select Components Window

12. Select a slice equal to or greater than the size of the primary submirror RAID-0 volume.
13. Click Add to move it to the Selected list.
14. Click Next to continue.

You can select the order of presentation of the slices within the stripe group, if you are mirroring a file system that can span multiple slices, as shown in Figure 9-19.

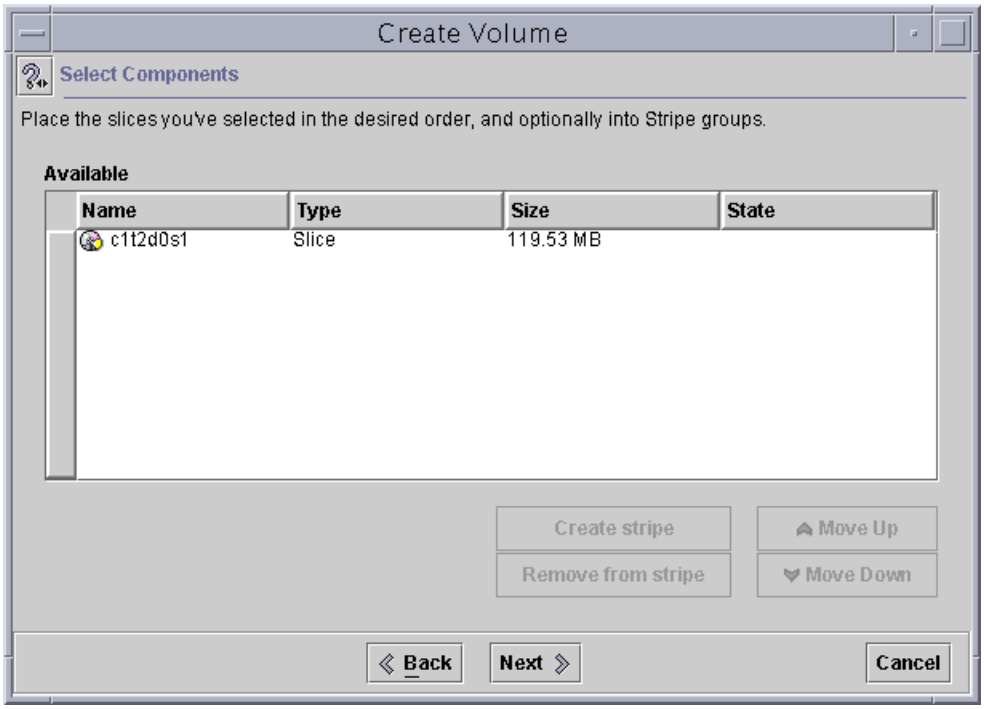


Figure 9-19 Create Volume: Select Components Window

Note – When mirroring root (/), you cannot span multiple slices.

15. Click Next to continue.

A hot spare pool is a set of slices you can use to improve the fault tolerance of the system. To allow continued data accesses to a failed volume until you can replace a failed slice, hot spares are automatically swapped in to replace the failed slice. After replacing the failed slice, the hot spare is automatically swapped back onto the replacement slice, as shown in Figure 9-20

16. Because no hot spare pools have been created, select No Hot Spare Pool.

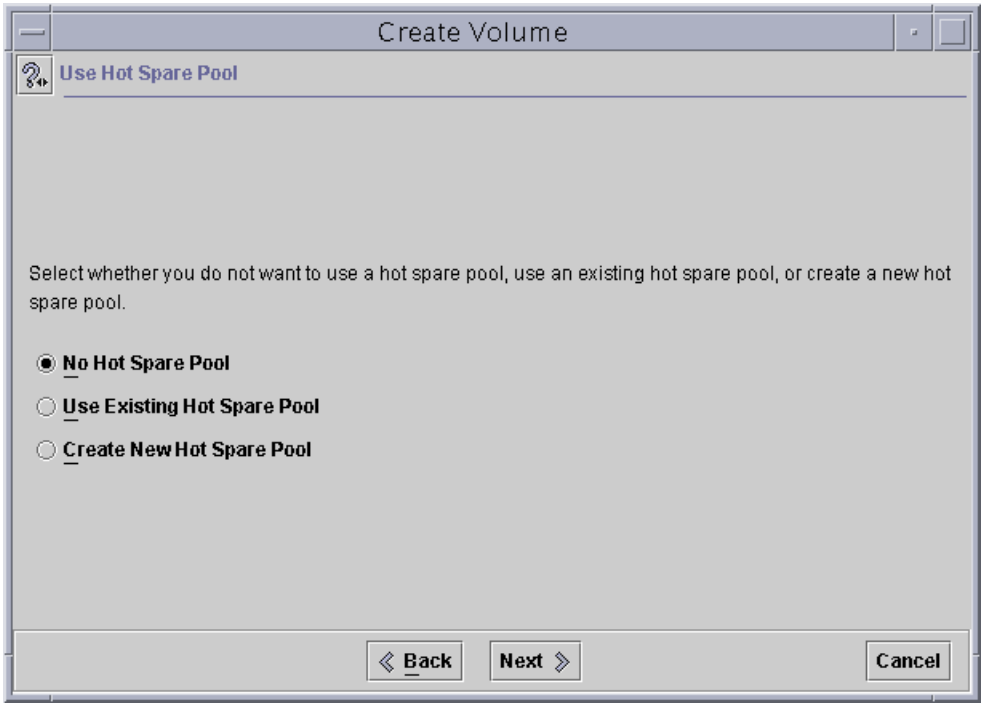


Figure 9-20 Create Volume: Use Hot Spare Pool Window

17. Click Next to continue.

The Create Volume: Review Window window provides a confirmation of your selections. It also provides a summary of the commands necessary to accomplish the identical task from the command line, as shown in Figure 9-21

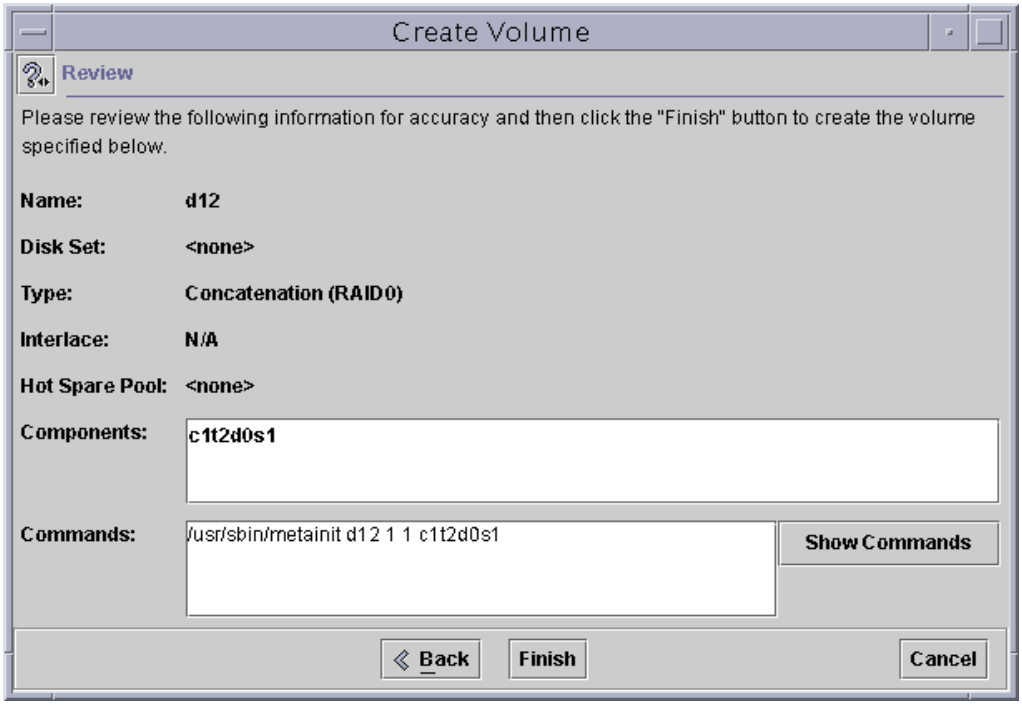


Figure 9-21 Create Volume: Review Window

18. Click Finish.

Figure 9-22 shows the metadevice for the newly created RAID-0 volume.

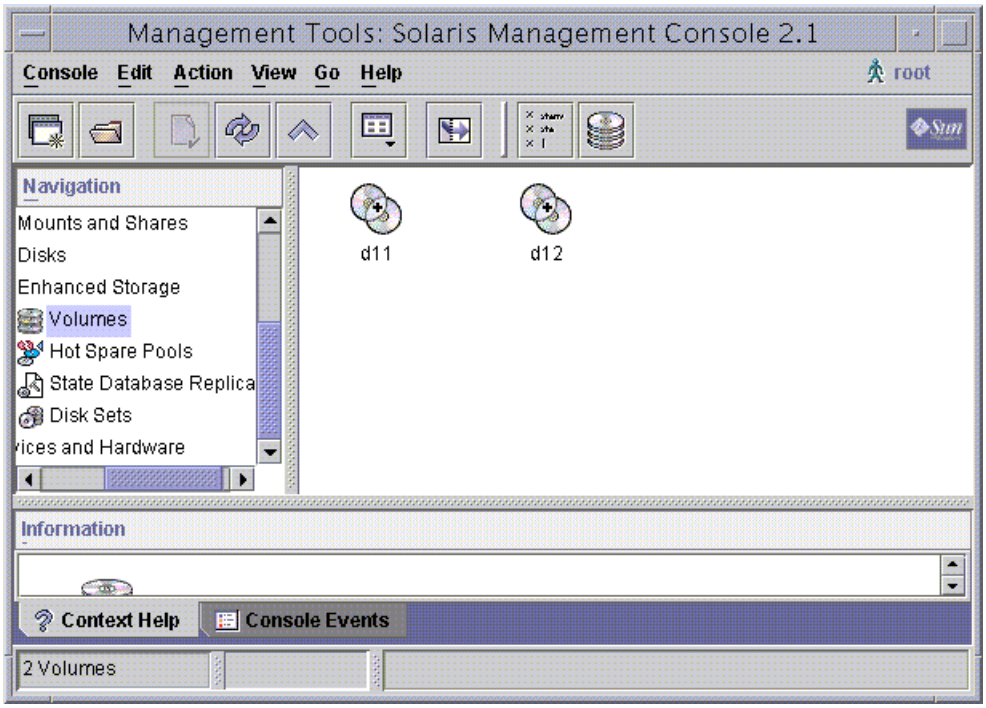


Figure 9-22 Solaris Management Console: Volumes Window

In this procedure, you created two RAID-0 volumes, `d11` and `d12`. The `d11` volume contains the slice where the root (/) file system is stored, and the `d12` volume contains space for a copy of the root (/) file system.

Creating a RAID-1 Volume

You can create the RAID-1 volume using:

- The `metainit` command
- The Enhanced Storage Tool within the Solaris Management Console

The `metainit` Command

The syntax for creating a RAID-1 volume by using the `metainit` command is:

```
metainit mirror -m submirror [read_options] [write_options] [pass_num]
```

where:

mirror `-m` *submirror*	Specifies the volume name of the mirror. The `-m` indicates that the configuration is a mirror. Submirror is a volume (stripe or concatentation) that makes up the initial one-way mirror.
read_options	The following read options for mirrors are available: • `-g` – Enables the geometric read option, which results in faster performance on sequential reads. • `-r` – Directs all reads to the first submirror. Use the `-r` option only when the devices that comprise the first submirror are substantially faster than those of the second mirror. You cannot use the `-r` option with the `-g` option.
write_options	The following write option is available: `S` – Performs serial writes to mirrors. The default setting for this option is parallel write.
pass_num	A number (0–9) at the end of an entry defining a mirror that determines the order in which that mirror is resynchronized during a reboot. The default is 1. Smaller pass numbers are resynchronized first. Equal pass numbers are run concurrently. If 0 is used, the resynchronization is skipped. Use 0 only for mirrors mounted as read-only, or as swap space.

Note – If neither the `-g` nor `-r` options are specified, reads are made in a round-robin order from all submirrors in the mirror. This process enables load balancing across the submirrors.

The following command-line example creates a mirrored volume named `d10`, and attaches a one-way mirror using volume `d11`. Volume `d11` is a submirror of the mirror named `d10`.

```
# /usr/sbin/metainit d10 -m d11
d10: Mirror is setup
```

The Enhanced Storage Tool

You can also create the mirror by using the Enhanced Storage Tool within the Solaris Volume Manager software.

To create a mirror:

1. Click the Volumes icon.

   The previously configured RAID-0 volumes are displayed, as shown in Figure 9-23. If these volumes are not displayed, you must first configure the RAID-0 volumes before you can use them as submirrors of the RAID-1 volume.

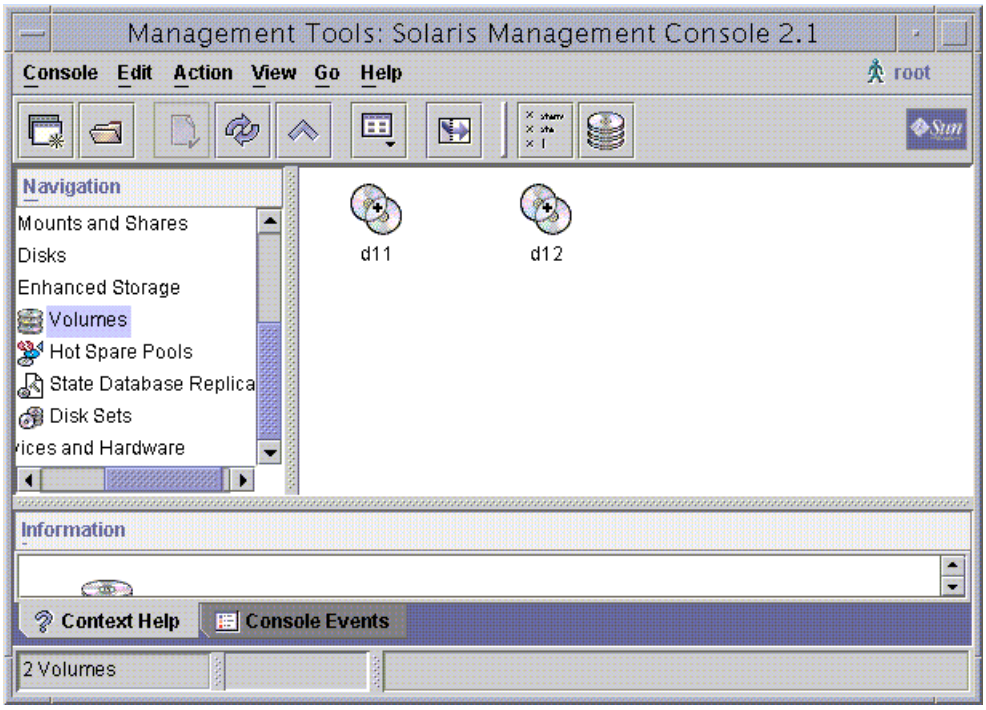


Figure 9-23 Solaris Management Console: Volume

2. Select Create Volume from the Action menu, as shown in Figure 9-24.

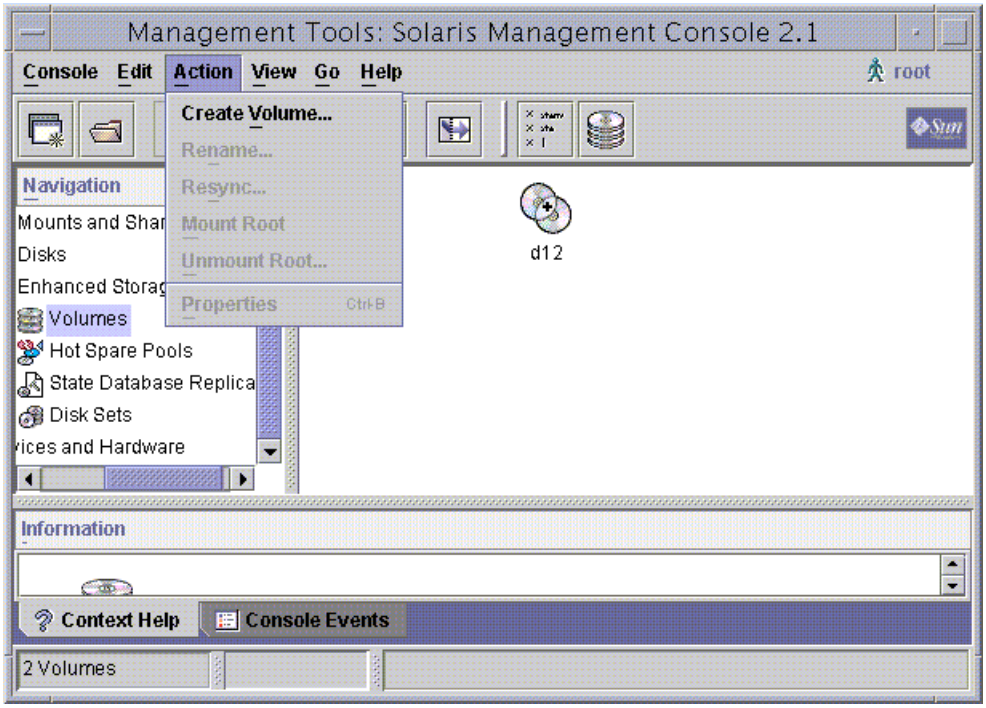


Figure 9-24 Solaris Management Console: Action Menu Window

Because the dirty region logs that are used to track which data blocks in the sub-mirrors have been modified and are recorded within the state database replicas, when you create RAID-1 volumes, you can add additional state database replicas. You do not have to create additional replicas when creating RAID-1 volumes, but mirror performance might suffer if you do not.

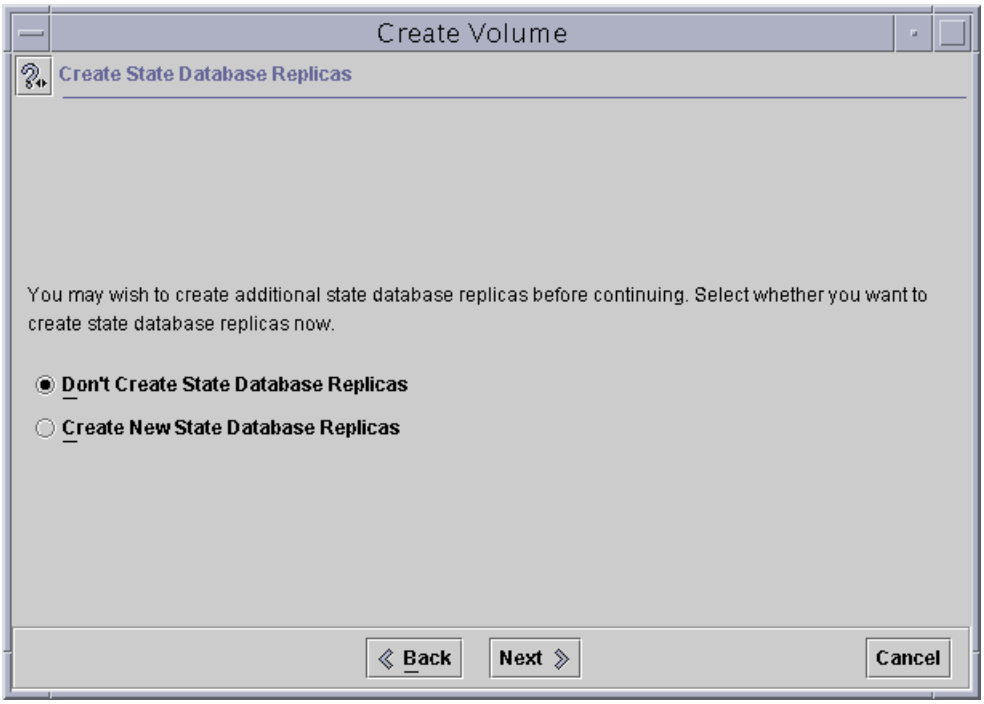


Figure 9-25 Create Volume: Create State Database Replicas Window

3. Due to equipment limitations in the classroom, select Don't Create State Database Replicas, as shown in Figure 9-25.
4. Click Next to continue.

You can relocate the mirror to alternate disk sets.

5. If only one disk set exists on the system, select the default of <none>, as shown in Figure 9-26.

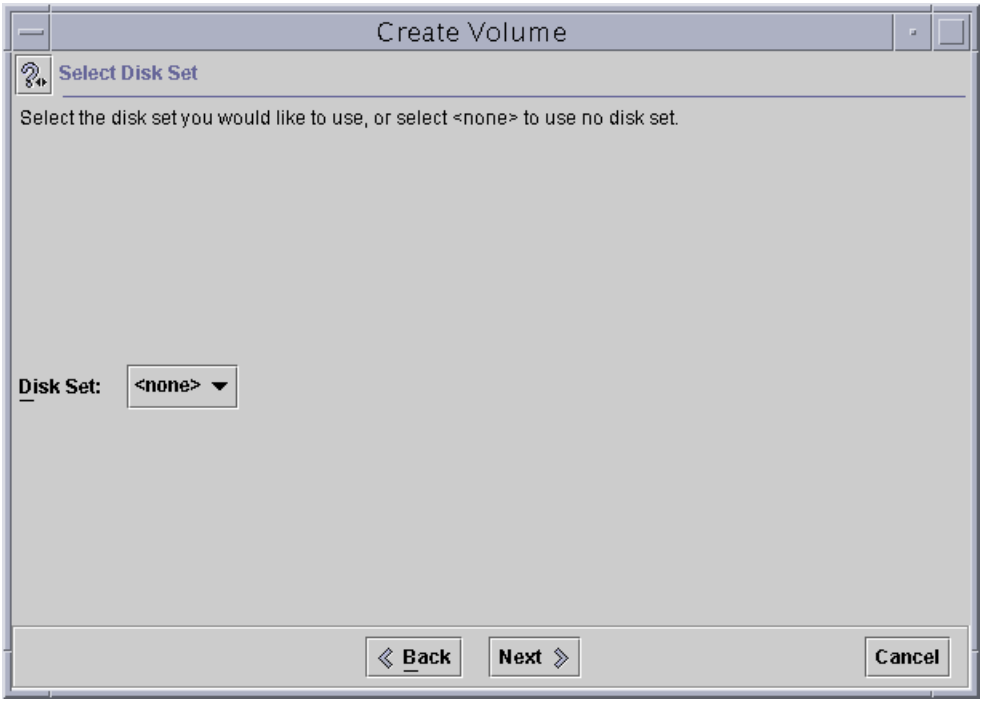


Figure 9-26 Create Volume: Select Disk Set Window

6. Click Next to continue.

Note – When you are mirroring root, you must use the local disk set.

The Create Volume: Select Volume Type Window window displays which volume configurations you can create, as shown in Figure 9-27.

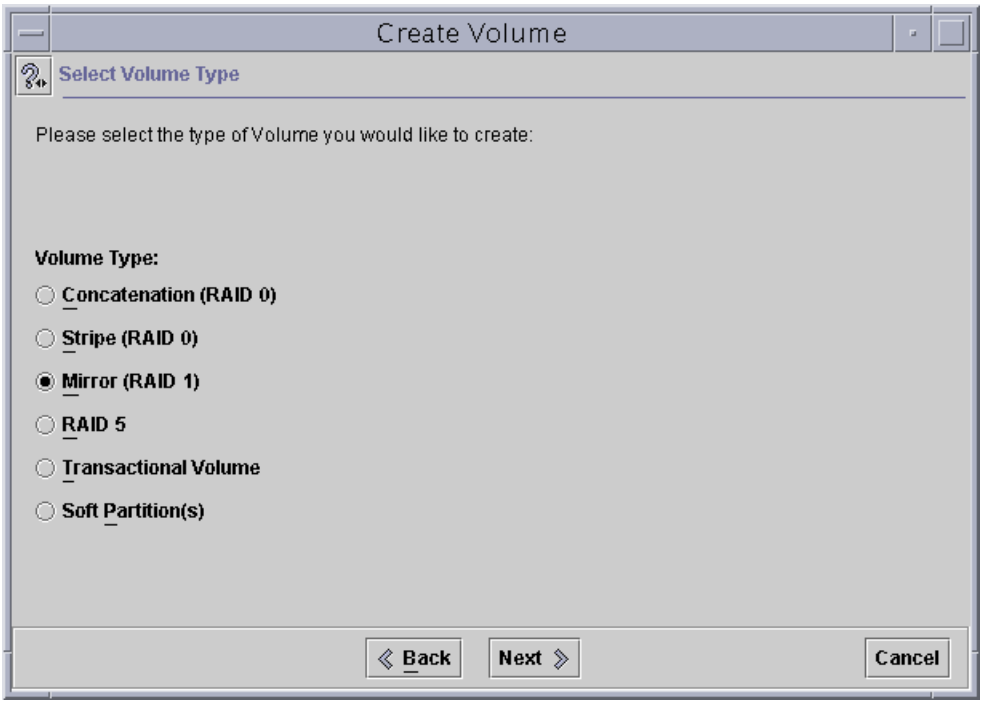


Figure 9-27 Create Volume: Select Volume Type Window

7. Choose Mirror (RAID 1).
8. Click Next to continue.

In the Create Volume: Name Volume Window window, you can enter a volume name, as shown in Figure 9-28. Choose a pattern that is easy to remember so that it is easy to identify the volume types. For example, you could name the RAID-1 volumes with names ending in zero, such as `d10`. Then you can number the submirrors or RAID-0 volumes as `d11` for the first submirror and `d12` for the second submirror.

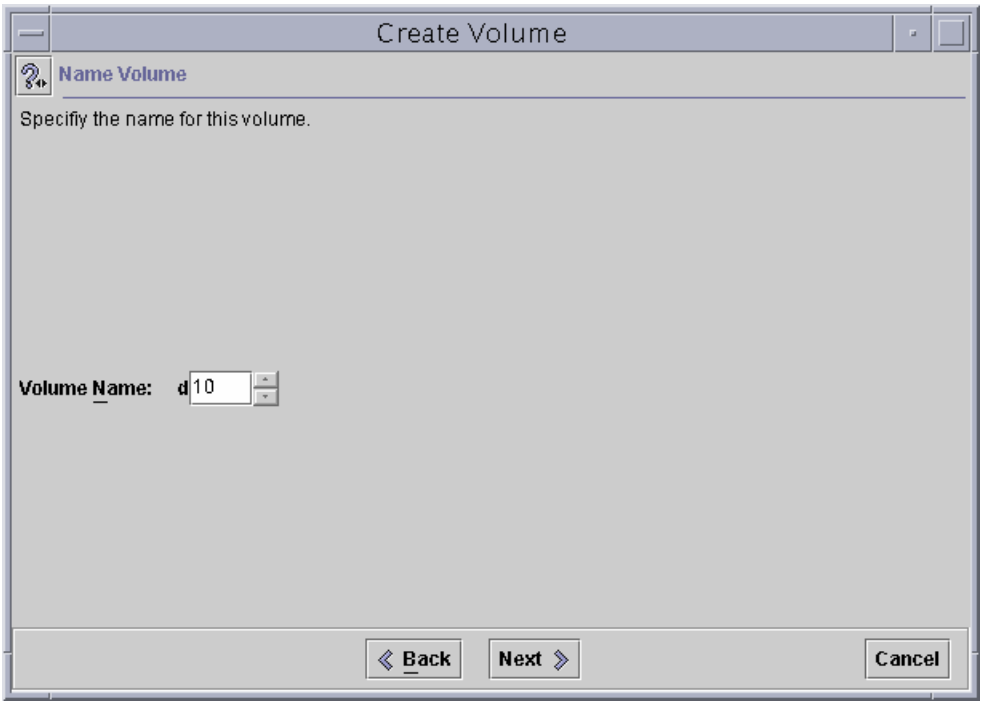


Figure 9-28 Create Volume: Name Volume Window

9. Enter `10` as the volume name d field.
10. Click Next to continue.

11. Select metadevice `d11` for use as the primary submirror, as shown in Figure 9-29.

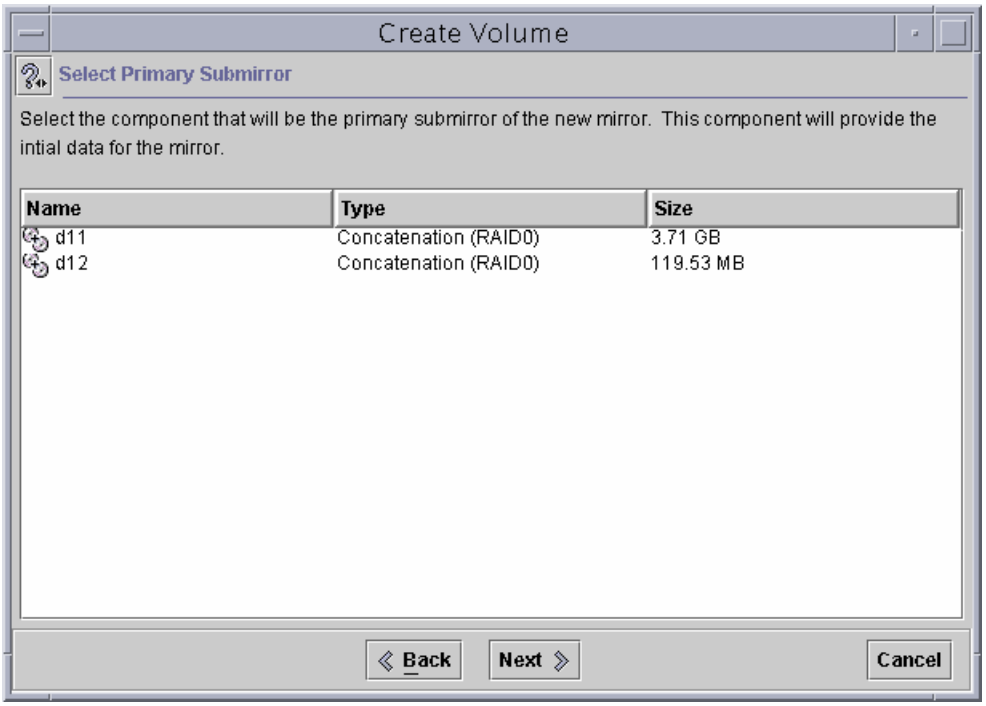


Figure 9-29 Create Volume: Select Primary Submirror Window

12. Click Next to continue.

13. Bypass the Create Volume: Select Remaining Submirrors Window window shown in Figure 9-30, because you are mirroring the root partition, which means that you must attach the secondary submirror by using the command line.

    - When mirroring the root (/) partition, the procedure requires a few additional steps prior to attaching the secondary submirror.
    - When building a mirror that does not already contain data, you can select the secondary submirror, as shown in Figure 9-30.

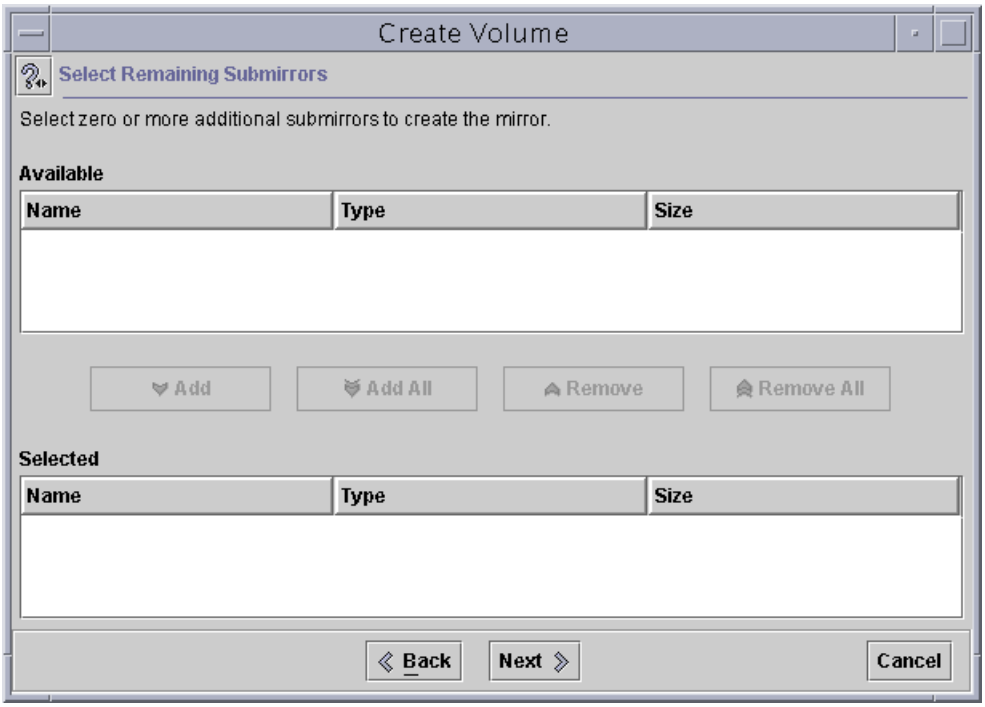


Figure 9-30 Create Volume: Select Remaining Submirrors Window

14. Click Next to continue.

The Create Volume: Set Mirror Parameters Window window lets you set the mirror parameters, as shown in Figure 9-31. These parameters were described in the `metainit` command example that was used to configure a RAID-1 volume.

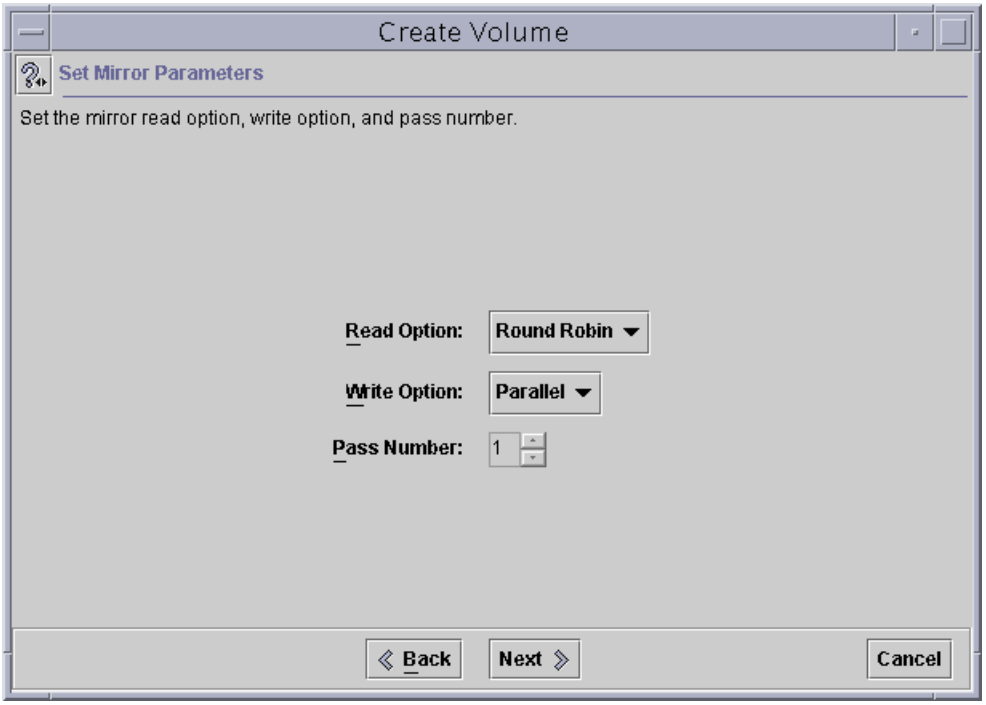


Figure 9-31 Create Volume: Set Mirror Parameters Window

15. To accept the defaults, click Next to continue.

Review your selections in The Create Volume: Review Window window, as shown in Figure 9-32. This window provides a confirmation of your selections. It also provides a summary of the commands necessary to accomplish the identical task from the command line.

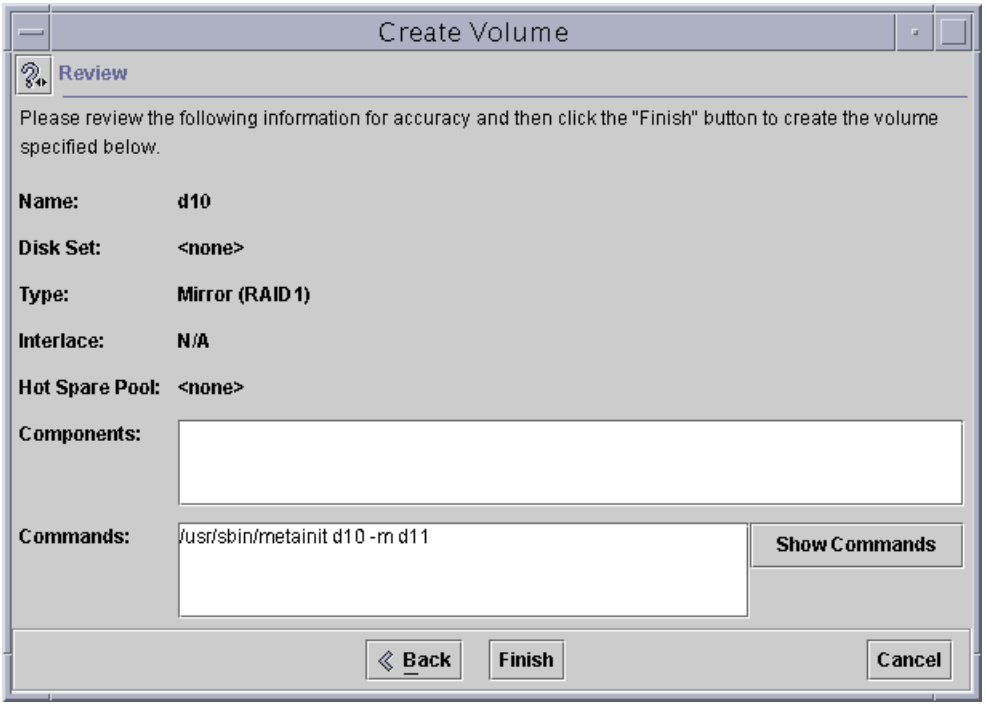


Figure 9-32 Create Volume: Review Window

16. Click Finish.

The RAID-1 volume named `d10` is created, and the display is updated, as shown in Figure 9-33. The primary submirror (`d11`) is attached to the mirror (`d10`), but the process of creating the mirrored partition is not complete.

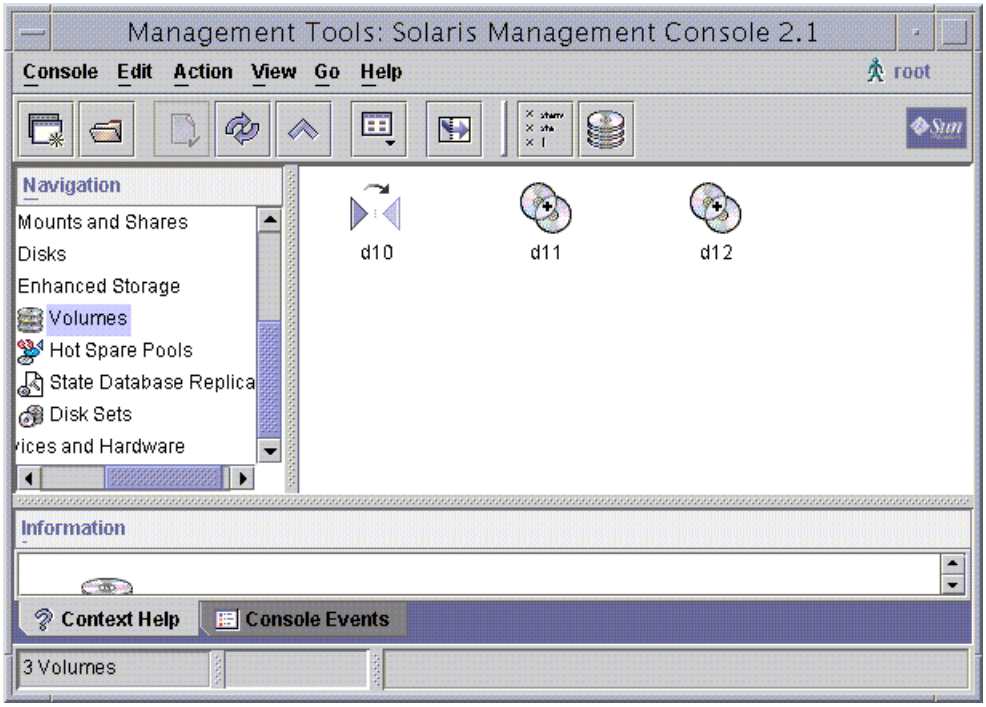


Figure 9-33 Solaris Management Console: Volumes

17. Go to the command line, and use the `metaroot` command to complete building the mirror of the root (/) file system, as described in the "Executing the metaroot Command" section on page 9-40.

Executing the metaroot Command

When creating mirrors of mounted file systems, you must update the /etc/vfstab file to change the mount point from a slice, such as /dev/dsk/c#t#d#s#, to a volume, such as /dev/md/dsk/d##. When mirroring any mounted file system other than root (/), you can use the vi editor to update the /etc/vfstab file.

When mirroring the root (/) file system, use the metaroot command to modify the /etc/vfstab and /etc/system files, as follows:

```
metaroot device
```

where *device* specifies either the metadevice or the conventional disk device (slice) used for the root (/) file system.

The following example shows that the /etc/vfstab file has been updated by the metaroot command to point to the RAID-1 mirrored metadevice.

```
# metaroot d10
# grep md /etc/vfstab
/dev/md/dsk/d10/dev/md/rdsk/d10/ufs1no-
```

In addition to modifying the /etc/vfstab file to update the root (/) file system pointer, the metaroot command updates the /etc/system file to contain the forceload statement that loads the kernel modules that support the logical volumes. For example:

```
# tail /etc/system
forceload: misc/md_hotspares
forceload: misc/md_sp
forceload: misc/md_stripe
forceload: misc/md_mirror
forceload: drv/pcipsy
forceload: drv/simba
forceload: drv/glm
forceload: drv/sd
rootdev:/pseudo/md@0:0,10,blk
```

You must reboot the system before attaching the secondary submirror. Enter the `init` command to reboot the system:

```
# init 6
```

After the reboot is complete, attach the secondary submirror by using the `metattach` command:

```
# metattach d10 d12
d10: submirror d12 is attached
```

Caution – Create a one-way mirror with the `metainit` command, and then attach the additional submirrors with the `metattach` command. If the `metattach` command is not used, no resynchronization operations occur. As a result, data could become corrupted as the Solaris Volume Manager software assumes that both sides of the mirror are identical and can be used interchangeably.

Updating the boot-device PROM Variable

If you mirror your root (/) file system, record the alternate boot path contained in the boot-device PROM variable. In the following example, you determine the path to the alternate boot device by using the `ls -l` command on the slice that is being attached as the secondary submirror to the root (/) mirror.

```
# ls -l /dev/dsk/c1t2d0s1
lrwxrwxrwx   1 root     root          46 Feb 28 08:58 /dev/dsk/c1t2d0s1
-> ../../devices/pci@1f,0/pci@1/scsi@4,1/sd@2,0:b
```

Record the path that follows the `/devices` directory:

```
/pci@1f,0/pci@1/scsi@4,1/sd@2,0:b
```

Caution – When using some disk controllers, the path to the device varies between the entries in the `/devices` directory and the entries in the OpenBoot™ programmable read-only memory (PROM). In these instances, follow the entries in the OpenBoot PROM.

If, for example, on one Ultra™ 5 workstation, the PCI-SCSI controller returns:

```
/pci@1f,0/pci@1/scsi@4,1/sd@2,0:b
```

from the `/devices` directory, yet the `show-devs` command from the OpenBoot PROM returned:

```
/pci@1f,0/pci@1/scsi@4,1/disk
```

then, the alternate boot path must be:

```
/pci@1f,0/pci@1/scsi@4,1/disk@2,0:b
```

If you do not adapt to the change when attempting to boot from the alternate boot device, you get an error stating:

```
can't open boot device
```

To get the system to boot automatically from the alternate boot device in the event of a primary root submirror failure, complete the following steps:

1. Use the OpenBoot `nvalias` command to define a `backup_root` device alias for the secondary root mirror. For example:

```
ok nvalias backup_root /pci@1f,0/pci@1/scsi@4,1/disk@2,0:b
```

2. Redefine the `boot-device` variable to reference both the primary and secondary submirrors, in the order in which you want to access them. For example:

```
ok printenv boot-device
boot-device= disk net
ok setenv boot-device disk backup_root net
boot-device= disk backup_root net
```

In the event of primary root disk failure, the system automatically boots from the secondary submirror. To test the secondary submirror, boot the system manually, as follows:

```
ok boot backup_root
```

Unmirroring the root (/) File System

Follow this procedure to unmirror the root (/) file system. This procedure assumes that the root (/) file system is mirrored on a Solaris Volume Manager software volume named `d10`, and that the mirror consists of two submirrors. The primary submirror is `d11`, and the secondary submirror is `d12`. To unmirror the root (/) file system, complete the following steps:

1. Run the `metastat` command on the mirror to verify that submirror 0 is in the `Okay` state.

```
# metastat d10
d10: Mirror
    Submirror 0: d11
      State: Okay
    Submirror 1: d12
      State: Okay
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 243810 blocks

d11: Submirror of d10
    State: Okay
    Size: 243810 blocks
    Stripe 0:
        Device     Start Block  Dbase        State Reloc Hot Spare
        c0t0d0s0          0     No            Okay   Yes


d12: Submirror of d10
    State: Okay
    Size: 244800 blocks
    Stripe 0:
        Device     Start Block  Dbase        State Reloc Hot Spare
        c1t2d0s1          0     No            Okay   Yes


Device Relocation Information:
Device   Reloc  Device ID
c0t0d0   Yes    id1,dad@AST34342A=_____________GG954138
c1t2d0   Yes    id1,sd@SSEAGATE_ST41600N_SUN1.3G141734__
```

2. Run the `metadetach` command on the mirror to make a one-way mirror.

```
# metadetach d10 d12
d10: submirror d12 is detached
```

3. Because this is a root (/) file system mirror, run the `metaroot` command to update the `/etc/vfstab` and `etc/system` files.

```
# metaroot /dev/dsk/c0t0d0s0
# grep c0t0d0s0 /etc/vfstab
/dev/dsk/c0t0d0s0/dev/rdsk/c0t0d0s0/ufs1no-
```

4. Reboot the system.

```
# init 6
```

5. Run the `metaclear` command to clear the mirror and submirrors. The `-r` option recursively deletes specified metadevices and hot spare pools, associated with the targeted metadevices specified in the `metaclear` command.

```
# metaclear -r d10
d10: Mirror is cleared
d11: Concat/Stripe is cleared
# metaclear d12
d12: Concat/Stripe is cleared
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Mirroring the root (/) File System (Level 1)

In this lab, you:

- Configure the Solaris Volume Manager software to create state database replicas
- Mirror the root (/) file system
- Update the default boot device
- Unmirror the root (/) file system

Preparation

This exercise mirrors the root (/) file system of the system disk.

This exercise mirrors the root (/) file system of the system disk. Use the auto-layout feature for the system disk when installing the Solaris 9 OE. This creates a root (/) partition approximately 120 Mbytes large.

As a setup requirement, the second disk on your system must be partitioned with one slice that is equal to or larger than the root (/) partition of the system disk. You must also partition space for the state database replicas on the second disk. You can define how the remaining slices of the second disk must be partitioned.

This exercise is performed on each individual system, so there is no need to partner students with each other for this exercise. Most steps in these procedures are executable by using either the Enhanced Storage Tool within the Solaris Volume Manager software or by using the command line.

For this exercise, the solutions to each step is presented using the command-line equivalent. The Enhanced Storage Tool within the Solaris Volume Manager software is open and used to display a visual record of the Solaris Volume Manager software's activities.

Tasks

Perform the following tasks:

- Map the available disk slices to the requirements for state database replicas and root (/) file system submirrors.
- Create the state database.
- Build the mirror of the root (/) file system.
- Modify the OpenBoot PROM variables to use the mirrored device as an alternate boot path in the event of a failure of the primary submirror.
- Reboot the system using the secondary root (/) submirror to test the mirror.
- Reboot the system using the primary root (/) submirror.
- Remove the mirror from the root (/) partition.

Exercise: Mirroring the root (/) File System (Level 2)

In this lab, you:

- Configure the Solaris Volume Manager software to create state database replicas
- Mirror the root (/) file system
- Update the default boot device
- Unmirror the root (/) file system

Preparation

This exercise mirrors the root (/) file system of the system disk. Use the auto-layout feature for the system disk when installing the Solaris 9 OE. This creates a root (/) partition approximately 120 Mbytes large.

As a setup requirement, the second disk on your system must be partitioned with one slice that is equal to or larger than the root (/) partition of the system disk. You must also partition space for the state database replicas on the second disk. You can define how the remaining slices of the second disk must be partitioned.

This exercise is performed on each individual system, so there is no need to partner students with each other for this exercise. Most steps in these procedures are executable by using either the Enhanced Storage Tool within the Solaris Volume Manager Software or by using the command line.

For this exercise, the solutions to each step is presented using the command-line equivalent. The Enhanced Storage Tool within the Solaris Volume Manager is open and used to display a visual record of the Solaris Volume Manager software's activities.

Task Summary

Perform the following tasks:

- Map the available disk slices to the requirements for state database replicas and root (/) file system submirrors.
- Create the state database.
- Build the mirror of the root (/) file system.
- Modify the OpenBoot PROM variables to use the mirrored device as an alternate boot path in the event of a failure of the primary submirror.
- Reboot the system using the secondary root (/) submirror to test the mirror.
- Reboot the system using the primary root (/) submirror.
- Remove the mirror from the root partition.

Tasks

Complete the following steps:

1. Open the Enhanced Storage Tool within the Solaris Management Console, and leave it open throughout this exercise to use it as a monitoring mechanism.
2. Fill in the blanks to record the information needed to complete this exercise:
   - Disk slice for the state database replica 1:
     _____
   - Disk slice for the state database replica 2:
     _____
   - Disk slice for the state database replica 3:
     _____
   - Disk slice for the state database replica 4 (optional):
     _____
   - Disk slice for the state database replica 5 (optional):
     _____
   - Disk slice for the root file system primary submirror:
     _____

- Metadevice to map to the root (/) file system primary submirror:

  _____

- Disk slice for the root (/) file system secondary submirror:

  _____

- Metadevice to map to the root (/) file system secondary submirror:

  _____

- Metadevice to map to the root (/) file system mirror:

  _____

3. Create a sufficient number of state database replicas to support the majority consensus algorithm used in the Solaris Volume Manager software.

   What is the minimum number of state database replicas necessary to support the majority consensus algorithm?

   _____

4. Create a RAID-0 volume to use as the root (/) file system's primary submirror.
5. Create a RAID-0 volume on the secondary drive to use as the root (/) file system's secondary submirror.
6. Create a RAID-1 volume as a one-way mirror using the root (/) file system primary submirror as the source of the mirror's data.
7. Update the `/etc/vfstab` file to use the RAID-1 volume as the mount point for the root (/) file system.
8. Reboot the system.
9. Attach the RAID-0 volume used as the root (/) file system's secondary submirror to the RAID-1 volume, and allow the mirror synchronization to complete before continuing.

   What is the primary reason for using the command line to attach a secondary submirror to a mirror?

   _____

Note – To view the status of the resynchronization process, perform the `/usr/sbin/metastat | grep resync` command

10. Determine the path to the alternate root (/) device (as reported by the Solaris 9 OE).

    _____

11. Determine the path to the alternate root (/) device (as reported by the OpenBoot PROM).

    _____

12. Define a backup root (/) device alias.

    _____

13. Add the backup root (/) device alias to the `boot-device` variable.

    _____

14. Test the ability to boot the secondary root (/) submirror.

    _____

15. Verify the status of the root (/) submirrors.

    _____

16. Detach one submirror to make the root (/) mirror a one-way mirror.

    _____

17. Update the `/etc/vfstab` file to redefine the root (/) mount point using the original disk slice and the `/etc/system` file to include the `forceload` statements.
18. Reboot the system.
19. Clear the mirror and submirrors.

Exercise: Mirroring the root (/) File System (Level 3)

In this lab, you:

- Configure the Solaris Volume Manager software to create state database replicas
- Mirror the root (/) file system
- Update the default boot device
- Unmirror the root (/) file system

Preparation

This exercise mirrors the root (/) file system of the system disk. Use the auto-layout feature for the system disk when installing the Solaris 9 OE. This creates a root (/) partition approximately 120 Mbytes large.

As a setup requirement, the second disk on your system must be partitioned with one slice that is equal to or larger than the root (/) partition of the system disk. You must also partition space for the state database replicas on the second disk. You can define how the remaining slices of the second disk must be partitioned.

This exercise is performed on each individual system, so there is no need to partner students with each other for this exercise. Most steps in these procedures are executable by using either the Enhanced Storage Tool within the Solaris Volume Manager or by using the command line.

For this exercise, the solutions to each step is presented using the command-line equivalent. The Enhanced Storage Tool within the Solaris Volume Manager is open and used to display a visual record of the Solaris Volume Manager software's activities.

Task Summary

Perform the following tasks:

- Map the available disk slices to the requirements for state database replicas and root (/) file system submirrors.
- Create the state database.
- Build the mirror of the root (/) file system.
- Modify the OpenBoot PROM variables to use the mirrored device as an alternate boot path in the event of a failure of the primary submirror.
- Reboot the system using the secondary root (/) submirror to test the mirror.
- Reboot the system using the primary root (/) submirror.
- Remove the mirror from the root (/) partition.

Tasks and Solutions

This sections provides the tasks and their solutions.

1. Open the Enhanced Storage Tool within the Solaris Management Console, and leave it open throughout this exercise to use it as a monitoring mechanism.

```
# smc &
```

Note – The task solutions are presented using the command-line equivalents because every task step can be performed by using the command line.

2. Fill in the blanks to record the information needed to complete this exercise:
   - Disk slice for the state database replica 1:

     As defined for your lab system.
   - Disk slice for the state database replica 2:

     As defined for your lab system.
   - Disk slice for the state database replica 3:

     As defined for your lab system.

- Disk slice for the state database replica 4 (optional):

  As defined for your lab system.

- Disk slice for the state database replica 5 (optional):

  As defined for your lab system.

- Disk slice for the root (/) file system primary submirror:

  As defined for your lab system.

- Volume to map to the root (/) file system primary submirror:

  As defined for your lab system.

- Disk slice for the root (/) file system secondary submirror:

  As defined for your lab system.

- Metdevice to map to the root (/) file system secondary submirror:

  As defined for your lab system.

- Metadevice to map to the root (/) file system mirror:

  As defined for your lab system.

3. Create a sufficient number of state database replicas to support the majority consensus algorithm used in the Solaris Volume Manager software.

```
# /usr/sbin/metadb -a -f c#t#d#s0
# /usr/sbin/metadb -a c#t#d#s1
# /usr/sbin/metadb -a c#t#d#s3
```

What is the minimum number of state database replicas necessary to support the majority consensus algorithm?

Three state database replicas are recommended as the minimum to support the majority consensus algorithm.

4. Create a RAID-0 volume to use as the root (/) file system's primary submirror.

```
# /usr/sbin/metainit -f d11 1 1 c#t#d#s#
```

(The variable points to the root (/) slice.)

```
d11: Concat/Stripe is setup
```

5. Create a RAID 0 volume on the secondary drive to use as the root (/) file system's secondary submirror.

```
# /usr/sbin/metainit d12 1 1 c#t#d#s#
d12: Concat/Stripe is setup
```

6. Create a RAID-1 volume as a one-way mirror using the root (/) file system primary submirror as the source of the mirror's data.

```
# /usr/sbin/metainit d10 -m d11
d10: Mirror is setup
```

7. Update the /etc/vfstab file to use the RAID-1 volume as the mount point for the root (/) file system.

```
# /usr/sbin/metaroot d10
```

8. Reboot the system.

```
# init 6
```

9. Attach the RAID-0 volume used as the root (/) file system's secondary submirror to the RAID-1 volume, and allow the mirror synchronization to complete before continuing.

```
# /usr/sbin/metattach d10 d12
```

What is the primary reason for using the command line to attach a secondary submirror to a mirror?

The primary reason for using the command line to attach a secondary submirror to a mirror is to force a resynchronization of the data between the primary and secondary submirror.

Note – To view the status of the resynchronization process, perform the /usr/sbin/metastat | grep resync command

10. Determine the path to the alternate root (/) device (as reported by the Solaris OE).

Varies by system. Use the ls -l *command.*

```
# ls -l /dev/dsk/c#t#d#s4
```

11. Determine the path to the alternate root (/) device (as reported by the OpenBoot PROM).

Varies by system. Use the show-devs *command.*

```
ok show-devs
```

12. Define a backup root (/) device alias.

Varies by system. Use the `nvalias` *command.*

```
ok nvalias backup_root device_path
```

13. Add the backup root (/) device alias to the `boot-device` variable.

Varies by system. Use a combination of the `printenv` *and* `setenv` *commands.*

```
ok printenv boot-device
boot-device = disk net
ok setenv boot-device disk backup_root net
boot-device = disk backup_root net
```

14. Test the ability to boot the secondary root (/) submirror.

```
ok boot backup_root
```

15. Verify the status of the root (/) submirrors.

```
# /usr/sbin/metastat d10
```

16. Detach one submirror to make the root (/) mirror a one-way mirror.

```
# /usr/sbin/metadetach d10 d12
```

17. Update the `/etc/vfstab` file to redefine the root (/) mount point using the original disk slice and the `/etc/system` file to include the `forceload` statements.

```
# /usr/sbin/metaroot /dev/dsk/c#t#d#s#
```

18. Reboot the system.

```
# init 6
```

19. Clear the mirror and submirrors.

```
# /usr/sbin/metaclear -r d10
# /usr/sbin/metaclear d12
```

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Configuring Access Control Lists (ACLs)

Objectives

This module teaches you how to create and configure unique access permissions on files and directories using access control lists (ACLs). Upon completion of this module, you should be able to:

- Describe ACLs
- Manipulate ACLs using the command line
- Manipulate ACLs using the File Manager graphic user interface (GUI)
- Create default ACLs

The following course map shows how this module fits into the current instructional goal.

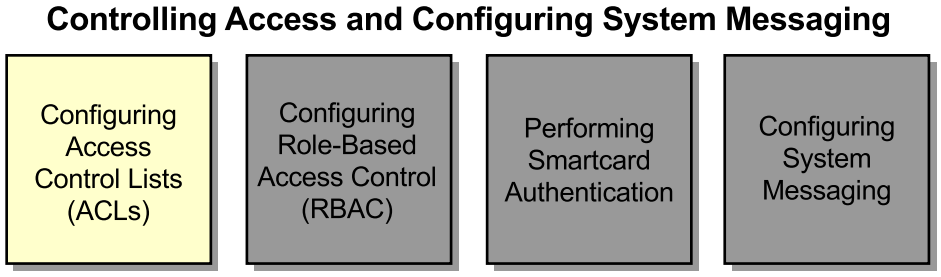


Figure 10-1 Course Map

Introducing ACLs

When an ACL is created for a file or directory, the ACL provides an extended and customized set of permissions for the file or directory. These permissions are used in addition to the conventional UNIX® permissions associated with each file or directory.

Standard UNIX file protection provides read, write, and execute permissions for the three user classes: file owner, file group, and other. ACLs provide greater data access control for each file or directory. ACLs enable you to define permissions for specific users and groups. Default ACL permissions also exist, and they can be set on files and directories.

Defining ACL Entries

Each ACL entry has the following syntax:

entry-type:[*UID or GID*]:*perm*

where:

entry-type	Specifies the scope of the file permissions to the owner, owner's group, specific users, additional groups, or the ACL mask.
UID or *GID*	Specifies the user's name or user's identification number (UID), or the group's name or group's identification number (GID).
perm	Symbolically specifies permissions for *entry-type* by using `r`, `w`, `x`, and `-`, or by using octal values from `0` to `7`.

Note – ACL entries are labeled as *acl_entry* in all the command-line examples.

ACL Entry Types

Table 10-1 shows the syntax than an ACL entries can have.

Table 10-1 ACL Entry Types

Entry Type	Description
u[ser]::*perm*	The permissions for the file owner.
g[roup]::*perm*	The permissions for the owner's group.
o[ther]:*perm*	The permissions for users other than the owner or members of the owner's group.
u[ser]:*UID*:*perm* or u[ser]:*username*:*perm*	The permissions for a specific user. The *username* must exist in the /etc/passwd file.
g[roup]:*GID*:*perm* or g[roup]:*groupname*:*perm*	The permissions for a specific group. The *groupname* must exist in the /etc/group file.
m[ask]:*perm*	The ACL mask, which indicates the maximum effective permissions allowed for all specific users and groups. The mask does not set the permissions for the owner or others. You can use the mask as a quick way to change effective permissions for all the specific users and groups.

ACL Permissions

The permissions field in each entry represents the permissions allowed. You can express the ACL permissions variable using either the symbolic characters `rwx` or an octal number, just as you would for conventional UNIX permissions. Table 10-2 lists the possible permissions and their descriptions.

Table 10-2 ACL Permissions and Descriptions

Symbolic			**Binary Equivalent**	**Octal Permission**	**Definition**
r	**w**	**x**			
r	w	x	1 1 1	7	Read, write, and execute allowed
r	w	–	1 1 0	6	Read and write allowed, execute denied
r	–	x	1 0 1	5	Read and execute allowed, write denied
r	–	–	1 0 0	4	Read allowed, write and execute denied
–	–	–	0 0 0	0	Read, write, and execute denied

Comparing ACL Permissions to Standard UNIX Permissions

Although both ACLs and standard UNIX permission bits affect access rights for files and directories, ACL permissions are not a replacement for standard permissions. The `umask` value sets permissions on the file or directory at the time of initial creation. The associated inode records these permissions. After the file or directory is created and the initial permissions are recorded, the `umask` value is no longer referenced for that file or directory.

When you create an ACL, the existing inode points to a newly allocated inode called a shadow inode. When a specific ACL entry is placed on the ACL list, the shadow inode contains a pointer to a data block containing the list of ACL entries, as shown in Figure 10-2.

Note:
Permissions are determined by
the `umask` value at creation time.

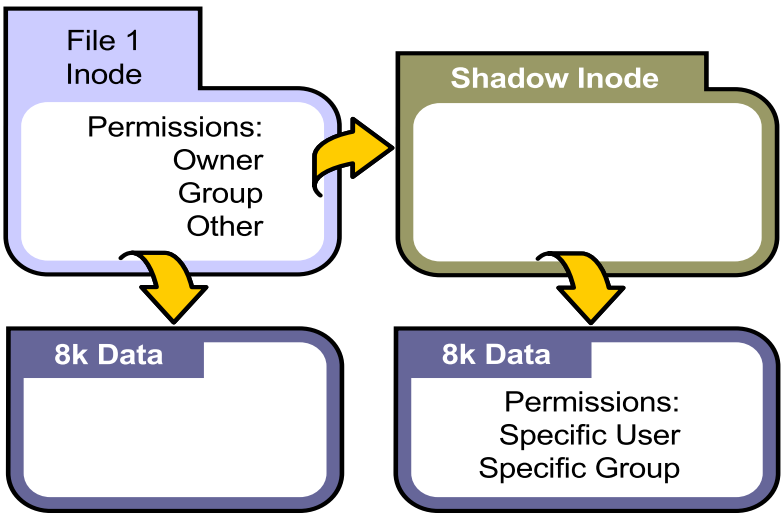


Figure 10-2 Shadow Inode of a File With an ACL

After the `umask` value has been applied, the inode records the standard permissions, while the ACL data block records the permissions of the ACL entries. You can modify the standard permissions without affecting the permissions of the ACL entries. You can also modify the permissions of the ACL entries without affecting the standard permissions.

Introducing ACL Commands

Table 10-3 shows you which command and options to enter when you want to set or view ACLs for a file or directory.

Table 10-3 ACL Command Options and Descriptions

Command/Option	**Description**
`getfacl` *`filename(s)`*	Displays ACL entries for files
`setfacl -m` *`acl_entries filename`*	Creates or modifies ACL entries on files
`setfacl -s` *`acl_entries filename`*	Substitutes new ACL entries for old ACL entries
`setfacl -d` *`acl_entries filename`*	Deletes one or more ACL entries on files
`setfacl -f` *`acl_file filename`*	Specifies an ACL configuration file that contains a list of permissions to set on other files
`setfacl -r` *`filename`*	Recalculates the ACL mask based on the ACL entries

Manipulating ACLs Using the Command Line

You can set ACLs using the command line or the File Manager GUI. You can launch the File Manager GUI using the `/usr/dt/bin/dtfile` command. These tools allow you to:

- Determine if a file has an ACL
- Display an ACL
- Modify an ACL
- Delete an ACL
- Substitute an ACL
- Recalculate an ACL mask
- Copy an ACL list from a file

Determining if a File Has an ACL

You can use the `ls -l` command to see which files or directories have an ACL entry. The `ls` command does not display the actual list of ACL entries. To display the list of ACL entries, use the `getfacl` command.

When viewing the output of the `ls -l` command, if a file has an ACL entry, a plus (+) sign appears at the end of the permission field.

```
$ pwd
/export/home/userc
$ ls -l
total 0
-rw-r--r--   1 userc    staff          0 Jan 22 13:40 file1
-rw-r--r--+  1 userc    staff          0 Jan 22 13:40 file2
```

In this example, the lack of a + sign for the file named `file1` shows that it does not contain an ACL entry. Therefore, `file1` is considered to have a trivial ACL. The presence of a + sign for the file named `file2` indicates that this file has an ACL entry. Therefore, `file2` is considered to have a non-trivial ACL. The output of the `getfacl` command further shows the concept of trivial ACLs.

Displaying ACLs

Use the `getfacl` command to display the contents of ACL entries for a file or directory. The syntax of the command is:

```
getfacl [-a | -d] filename1 [filename2 ...]
```

where:

`-a`	Displays the file name, file owner, file group, and ACL entries for the specified file or directory
`-d`	Displays the file name, file owner, file group, and default ACL entries for the specified directory
`filename#`	Specifies one or more files or directories

The `pwd` and `ls -l` commands show the current working directory and its contents.

```
$ pwd
/export/home/userc
$ ls -l
total 0
-rw-r--r--   1 userc    staff          0 Jan 22 13:40 file1
-rw-r--r--+  1 userc    staff          0 Jan 22 13:40 file2
```

To list the ACL entries for the contents of the current directory, enter the `getfacl` command. If you specify multiple file names on the command line, the ACL entries in the output are separated by a blank line.

Custom ACL entries define the permissions for the user or group named in the ACL entry. Each file or directory also contains an ACL mask value. The ACL mask value globally limits the *effective* permissions for every custom ACL entry on a particular file or directory. There are no effective permissions listed for a file's owner or "other" users. However, the file's group and any other specific users or groups present in the ACL list have effective permissions. When no ACL mask is specifically set on a file or directory, the ACL mask has the same permissions as the group permissions for that file or directory.

The ACL permission bits define specific user or specific group permissions that are allowed, subject to the ACL mask. The ACL mask defines the maximum set of effective permissions that are allowed for an ACL entry. An ACL mask setting of `rw-` (or octal number `6`) on a file allows read and write permission on the file but does not allow execute permission on this file.

Note – In the previous context, the ACL mask is not directly related to the shell's `umask` value in any way. The `umask` value globally controls the initial permissions that are set for files or directories for each shell. The ACL mask controls the effective permissions granted for that file or directory. Each file or directory has its own ACL mask.

The following examples show the output of the `getfacl` command:

```
$ getfacl file1

# file: file1
# owner: userc
# group: sysadmin
user::rw-
group::r--              #effective:r--
mask:r--
other:r--
```

If no custom ACL entries are configured, the ACL is trivial. The permissions listed in the output from the `getfacl` command are the same as the current permissions for the file or directory. The ACL mask is listed after the permissions of the group.

```
$ getfacl file2
# file: file2
# owner: userc
# group: sysadmin
user::rw-
user:usera:rwx          #effective:r--
group::r--              #effective:r--
mask:r--
other:r--
```

If a custom ACL entry is configured, the ACL is non-trivial. The file named `file2` has a custom ACL entry for the user named `usera`. The effective permission shows which permissions are allowed when you compute the intersection (a Boolean logical AND operation) of the ACL entry and the ACL mask.

For example, `usera` is given a custom ACL entry that permits read, write, and execute permissions (`rwx`) on `file2`. However, the ACL mask on `file2` allows only read permission (`r--`). Therefore, because of the intersection of `rwx` and `r--`, `usera` has an effective permission of only
`r--`.

Modifying an ACL

The most common method used to configure an ACL is to modify the ACL. To modify ACL entries on a file, use the `setfacl` command. The syntax of the command is:

```
setfacl -m acl_entry, filename
```

where:

`-m`	Modifies the existing ACL entry.
`acl_entry`	Specifies a list of modifications to apply to the ACLs for one or more files, directories, or both. See Table 10-1 on page 10-3 for a description of available ACL entries.
`filename`	Specifies one or more files or directories.

Note – To verify the new ACL entries, use the `getfacl` command.

The following example shows you how to add an ACL entry to a file with existing ACL entries.

```
$ getfacl file2

# file: file2
# owner: userc
# group: sysadmin
user::rw-
user:usera:rwx          #effective:r--
```

```
group::r--              #effective:r--
mask:r--
other:r--
$ setfacl -m u:userb:7 file2
$ getfacl file2

# file: file2
# owner: userc
# group: sysadmin
user::rw-
user:usera:rwx          #effective:r--
user:userb:rwx          #effective:r--
group::r--              #effective:r--
mask:r--
other:r--
```

Even though the ACL entries for `usera` and `userb` request read, write, and execute permissions, the ACL mask does not allow write and execute permissions for both entries.

Deleting an ACL

You use the `setfacl` command to delete an ACL. When deleting an ACL, specify the entry-type and the UID or GID that you want to delete. You cannot delete the ACL entries for the file owner, file group owner, other, and the ACL mask. The syntax of the command is:

`setfacl -d` *`acl_entry filename`*

where:

`-d`	Deletes one or more *`acl_entry`* arguments.
`acl_entry`	Specifies which ACL entry to delete in the file or directory. See Table 10-1 on page 10-3 for a description of available ACL entries.
`filename`	Specifies the file or directory from which to delete the *`acl_entry`* argument.

The outputs of the `pwd` and `ls -l` commands show the current working directory and its contents.

```
$ pwd
/export/home/userc
$ ls -l
total 0
-rw-r--r--   1 userc    staff          0 Jan 22 13:40 file1
-rw-r--r--+  1 userc    staff          0 Jan 22 13:40 file2
```

The output of the `getfacl` command shows the current ACL configuration for `file2`.

```
$ getfacl file2

# file: file2
# owner: userc
# group: sysadmin
user::rw-
user:usera:rwx          #effective:r--
user:userb:rwx          #effective:r--
group::r--              #effective:r--
mask:r--
other:r--
```

The following example shows how to delete an ACL entry from the file named `file2`:

```
$ setfacl -d u:usera file2
```

The output from the `ls -l` command shows that the deleted ACL entry was not the only ACL entry on the file named `file2`. To verify this output, perform the `getfacl` command.

```
$ ls -l
total 0
-rw-r--r--   1 userc    staff          0 Jan 22 13:40 file1
-rw-r--r--+  1 userc    staff          0 Jan 22 13:40 file2
```

The ACL entry for `usera` is deleted, but the ACL entry for `userb` remains.

```
$ getfacl file2

# file: file2
# owner: userc
# group: sysadmin
user::rw-
user:userb:rwx          #effective:r--
group::r--              #effective:r--
mask:r--
other:r--
```

When you remove the last ACL entry on a file, the output of the `ls -l` command reports that the file has a trivial ACL (shown by the lack of a `+` symbol in the output).

```
$ setfacl -d u:userb file2
$ ls -l
total 0
-rw-r--r--   1 userc    staff          0 Jan 22 13:40 file1
-rw-r--r--   1 userc    staff          0 Jan 22 13:40 file2
```

Substituting an ACL

To replace the entire ACL for a file from the command line, you must specify at least the basic set of ACL entries: user, group, other, and ACL mask permissions. Use the `setfacl` command with the following options to substitute an ACL on a file:

```
setfacl -s u::perm,g::perm,o:perm,m:perm,[u:UID:perm],[g:GID:perm] filename
```

where:

`-s`	Specifies a substitution is being made for the entire ACL contents.
`acl_entry`	Specifies which ACL entry (from a list of one or more ACL entries) to modify on the file or directory. See Table 10-1 on page 10-3 for a description of available ACL entries
`filename`	Specifies one or more files or directories.

The following example shows that no ACL entries currently exist on `file1` and `file2`.

```
$ ls -l
total 0
-rw-r--r--   1 userc    sysadmin       0 Apr 18 15:44 file1
-rw-r--r--   1 userc    sysadmin       0 Apr 18 15:44 file2
```

To display the trivial ACL permissions for `file1`, enter the `getfacl` command:

```
$ getfacl file1

# file: file1
# owner: userc
# group: sysadmin
user::rw-
group::r--              #effective:r--
mask:r--
other:r--
```

The following example shows you how to substitute an ACL on the file named `file1`. The ACL permissions are configured as follows:

- The file owner has read, write, and execute permissions
- The group has read and write permissions
- The other users have read-only permissions
- The user named `usera` has read, write, and execute permissions on the file
- The ACL mask has read and write permissions
- The user named `usera` has read, write, and execute permissions

Perform the `setfacl` command with the following options to substitute an ACL on `file1`:

```
$ setfacl -s u::rwx,g::rw-,o:r--,m:rw-,u:usera:rwx file1
```

The `ls -l` command shows that an ACL exists on the file named `file1`.

```
$ ls -l
total 0
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file1
-rw-r--r--   1 userc    sysadmin       0 Apr 18 15:44 file2
$ getfacl file1

# file: file1
# owner: userc
# group: sysadmin
user::rwx
user:usera:rwx          #effective:rw-
group::rw-              #effective:rw-
mask:rw-
other:r--
```

The following example shows how to substitute an ACL on the file named `file2`, using octal notations to establish the ACL entries. Before you replace the entire ACL, use the `getfacl` command to display the ACL for `file2`.

```
$ getfacl file2

# file: file2
# owner: userc
# group: sysadmin
user::rw-
group::r--              #effective:r--
mask:r--
other:r--

$ setfacl -s u::7,g::6,o:4,m:6,u:usera:7 file2
```

After you substitute the ACL permissions using the `setfacl` command, use the `ls -l` command to verify if an ACL exists on the file named `file2`.

```
$ ls -l
total 0
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file1
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file2
```

The output of the `getfacl` command shows the ACLs are identical, regardless of which method is used to create the ACL.

```
$ getfacl file1 file2

# file: file1
# owner: userc
# group: sysadmin
user::rwx
user:usera:rwx          #effective:rw-
group::rw-              #effective:rw-
mask:rw-
other:r--

# file: file2
# owner: userc
# group: sysadmin
user::rwx
user:usera:rwx          #effective:rw-
group::rw-              #effective:rw-
mask:rw-
other:r--
```

Recalculating an ACL Mask

You can globally control the effective permissions of a custom ACL entry by using the ACL mask. However, examples in this module show that the ACL mask sometimes does not allow you to set some of the requested permissions that are indicated in the list of ACL entries.

Therefore, recalculate the ACL mask, and modify it to allow all the requested permissions in the list of ACL entries. After recalculating the ACL mask, the effective permission of each ACL entry allows the full set of requested permissions for the entry.

You can recalculate the ACL mask entry so that it does not limit the effective permissions of any specific user or group. The following example shows how to recalculate the ACL mask entry:

```
setfacl -r -m acl_entry filename...
```

where:

`-r`	Recalculates the ACL mask entry to allow maximum effective permissions for every ACL entry.
`-m`	Modifies the existing ACL entry.
`acl_entry`	Specifies a list of modifications to apply to the ACLs for one or more files, directories, or both. ACL entries can also be added, modified, or deleted in addition to the recalculation of the mask. See Table 10-1 on page 10-3 for a description of available ACL entries.
`filename`	Specifies one or more files or directories.

```
$ getfacl file1

# file: file1
# owner: userc
# group: sysadmin
user::rwx
user:usera:rwx          #effective:rw-
group::rw-              #effective:rw-
mask:rw-
other:r--
```

```
$ setfacl -r -m u:usera:7 file1
$ getfacl file1

# file: file1
# owner: userc
# group: sysadmin
user::rwx
user:usera:rwx          #effective:rwx
group::rw-              #effective:rw-
mask:rwx
other:r--
```

Note – The file owner and other permissions are not considered when recalculating the ACL mask.

Copying an ACL List

To copy an ACL from one file to another, use a combination of the `getfacl` and `setfacl` commands. The syntax of the command is:

```
getfacl filename1 | setfacl -f – filename2
```

where:

filename1	Specifies the file or directory from which to copy the ACL entries.
-f	Sets a file's ACL using ACL entries from the *acl_file* file. If you specify a dash (–) for *acl_file,* the `getfacl` command uses standard input to set the file's ACL.
filename2	Specifies a file or directory to which to copy the ACL entries.

To copy the ACL entry from `file1` to `file3`, you must first create `file3`. Use the `touch` command to create `file3`.

```
$ touch file3
$ ls -l
total 0
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file1
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file2
-rw-r--r--   1 userc    sysadmin       0 Apr 29 14:30 file3
```

To display the ACLs for `file1` and `file3`, perform the `getfacl` command:

```
$ getfacl file1 file3

# file: file1
# owner: userc
# group: sysadmin
user::rwx
user:usera:rwx          #effective:rwx
group::rw-              #effective:rw-
mask:rwx
other:r--

# file: file3
# owner: userc
# group: sysadmin
user::rw-
group::r--              #effective:r--
mask:r--
other:r--
```

To copy the ACL from `file1` to `file3`, use the combination of the `getfacl` and `setfacl` commands:

```
$ getfacl file1 | setfacl -f - file3
$ ls -l
total 0
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file1
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file2
-rwxrw-r--+  1 userc    sysadmin       0 Apr 29 14:30 file3
$ getfacl file1 file3

# file: file1
# owner: userc
# group: sysadmin
user::rwx
user:usera:rwx          #effective:rwx
group::rw-              #effective:rw-
mask:rwx
other:r--

# file: file3
# owner: userc
# group: sysadmin
```

```
user::rwx
user:usera:rwx          #effective:rwx
group::rw-              #effective:rw-
mask:rwx
other:r--
```

Note – Conventional UNIX permissions are set to match the source files when you copy an ACL.

Manipulating ACLs Using the File Manager GUI

The `/usr/dt/bin/dtfile` program, also known as the File Manager, contains mechanisms to perform the following tasks:

- Display ACLs
- Add ACLs
- Change ACLs
- Delete ACLs

Displaying ACLs Using the File Manager GUI

Figure 10-3 shows a partial view of the Common Desktop Environment (CDE) Front Panel. To display ACLs using the File Manager GUI, click the File Manager control in the Front Panel.

Figure 10-3 File Manager GUI Window

Note – You can use the `/usr/dt/bin/dtfile` command to launch the File Manager from the command line.

Figure 10-4 shows a view of the contents of the current folder that you are viewing.

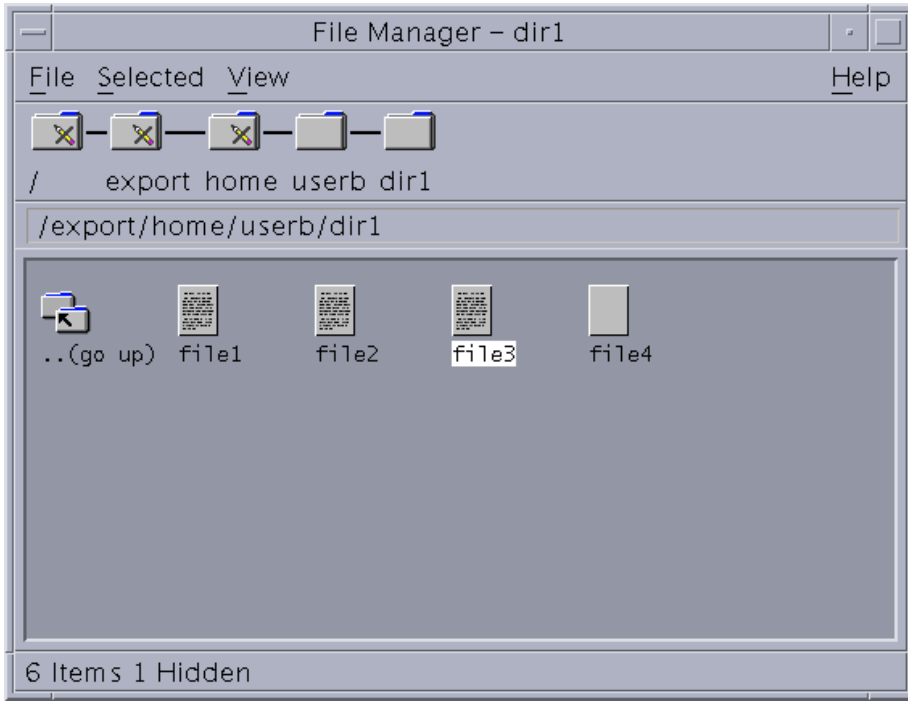


Figure 10-4 File Manager `dir1`, `file3` Window

In the File Manager GUI, select a file or directory to view its ACL. To display the file permissions, select Properties from the Selected menu.

The File Name field in the Properties window, as shown in Figure 10-5, displays the path and name of the selected file or directory. By default, the window shows the conventional UNIX permissions that exist on this file.

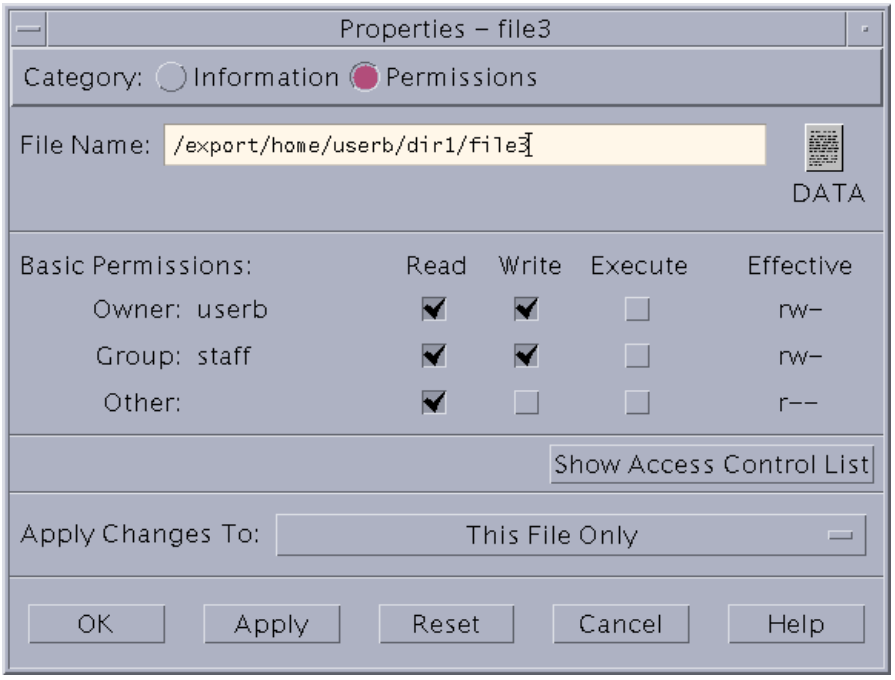


Figure 10-5 Properties – `file3` Window

To display the ACL, click Show Access Control List.

The Properties window expands and displays a list of any existing ACL entries, as shown in Figure 10-6. The ACL permissions contain the following fields:

Type	The type of users to whom you give ACL permissions.
Name	The name of the user or group listed in the Type field. The default group, default owning user, default owning group, default other, and default mask types do not have a name associated with them.
Requested	The read, write, and execute permissions for the specified values in the Type and Name fields.
Effective	The requested read, write, and execute permissions that the file has, after applying the file's ACL mask to the ACL entries.

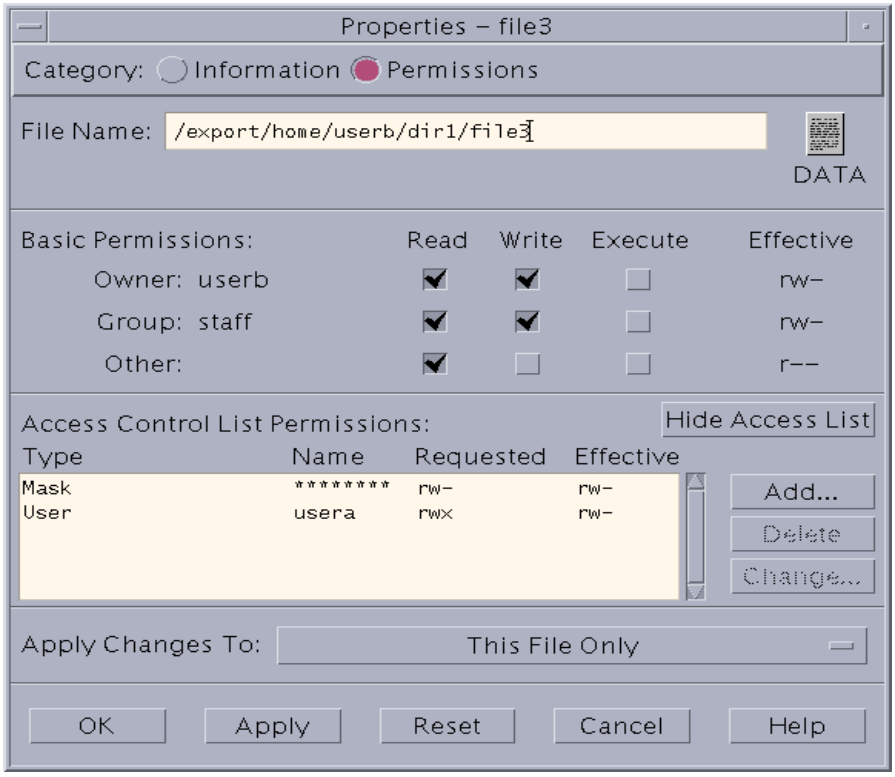


Figure 10-6 Permissions - `file3`

From the Properties window, you can perform any of the following:

- Click Add to advance to the Properties: Add Access List Entry window.
- Select an entry in the ACL list, and click Change to advance to the Properties: Change Access List Entry window.
- Select an entry in the ACL list, and click Delete to advance to the Properties: Delete Confirmation window.

Adding ACLs Using the File Manager GUI

Figure 10-7 shows the Properties: Add Access List Entry window, which appears when you click Add to add an entry to a file or to a folder's ACL

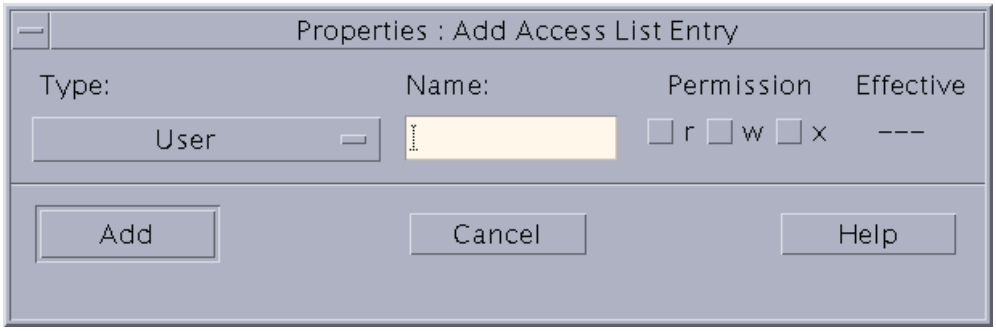


Figure 10-7 Properties: Add Access List Entry Window

Changing ACLs Using the File Manager GUI

Figure 10-8 shows the Properties: Change Access List Entry window, which appears when you click to Change to change an entry to a file or to a folder's ACL.

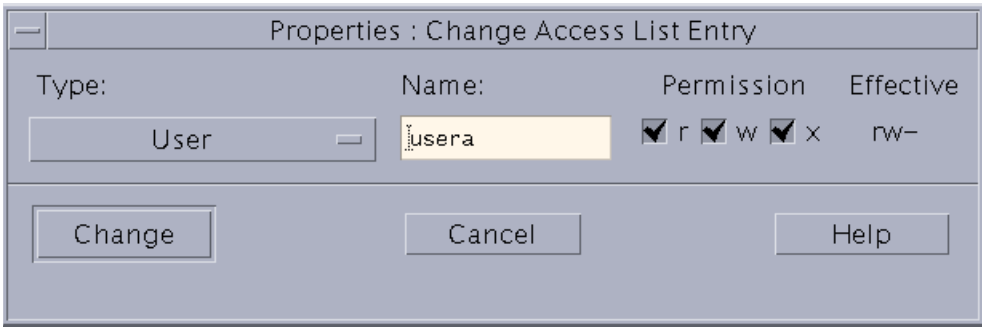


Figure 10-8 Properties: Change Access List Entry Window

Deleting ACLs Using the File Manager GUI

Figure 10-9 shows the confirmation that appears when you delete an ACL. To delete an ACL entry, click Delete in the Properties: Delete Confirmation window.

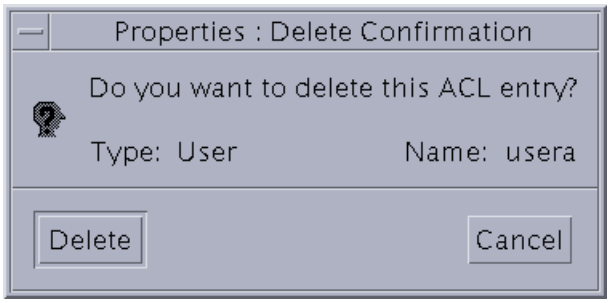


Figure 10-9 Properties: Delete Confirmation Window

To confirm the delete operation, click Delete.

Creating Default ACLs

Default ACLs can be described as the maximum discretionary access permissions that can be granted on files and directories. Default ACL entries provide a way to propagate ACL information automatically to files and directories. New files and directories inherit ACL information from their parent directory if that parent has an ACL that contains default entries.

You can set default ACL entries only on directories. You must set default ACL entries for the user, group, other, and ACL mask before you set a default ACL entry for an additional user or group. The following sections describe:

- How to add default ACL entries to a directory
- The effect on new subdirectories
- The effect on new files

Adding Default ACL Entries to a Directory

This example shows how to create a standard directory named `dir1`:

```
$ pwd
/export/home/userc
$ mkdir dir1
$ ls -l
total 2
drwxr-xr-x   2 userc    sysadmin     512 Apr 29 17:11 dir1
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file1
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file2
-rwxrw-r--+  1 userc    sysadmin       0 Apr 29 14:30 file3
```

This example displays the ACL list for the newly created standard directory:

```
$ getfacl dir1

# file: dir1
# owner: userc
# group: sysadmin
user::rwx
group::r-x              #effective:r-x
mask:r-x
other:r-x
```

Attempt to create a default ACL entry for an additional user.

```
$ setfacl -m default:user:usera:rwx dir1
Missing user/group owner, other, mask entry
aclcnt 5, file dir1
```

This `setfacl` command fails because the `dir1` directory has no default entry for the user, group, other, and ACL mask.

To add default permissions on a directory, use the `setfacl` command and precede the ACL entry types with the argument `default:`, or abbreviate the argument as `d:`.

```
$ setfacl -m d:u::rwx,d:g::r-x,d:o:r-x,d:m:r-x dir1
```

When you use the `ls -l` command, the `dir1` directory is now listed as containing a non-trivial ACL.

```
$ ls -l
total 2
drwxr-xr-x+  2 userc    sysadmin     512 Apr 29 17:11 dir1
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file1
-rwxrw-r--+  1 userc    sysadmin       0 Apr 18 15:44 file2
-rwxrw-r--+  1 userc    sysadmin       0 Apr 29 14:30 file3
```

Again, attempt to create a default ACL entry for an additional user. This time the command succeeds because default entries exist for the user, group, other, and ACL mask. Verify that the new entry exists.

```
$ setfacl -m default:user:usera:rwx dir1
```

Verify that the changes to `dir1` were caused by the creation of the default ACL entry.

```
$ getfacl dir1

# file: dir1
# owner: userc
# group: sysadmin
user::rwx
group::r-x              #effective:r-x
mask:r-x
other:r-x
default:user::rwx
default:user:usera:rwx
default:group::r-x
default:mask:r-x
default:other:r-x
```

Effect of Default ACLs on New Subdirectories

When a directory contains a default ACL, the permissions granted to the user, group, and other categories for the directory represent the intersection of mode 777, which is the UNIX default for directories without `umask` influence.

When a subdirectory is created in a directory containing default ACL entries, the permissions on the newly created subdirectory are generated according to the intersection between the default ACL entries and the permissions set at the initial time of creation of the directory.

The following sequence of commands shows that the default ACL entries are copied into newly created subdirectories.

When you create a new subdirectory after adding default ACL entries to the parent directory, the new subdirectory contains the default ACL entry from the parent. For example, the ACL entry is added to the new `subdir2` directory.

The following examples show the effect of default ACLs on new directories. Display the ACL for the parent directory `dir1`.

```
$ getfacl dir1

# file: dir1
# owner: userc
# group: sysadmin
user::rwx
group::r-x              #effective:r-x
mask:r-x
other:r-x
default:user::rwx
default:user:usera:rwx
default:group::r-x
default:mask:r-x
default:other:r-x
```

Create a new directory called `dir1/subdir1`, and display its ACL. The content of the ACL for `dir1/subdir1` is determined by the default entries associated with the `dir1` directory.

```
$ mkdir dir1/subdir1
$ ls -l dir1
total 2
drwxr-xr-x+  2 userc    sysadmin     512 Apr 30 08:01 subdir1
$ getfacl dir1/subdir1

# file: dir1/subdir1
# owner: userc
# group: sysadmin
user::rwx
group::r-x              #effective:r-x
mask:r-x
other:r-x
default:user::rwx
default:user:usera:rwx
default:group::r-x
default:mask:r-x
default:other:r-x
```

The permissions granted to the user, group, and other categories for `dir1/subdir1` represent the intersection of mode 777 (the UNIX default for directories without `umask` influence) with the default entries associated with `dir1`. Because the default entries for `dir1` happen to correspond with what the `mkdir` command would set in the absence of default ACL entries, the example of intersection is less than clear.

Change the default entries associated with `dir1` so they all reflect mode `rwx`. Verify that the ACL reflects the changes you made.

```
$ setfacl -m d:group::rwx,d:other:rwx,d:mask:rwx dir1
$ getfacl dir1

# file: dir1
# owner: userc
# group: sysadmin
user::rwx
group::r-x              #effective:r-x
mask:r-x
other:r-x
default:user::rwx
default:user:usera:rwx
default:group::rwx
default:mask:rwx
default:other:rwx
```

Create a new directory called `dir1/subdir2`, and display its ACL. The content of the ACL for `dir1/subdir2` is determined by the current default entries associated with `dir1`.

The permissions granted to the user, group, and other categories for `dir1/subdir2` represent the intersection of mode 777 (the UNIX default for directories without `umask` influence) with the default entries associated with `dir1`. Now that the default entries for `dir1` are set to `rwx`, the example of intersection is clearer.

```
$ mkdir dir1/subdir2
$ ls -l dir1
total 4
drwxr-xr-x+  2 userc    sysadmin     512 Apr 30 08:01 subdir1
drwxrwxrwx+  2 userc    sysadmin     512 Apr 30 08:04 subdir2
$ getfacl dir1/subdir2

# file: dir1/subdir2
# owner: userc
# group: sysadmin
user::rwx
group::rwx              #effective:rwx
mask:rwx
other:rwx
default:user::rwx
default:user:usera:rwx
default:group::rwx
default:mask:rwx
default:other:rwx
$
```

Effect of Default ACLs on New Files

When a file is created in a directory that contains default ACL entries, the new file will have permissions set according to the intersection of the default ACL entries and the permissions requested at creation time.

In the following example, you create a new file called `filea` beneath the `/dir/subdir2` directory, and display its ACL. The ACL for `filea` is determined by the default entries associated with the `dir1/subdir2` directory.

```
$ pwd
/export/home/userc
$ cd dir1/subdir2
$ touch filea
$ ls -l
total 0
-rw-rw-rw-+  1 userc    sysadmin       0 Apr 30 13:34 filea
$ getfacl filea

# file: filea
# owner: userc
# group: sysadmin
user::rw-
user:usera:rwx          #effective:rw-
group::rw-              #effective:rw-
mask:rw-
other:rw-
```

The permissions granted to the user, group, and other categories for `filea` represent the intersection of mode 666 (the UNIX default for files without `umask` influence) with the default entries associated with the`dir1/subdir2` directory. Because the default entries for the `dir1/subdir2` directory are set to `rwx`, the example of intersection is clear.

Note – The mask value does not exceed the permissions assigned to the group. Even though the `dir1/subdir2` directory lists `rwx` as the default mask value, files inherit only up to `rw-`.

The entry for `usera` was applied as a standard ACL entry and not as a default entry, because only directories replicate default entries. For each default ACL entry for additional users or groups found in the parent directory's list, files inherit a corresponding standard ACL entry.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using Access Control Lists (Level 1)

In this exercise, you create two files, and modify the ACLs associated with them.

Preparation

This exercise requires a user named `user10` and a group named `group1`. Refer to the lecture notes as necessary to perform the tasks listed.

Tasks

Perform the following tasks:

- Create the user named `user10`. Create a directory named `/var/test`. In this directory, create two files called `file1` and `file2`. Add a line of text to the file named `file1`. Record the permissions set on each file. Verify that the permissions and ACL information for `file1` agree.
- Set `440` permissions on `file1`. Change the user to `user10`, and attempt to read the file. Record the result. Exit your `su` session. Create an ACL entry that grants the user named `user10` read permission. Verify that the new ACL entry exists, and record how the entry's presence is indicated in the permissions list. Change the user to `user10` and read the file again. Record the result.
- Display the ACL for the file named `file2`. Verify that the group and mask permissions match. Use the `chmod` command to grant full permissions to the group that owns the file named `file2`. Verify that the mask and group permissions match. Set the mask permissions to read-only for the file named `file2`. Verify that the group and mask permissions match.
- Add the group named `group1` if it does not exist. Add ACL entries that grant read and execute permissions for `group1` and only grant execute permissions for `user10`. Record the effective permissions for `user10` and `group1`. Set the mask to grant read, write, and execute permissions. Record the effective permissions for `user10` and `group1`. Record the permissions for the group that owns the file named `file2`.

Exercise: Using Access Control Lists (Level 2)

In this exercise, you create two files, and modify the ACLs associated with them.

Preparation

This exercise requires a user named `user10` and a group named `group1`. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

Perform the following tasks:

- Create the user named `user10`. Create a directory named `/var/test`. In this directory, create two files called `file1` and `file2`. Add a line of text to the file named `file1`. Record the permissions set on each file. Verify that the permissions and ACL information for `file1` agree.
- Set `440` permissions on `file1`. Change the user to `user10`, and attempt to read the file. Record the result. Exit your `su` session. Create an ACL entry that grants the user named `user10` read permission. Verify that the new ACL entry exists, and record how the entry's presence is indicated in the permissions list. Change the user to `user10` and read the file again. Record the result.
- Display the ACL for the file named `file2`. Verify that the group and mask permissions match. Use the `chmod` command to grant full permissions to the group that owns the file named `file2`. Verify that the mask and group permissions match. Set the mask permissions to read-only for the file named `file2`. Verify that the group and mask permissions match.
- Add the group named `group1` if it does not exist. Add ACL entries that grant read and execute permissions for `group1` and only grant execute permissions for `user10`. Record the effective permissions for `user10` and `group1`. Set the mask to grant read, write, and execute permissions. Record the effective permissions for `user10` and `group1`. Record the permissions for the group that owns the file named `file2`.

Tasks

Complete the following steps:

1. Log in as the `root` user, and open a terminal window.

   _____

2. If `user10` does not exist on your system, create it with a user ID of `1010`, a group ID of `10`, a Korn login shell, and a home directory of `/export/home/user10`.

   _____

3. Create the directory named `/var/test`, and change directories to that location.

   _____

4. Create two new files named `file1` and `file2`. Record the permissions applied to each.

   _____

5. Use the `echo` command to create `file1`, and have the file contain the text string "Success for life."

   _____

6. Display the ACL for the file named `file1`.

   _____

   Do the permissions in the ACL match the permissions indicated by the `ls` command?

   _____

7. Change permissions on the file named `file1` so that only the owner (`root`) and group (`other`) have read access.

   _____

8. Change your user identity to `user10`.

   _____

9. Display the contents of the file named `file1`.

   What is the output?

   _____

10. Exit your `su` session.

    _____

11. Use the `setfacl` command to add an ACL entry that allows read access for `user10` to the ACL for the file named `file1`.

    _____

12. Verify that the new ACL entry exists.

    _____

13. Change your user identity back to `user10`.

    _____

14. Use the `ls` command to display the permissions applied to the file named `file1`.

    According to these permissions, does `user10` have read access?

    _____

    What indicates that an additional ACL entry exists for `file1`?

    _____

15. Attempt to display the contents of the file named `file1`.

    _____

    What is the result?

    _____

16. Exit your `su` session when finished.

    _____

17. Display the ACL for the file named `file2`.

    _____

    Do the group permissions match the permissions associated with the mask entry?

    _____

18. Change the permission mode to grant read, write, and execute permissions to the group that owns the file named `file2`.

    _____

19. Display the ACL and a long listing for the file named `file2`.

    _____

    Do the mask permissions match the group permissions?

    _____

20. Set the mask permissions for the file named `file2` to read-only.

    _____

21. Display the ACL and a long listing for `file2`.

    _____

    Do the mask permissions match the group permissions?

    _____

    In the long listing output, do you find an indication that `file2` has additional ACL entries?

    _____

22. If `group1` does not exist on your system, create it with a group ID of `101`.

    _____

23. Add an ACL entry for the group named `group1` to the file named `file2`. Grant only read and execute permissions for this group.

    _____

24. Add an ACL entry for the user named `user10` to `file2`. Grant only execute permissions for this user.

    _____

25. Verify the current ACL permissions for `file2`.

    What are the effective permissions for `user10` and `group1`?

    _____

26. Modify the owner's group permission to read and execute, and recalculate the mask.
27. Verify the effective permissions for `user10` and `group1`.

    Do the effective permissions for `user10` and `group1` match the mask? If not, what permissions were specifically granted?

    _____

    Did changing the permissions for the group that owns the file affect the mask permissions?

    _____

Exercise: Using Access Control Lists (Level 3)

In this exercise, you create two files, and modify the ACLs associated with them.

Preparation

This exercise requires a user named `user10` and a group named `group1`. Refer to the lecture notes as necessary to perform the tasks listed.

Task Summary

Perform the following tasks:

- Create the user named `user10`. Create a directory named `/var/test`. In this directory, create two files called `file1` and `file2`. Add a line of text to the file named `file1`. Record the permissions set on each file. Verify that the permissions and ACL information for `file1` agree.
- Set `440` permissions on `file1`. Change the user to `user10`, and attempt to read the file. Record the result. Exit your `su` session. Create an ACL entry that grants the user named `user10` read permission. Verify that the new ACL entry exists, and record how the entry's presence is indicated in the permissions list. Change the user to `user10` and read the file again. Record the result.
- Display the ACL for the file named `file2`. Verify that the group and mask permissions match. Use the `chmod` command to grant full permissions to the group that owns the file named `file2`. Verify that the mask and group permissions match. Set the mask permissions to read-only for the file named `file2`. Verify that the group and mask permissions match.
- Add the group named `group1` if it does not exist. Add ACL entries that grant read and execute permissions for `group1` and only grant execute permissions for `user10`. Record the effective permissions for `user10` and `group1`. Set the mask to grant read, write, and execute permissions. Record the effective permissions for `user10` and `group1`. Record the permissions for the group that owns the file named `file2`.

Tasks and Solutions

Complete the following steps:

1. Log in as the `root` user, and open a terminal window.
2. If `user10` does not exist on your system, create it with a user ID of `1010`, a group ID of `10`, a Korn login shell and a home directory of `/export/home/user10`.

```
# useradd -u 1010 -g 10 -d /export/home/user10 -m \
-s /bin/ksh -c "SA-299 Student" user10
```

3. Create the directory named `/var/test`, and change directories to that location.

```
# mkdir /var/test
# cd /var/test
```

4. Create two new files named `file1` and `file2`. Record the permissions applied to each.
5. Use the `echo` command to create `file1`, and have the file contain the text string "Success for life."

```
# echo "Success for Life" > file1
# touch file2
# ls -l
```

Both files use `-rw-r--r--` *(644) permissions.*

6. Display the ACL for the file named `file1`.

```
# getfacl file1
```

Do the permissions in the ACL match the permissions indicated by the `ls` command?

Yes, they should.

7. Change permissions on the file named `file1` so that only the owner (`root`) and group (`other`) have read access.

```
# chmod 440 file1
```

8. Change your user identity to `user10`.

```
# su user10
$
```

9. Display the contents of the file named `file1`.

What is the output?

```
$ cat file1
```

The following error message displays: `cat: cannot open file1`.

10. Exit your `su` session.

```
$ exit
```

11. Use the `setfacl` command to add an ACL entry that allows read access for `user10` to the ACL for the file named `file1`.

```
# setfacl -m user:user10:4 file1
```

12. Verify that the new ACL entry exists.

```
# getfacl file1
```

13. Change your user identity back to `user10`.

```
# su user10
$
```

14. Use the `ls` command to display the permissions applied to the file named `file1`.

```
$ ls -l file1
```

According to these permissions, does `user10` have read access?

No.

What indicates that an additional ACL entry exists for `file1`?

The "+" symbol at the end of the permissions string.

15. Attempt to display the contents of the file named `file1`.

```
$ cat file1
```

What is the result?

The file content appears.

16. Exit your `su` session when finished.

```
$ exit
#
```

17. Display the ACL for the file named `file2`.

```
# getfacl file2
```

Do the group permissions match the permissions associated with the mask entry?

Yes.

18. Change the permission mode to grant read, write, and execute permissions to the group that owns the file named `file2`.

```
# chmod g=rwx file2
```

19. Display the ACL and a long listing for the file named `file2`.

```
# getfacl file2
# ls -l file2
```

Do the mask permissions match the group permissions?

Yes.

20. Set the mask permissions for the file named `file2` to read-only.

```
# setfacl -m mask:r-- file2
```

21. Display the ACL and a long listing for `file2`.

```
# getfacl file2
# ls -l file2
```

Do the mask permissions match the group permissions?

Yes.

In the long listing output, do you find an indication that `file2` has additional ACL entries?

No.

22. If `group1` does not exist on your system, create it with a group ID of `101`.

```
# groupadd -g 101 group1
```

23. Add an ACL entry for the group named `group1` to the file named `file2`. Grant only read and execute permissions for this group.

```
# setfacl -m group:group1:5 file2
```

24. Add an ACL entry for the user named `user10` to `file2`. Grant only execute permissions for this user.

```
# setfacl -m user:user10:1 file2
```

25. Verify the current ACL permissions for `file2`.

```
# getfacl file2
```

What are the effective permissions for `user10` and `group1`?

The user named `user10` *has no permissions, and* `group1` *has read permission.*

26. Modify the owner's group permission to read and execute, and recalculate the mask.

```
# setfacl -r -m g::5 file2
```

27. Verify the effective permissions for `user10` and `group1`.

```
# getfacl file2
```

Do the effective permissions for `user10` and `group1` match the mask? If not, what permissions were specifically granted?

The permissions should match what you specifically granted.

Did changing the permissions for the group that owns the file affect the mask permissions?

Yes. Recalculating the mask after changing the group permissions caused the mask to change accordingly.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Configuring Role-Based Access Control (RBAC)

Objectives

Role-based access control (RBAC) is an alternative to the all-or-nothing superuser model. RBAC uses the security principle of least privilege. No user should be given more privilege than necessary for performing the user's job. RBAC makes it possible for an organization to separate superusers' capabilities and assign these capabilities to specific users or to special user accounts that are called roles. Roles can be assigned to specific individuals, according to their job needs. Upon completion of this module, you should be able to:

- Describe RBAC fundamentals
- Describe component interaction within RBAC
- Manage RBAC by using the Solaris™ Management Console
- Manage RBAC by using the command line

The following course map shows how this module fits into the current instructional goal.

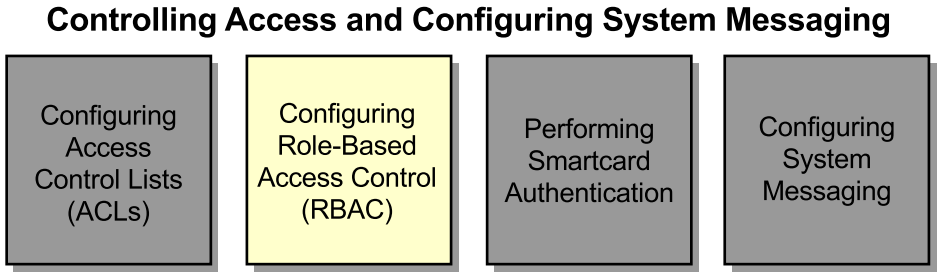


Figure 11-1 Course Map

Introducing RBAC Fundamentals

In conventional UNIX systems, the `root` user (also referred to as the superuser) is the most powerful user, with the ability to read and write to any file, run all programs, and send `kill` signals to any process. Anyone who can become superuser can modify a site's firewall, alter the audit trail, and read confidential records.

In systems implementing RBAC, individual users can be assigned to roles, such as system administrator, network administrator, or operator. Individual users may also be granted authorization to specific applications. Roles are associated with a rights profile. The rights profile lists the rights that are assigned to roles so that those roles can run specific commands and applications. The users, roles, profiles, and privileged commands are defined in four databases.

Roles

A role is a special identity for running privileged applications or commands that can be assumed by assigned users only.

While no predefined roles are shipped with the Solaris 9 OE, roles can be associated with a defined profile that is already set up. To define a role, you assign the rights profile to the role.

Note – You can also set up the `root` user as a role through a manual process. This approach prevents users from logging in directly as the `root` user. Therefore, they must log in as themselves first, and then use the `su` command to assume the role.

Rights Profiles

A right, also known as a profile or a rights profile, is a collection of privileges that can be assigned to a role or user. A rights profile can consist of authorizations, commands with `setuid` or `setgid` permissions (referred to as security attributes), and other rights profiles. The Solaris Management Console Rights tool lets you inspect the contents of rights profiles.

Many examples of rights profiles are shipped with the Solaris 9 OE. These example rights profiles provide a basis from which you can create your own profiles. Some of the rights profiles are described in Table 11-1.

Table 11-1 Rights Profiles and Role Descriptions

Rights Profile	Role Description
`All`	Provides a role access to commands without security attributes. In a non-RBAC system, these commands would be all commands that do not need `root` permission to run.
`Primary Administrator`	Designed specifically for the Primary Administrator role. In a non-RBAC system, this role would be equivalent to the `root` user.
`System Administrator`	Designed specifically for the System Administrator role. The System Administrator rights profile uses discrete supplementary profiles to create a powerful role.
`Operator`	Designed specifically for the Operator role. The Operator rights profile uses a few discrete supplementary profiles to create a basic role.
`Basic Solaris User`	Enables users to perform tasks that are not related to security.
`Printer Management`	Dedicated to the single area of printer administration.

The rights profiles include a pointer to help files. Help files are written in Hypertext Markup Language (HTML) and you can customize them, if required. These help files exist in the `/usr/lib/help/auths/locale/C` directory.

Authorizations

An authorization is a permission that you can assign to a role or to a user. Applications define most authorizations. You can give authorization to a user or role, but you generally cannot define new authorizations.

You can also embed authorizations in a rights profile for performing a class of actions that are otherwise prohibited by the security policy. RBAC-compliant applications check the user's or role's authorization before a user or role gets access to the application or to the specific operations within it.

Table 11-2 shows how a hierarchy can be established using authorizations.

Table 11-2 Role and Authorization Relationships

Role	Authorization	Action
Operator	`solaris.admin.usermgr.read`	Provides read but no write access to users' configuration files.
System Administrator	`solaris.admin.usermgr.read` `solaris.admin.usermgr.write`	Provides read and write access to users' configuration files. Cannot change passwords.

Primary Administrator	`solaris.admin.usermgr.read`
	`solaris.admin.usermgr.write`
	`solaris.admin.usermgr.pswd`
		Provides read, write, and password access to users' configuration files.

An authorization that ends with the suffix `grant` permits a user or role to delegate to other users any assigned authorizations that begin with the same prefix. For example, a role with the authorizations `solaris.admin.usermgr.grant` and `solaris.admin.usermgr.read` can delegate the `solaris.admin.usermgr.read` authorization to another user. A role with the `solaris.admin.usermgr.grant` and `solaris.admin.usermgr.*` can delegate any of the authorizations with the `solaris.admin.usermgr` prefix to other users.

Administrator Profile Shells

When a user runs the `su` command to assume a role, profile shells launch from within the parent shell. The profile shells are `pfsh`, `pfcsh`, and `pfksh`. These profile shells correspond to Bourne shell (`sh`), C shell (`csh`), and Korn shell (`ksh`), respectively.

Purpose of the Profile Shells

A profile shell is a special type of shell that enables access to the privileged applications that are assigned to the profile. The standard UNIX shells are not aware of the RBAC databases, and do not consult them.

When the user executes a command, the profile shell searches the role's profile and associated commands. If the same command appears in more than one profile, the profile shell uses the first matching entry. The `pfexec` command executes the command with the attributes specified in the database.

Each profile shell is called from within its corresponding shell. The shells are, in effect, rooted in the same command. In other words, the shell and the profile shell have the same inode number:

```
# ls -i /usr/bin/sh /usr/bin/pfsh
    247742 /usr/bin/pfsh      247742 /usr/bin/sh
# ls -i /usr/bin/csh /usr/bin/pfcsh
    247691 /usr/bin/csh       247691 /usr/bin/pfcsh
# ls -i /usr/bin/ksh /usr/bin/pfksh
    247746 /usr/bin/ksh       247746 /usr/bin/pfksh
```

Introducing the Component Interaction Within RBAC

There are four databases that are used by RBAC. The fields in these databases are interrelated. Figure 11-2 shows how these databases are related.

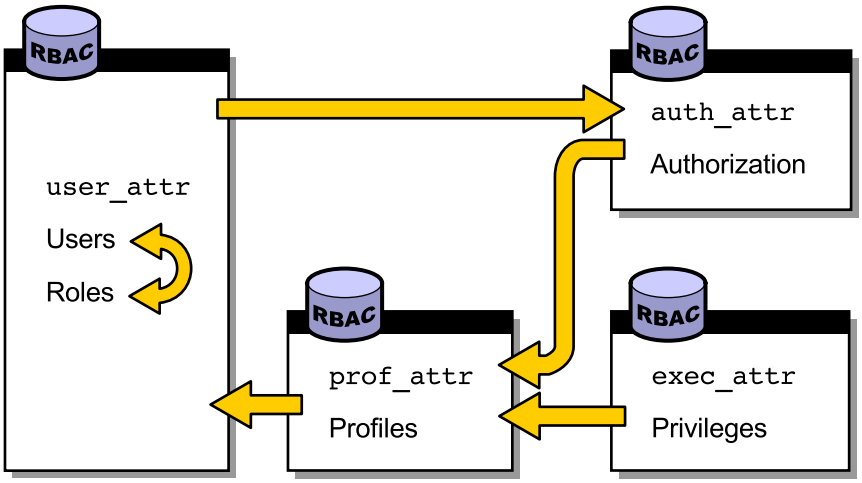


Figure 11-2 RBAC Databases

Introducing the RBAC Databases

In addition to the traditional authentication mechanism in the Solaris OE, RBAC uses four databases to provide users access to privileged operations. These databases are described in Table 11-3.

Table 11-3 RBAC Databases

Database	Contents
`/etc/user_attr`	The extended user attributes database, which associates users and roles with authorizations and rights profiles in addition to the `/etc/passwd`, `/etc/group`, and `/etc/shadow` files.
`/etc/security/prof_attr`	The rights profile attributes database, which defines profiles, lists the profile's assigned authorizations, and identifies the associated help file. Based on what a profile is designed to do, you can logically name profiles.

Table 11-3 RBAC Databases (Continued)

Database	Contents
`/etc/security/exec_attr`	The execution attributes database, which defines the privileged operations assigned to a profile.
`/etc/security/auth_attr`	The authorization attributes database, which defines authorizations and their attributes. This database also identifies the associated help file.

In addition to the four databases that configure specific rights profiles, roles, and authorizations, the `/etc/security/policy.conf` file provides system default authorizations for users.

Using the RBAC Delimiters

The RBAC databases uses a common set of delimiters. These delimiters are as follows:

- Colon (`:`) – Use the colon as a field separator within each database; for example:

name`:`*qualifier*`:`*res1*`:`*res2*`:`*attr*

- Semicolon (`;`) – Use the semicolon to separate key-value pairs within attribute fields; for example:

`...:attribute_type=`*value*`;attribute_profile=`*value*`;attribute_auth=`*value*

- Comma (`,`) – Use the comma to separate an ordered list within a specific attribute value; for example:

```
...;attribute_profile=profile_access1,profile_access2,profile_access3;...
```

- Dot (`.`) – Use the dot to separate the prefix from suffixes within authorization names to define execution profiles with finer granularity; for example:

```
solaris.system.date:::Set Date & Time::help=SysDate.html
```

The /etc/user_attr Database

The /etc/user_attr database contains user and role information that supplements the /etc/passwd and /etc/shadow databases. The /etc/user_attr database lists the profiles and authorizations associated with the defined roles. The /etc/user_attr database also associates users with their roles. You can assign users, roles, authorizations, and profiles. Figure 11-3 shows how the roles and users are associated within the database.

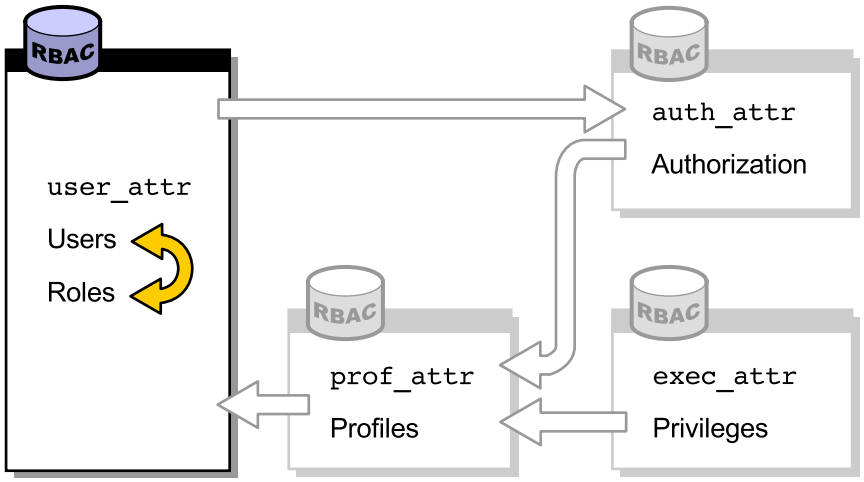


Figure 11-3 The /etc/user_attr Database

The fields in the /etc/user_attr database are separated by colons, as follows:

user:*qualifier*:*res1*:*res2*:*attr*

where:

user	The name of the user, as specified in the passwd database.
qualifier	Reserved for future use.
res1	Reserved for future use.
res2	Reserved for future use.

attr An optional list of semicolon-separated (;) key-value pairs that describes the security attributes to be applied when the user runs commands. There are four valid keys: `type`, `auths`, `roles`, and `profiles`.

- `type` – Can be `normal` or `role`. A role is assumed by a `normal` user after the user has logged in.
- `auths` – Specifies a list of authorization names chosen from names defined in the `auth_attr` database. Authorization names can include the asterisk (`*`) character as a wildcard. For example, `solaris.device.*` means all of the Solaris OE device authorizations.
- `profiles` – Specifies a list of profile names chosen from the `/etc/prof_attr` database. The order of profiles works similarly to UNIX search paths. The first profile in the list that contains the command to be executed defines which (if any) attributes are to be applied to the command.
- `roles` – Specifies a list of role names. Roles are defined in the same `/etc/user_attr` database. Roles are indicated by setting the `type` value to `role`. Roles cannot be assigned to other roles.

Figure 11-4 shows a portion of a `/etc/user_attr` database. The user `johndoe` is a `normal` user. The user is given the `role` of `sysadmin`. The user `sysadmin` is a `role` user. When assuming the `sysadmin` role, `johndoe` has access to specific profiles, defined as Device Management, Filesystem Management, and Printer Management profiles.

```
root::::type=normal;auth=solaris.*,solaris.grant
sysadmin::::type=role;profiles=Device Management,Filesystem
Management,Printer Management
johndoe::::type=normal;auths=solaris.system.date;roles=sysadmin
```

Figure 11-4 User and Role Association

The /etc/security/prof_attr Database

The /etc/security/prof_attr database holds the rights profiles, as shown in Figure 11-5.

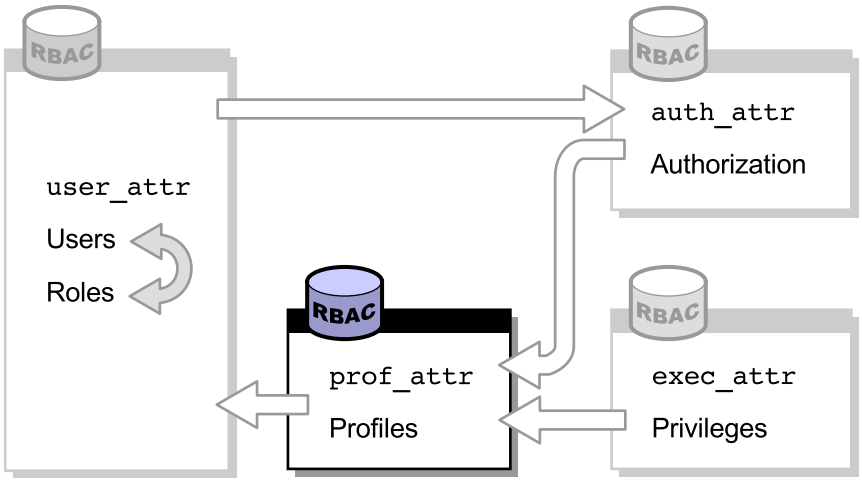


Figure 11-5 The prof_attr Database

The profiles consist of a name, description, authorizations, and help file location. The fields in the /etc/security/prof_attr database are separated by colons:

profname:*res1*:*res2*:*desc*:*attr*

where:

profname	The name of the profile. Profile names are case sensitive.
res1	Reserved for future use.
res2	Reserved for future use.
desc	A long description. This field explains the purpose of the profile, including what type can use it. The long description should be suitable for displaying in the help text of an application.

attr — An optional list of key-value pairs separated by semicolons (;) that describes the security attributes to apply to the object upon execution. You can specify zero or more keys. The two valid keys are `help` and `auths`:

- `help` – Identifies a help file.
- `auths` – Specifies a list of authorization names chosen from those names defined in the `auth_attr` database. Authorization names can be specified with the asterisk (*) character as a wildcard.

In the following example, the `Printer Management` rights profile is a supplementary rights profile that is assigned to the `Operator` rights profile and the `System Administrator` rights profile.

```
# grep 'Printer Management' /etc/security/prof_attr
Printer Management:::Manage printers, daemons,\
spooling:help=RtPrntAdmin.html;auths=solaris.admin.printer.read,\
solaris.admin.printer.modify,solaris.admin.printer.delete
Operator:::Can perform simple administrative tasks:profiles=Printer
Management,Media Backup,All;help=RtOperator.html
System Administrator:::Can perform most non-security administrative\
tasks:profiles=Audit Review,Printer Management,Cron Management,Device\
Management,File System Management,Mail Management,Maintenance and
Repair,Media Backup,Media Restore,Name Service Management,Network
Management,Object Access Management,Process Management,Software
Installation,User Management,All;help=RtSysAdmin.html
```

Figure 11-6 shows one relationship between the /etc/security/prof_attr and the /etc/user_attr databases. The Printer Management profile, which is defined in the /etc/security/prof_attr database, is assigned to the sysadmin role in the /etc/user_attr database.

From the /etc/security/prof_attr database:

```
Printer Management:::Manage printers, daemons,\
spooling:help=RtPrntAdmin.html;auths=solaris.admin.printer.read,\
solaris.admin.printer.modify,solaris.admin.printer.delete
```

From the /etc/user_attr database:

```
root::::type=normal;auth=solaris.*,solaris.grant
sysadmin::::type=role;profile=Device Management,Printer Management
...
```

Figure 11-6 User and Profile Association

Figure 11-7 shows the relationship between the /etc/security/prof_attr and the /etc/security/auth_attr databases. The Printer Management profile is defined in the /etc/security/prof_attr database as having all authorizations, beginning with the solaris.admin.printer. string, assigned to it. These authorizations are defined in the /etc/security/auth_attr database.

From the /etc/security/prof_attr database:

```
Printer Management:::Manage printers, daemons, spooling: \
help=RtPrntAdmin.html;auths=solaris.admin.printer.read, \
solaris.admin.printer.modify,solaris.admin.printer.delete
```

From the /etc/security/auth_attr database:

```
solaris.admin.printer.modify:::Update Printer Information:: \
help=AuthPrinterModify.html
solaris.admin.printer.delete:::Delete Printer Information:: \
help=AuthPrinterDelete.html
solaris.admin.printer.:::Printer Information::help=AuthPrinterHeader.html
solaris.admin.printer.read:::View Printer Information:: \
help=AuthPrinterRead.html
```

Figure 11-7 Profile and Authorization Association

The /etc/security/exec_attr Database

The /etc/security/exec_attr database holds the execution attributes. An execution attribute associated with a profile is a command or a script that contains a command with options (because the only way to add options to a command is by using a script). Only the users and roles assigned to this profile can run the command with special security attributes. Special security attributes refer to attributes, such as UID, EUID, GID, and EGID, that can be added to a process when the command is run. The definitions of the execution attributes are stored in the /etc/security/exec_attr database. Figure 11-8 shows the /etc/security/exec_attr database.

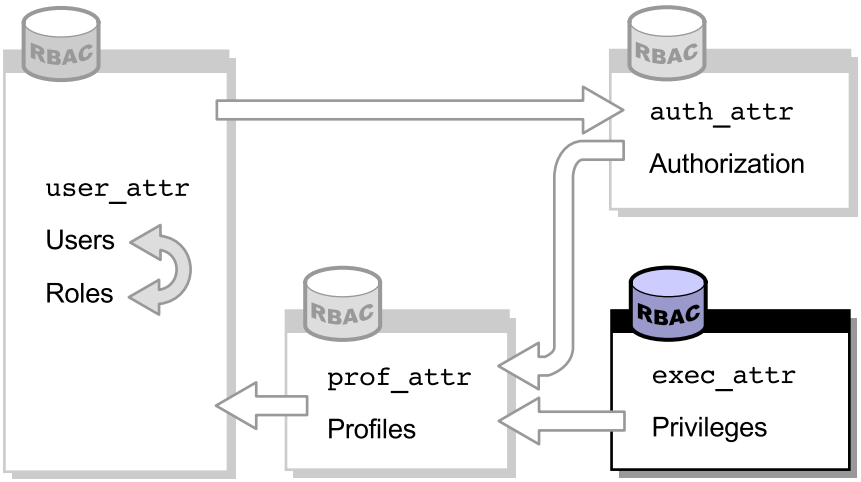


Figure 11-8 The exec_attr Database

The fields in the /etc/security/exec_attr database are separated by colons:

name:*policy*:*type*:*res1*:*res2*:*id*:*attr*

where:

name	The name of the profile. Profile names are case sensitive.
policy	The security policy associated with this entry. The suser (superuser policy model) is the only valid policy entry.
type	The type of entity whose attributes are specified. The only valid type is cmd (command).
res1	Reserved for future use.
res2	Reserved for future use.

id — A string identifying the entity. You can use the asterisk (*) wildcard. Commands should have the full path or a path with a wildcard. To specify arguments, write a script with the arguments, and point the *id* to the script.

attr — An optional list of key-value pairs that describes the security attributes to apply to the entity when executed. You can specify zero or more keys. The list of valid key words depends on the policy being enforced. There are four valid keys: `euid`, `uid`, `egid`, and `gid`.

- `euid` and `uid` – Contain a single user name or a numeric user ID. Commands designated with `euid` run with the effective UID indicated, which is similar to setting the `setuid` bit on an executable file. Commands designated with `uid` run with both the real and effective UIDs set to the UID you specify.
- `egid` and `gid` – Contain a single group name or numeric group ID. Commands designated with `egid` run with the effective GID indicated, which is similar to setting the `setgid` bit on an executable file. Commands designated with `gid` run with both the real and effective GIDs set to the GID you specify.

The following example is part of a `/etc/security/exec_attr` database with some typical values:

```
...
Printer Management:suser:cmd:::/usr/sbin/accept:euid=lp
Printer Management:suser:cmd:::/usr/ucb/lpq:euid=0
Printer Management:suser:cmd:::/etc/init.d/lp:euid=0
Printer Management:suser:cmd:::/usr/bin/lpstat:euid=0
Printer Management:suser:cmd:::/usr/lib/lp/lpsched:uid=0
...
```

Figure 11-9 shows the relationship between the /etc/security/exec_attr and /etc/security/prof_attr databases.

From the /etc/security/prof_attr database:

```
Printer Management:::Manage printers, daemons,
spooling:help=RtPrntAdmin.html;auths=solaris.admin.printer.read,solaris.a
dmin.printer.modify,solaris.admin.printer.delete
```

From the /etc/security/exec_attr database:

```
Printer Management:suser:cmd:::/usr/sbin/accept:euid=lp
Printer Management:suser:cmd:::/usr/ucb/lpq:euid=0
Printer Management:suser:cmd:::/etc/init.d/lp:euid=0
Printer Management:suser:cmd:::/usr/bin/lpstat:euid=0
Printer Management:suser:cmd:::/usr/lib/lp/lpsched:uid=0
Printer Management:suser:cmd:::/usr/sbin/lpfilter:euid=lp
Printer Management:suser:cmd:::/usr/bin/lpset:egid=14
Printer Management:suser:cmd:::/usr/sbin/lpadmin:egid=14
Printer Management:suser:cmd:::/usr/sbin/lpsystem:uid=0
Printer Management:suser:cmd:::/usr/sbin/lpmove:euid=lp
Printer Management:suser:cmd:::/usr/sbin/lpshut:euid=lp
Printer Management:suser:cmd:::/usr/bin/cancel:euid=0
Printer Management:suser:cmd:::/usr/bin/disable:euid=lp
Printer Management:suser:cmd:::/usr/sbin/lpforms:euid=lp
Printer Management:suser:cmd:::/usr/sbin/reject:euid=lp
Printer Management:suser:cmd:::/usr/ucb/lprm:euid=0
Printer Management:suser:cmd:::/usr/bin/enable:euid=lp
Printer Management:suser:cmd:::/usr/sbin/lpusers:euid=lp
```

Figure 11-9 Profile and Execution Association

The Printer Management profile lists execution attributes (or commands) with the appropriate security attributes assigned in the /etc/security/exec_attr database.

The /etc/security/auth_attr Database

An authorization is an RBAC feature that grants access to restricted functions. It identifies, by a unique string, what is being authorized, as well as who created the authorization.

You cannot create new authorizations. However, system programmers can create and assign authorizations to applications.

Certain privileged programs check authorizations to determine whether users can execute restricted functionality. For example, the `solaris.jobs.admin` authorization is required for a user to edit another user's `crontab` file.

All authorizations are stored in the `/etc/security/auth_attr` database. You can assign authorizations directly to users or roles in the `/etc/user_attr` database. You can also assign authorizations to rights profiles, which are assigned to roles.

Figure 11-10 shows the `/etc/security/auth_attr` database.

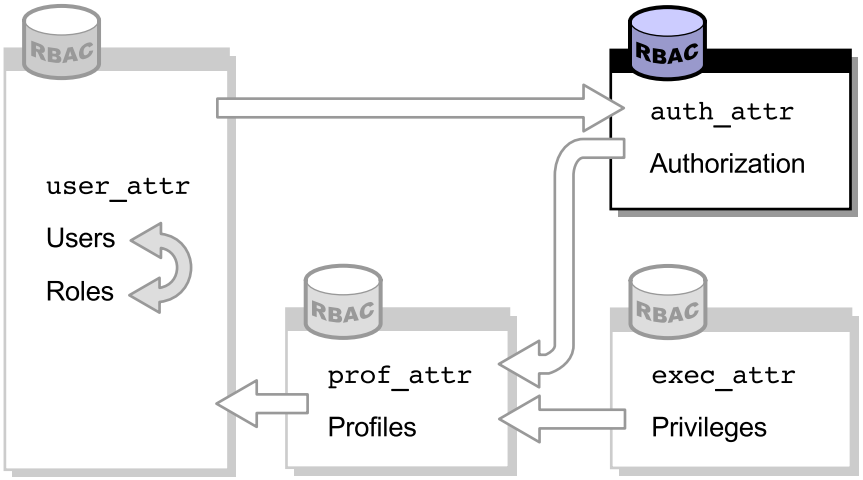


Figure 11-10 The `auth_attr` Database

The fields in the /etc/security/auth_attr database are separated by colons, as follows:

authname:*res1*:*res2*:*short_desc*:*long_desc*:*attr*

where:

authname — A unique character string that identifies the authorization in the *prefix.suffix[.]* format. Authorizations for the Solaris OE use solaris as a prefix. All other authorizations use a prefix that begins with the reverse-order Internet domain name of the organization that creates the authorization (for example, com.xyzcompany). The suffix indicates what is being authorized, typically the functional area and operation.

When there is no suffix (that is, the *authname* consists of a prefix, a functional area, and ends with a period), the *authname* serves as a heading for use by applications in their GUI rather than as an authorization. The *authname* solaris.printmgr. is an example of a heading.

When *authname* ends with the word grant, the *authname* serves as a grant authorization and lets the user delegate related authorizations (that is, authorizations with the same prefix and functional area) to other users. The *authname* solaris.printmgr.grant is an example of a grant authorization. It gives the user the right to delegate such authorizations as solaris.printmgr.admin and solaris.printmgr.nobanner to other users.

res1 — Reserved for future use.

res2 — Reserved for future use.

short_desc — A concise name for the authorization that is suitable for displaying in user interfaces.

long_desc — A long description. This field identifies the purpose of the authorization, the applications in which it is used, and the type of user who wants to use it. The long description can be displayed in the help text of an application.

attr — An optional list of key-value pairs that describes the attributes of an authorization. There can be zero or more keys. For example, the keyword help identifies a help file.

The following is an example of an /etc/security/auth_attr database, with some typical values:

```
solaris.*:::Primary Administrator::help=PriAdmin.html
solaris.grant:::Grant All Rights::help=PriAdmin.html
...
solaris.device.:::Device Allocation::help=DevAllocHeader.html
solaris.device.allocate:::Allocate Device::help=DevAllocate.html
solaris.device.config:::Configure Device Attributes::help=DevConfig.html
solaris.device.grant:::Delegate Device Administration::help=DevGrant.html
solaris.device.revoke:::Revoke or Reclaim Device::help=DevRevoke.html
```

Note – The solaris.device. entry is defined as a heading, because it ends in a dot (.). Headings are used by the GUI to organize families of authorizations.

Figure 11-11 shows the relationship between the /etc/security/auth_attr and the /etc/user_attr databases. The solaris.system.date authorization, which is defined in the /etc/security/auth_attr database, is assigned to the user johndoe in the /etc/user_attr database.

From the /etc/security/auth_attr database:

```
solaris.*:::Primary Administrator::help=PriAdmin.html
...
solaris.system.date:::Set Date & Time::help=SysDate.html
...
```

From the /etc/user_attr database:

```
johndoe::::type=normal;auths=solaris.system.date;roles=sysadmin
```

Figure 11-11 User, Role, and Authorization Association

Relationships Between the Four RBAC Databases

Figure 11-12 shows how the fields of the four databases are related.

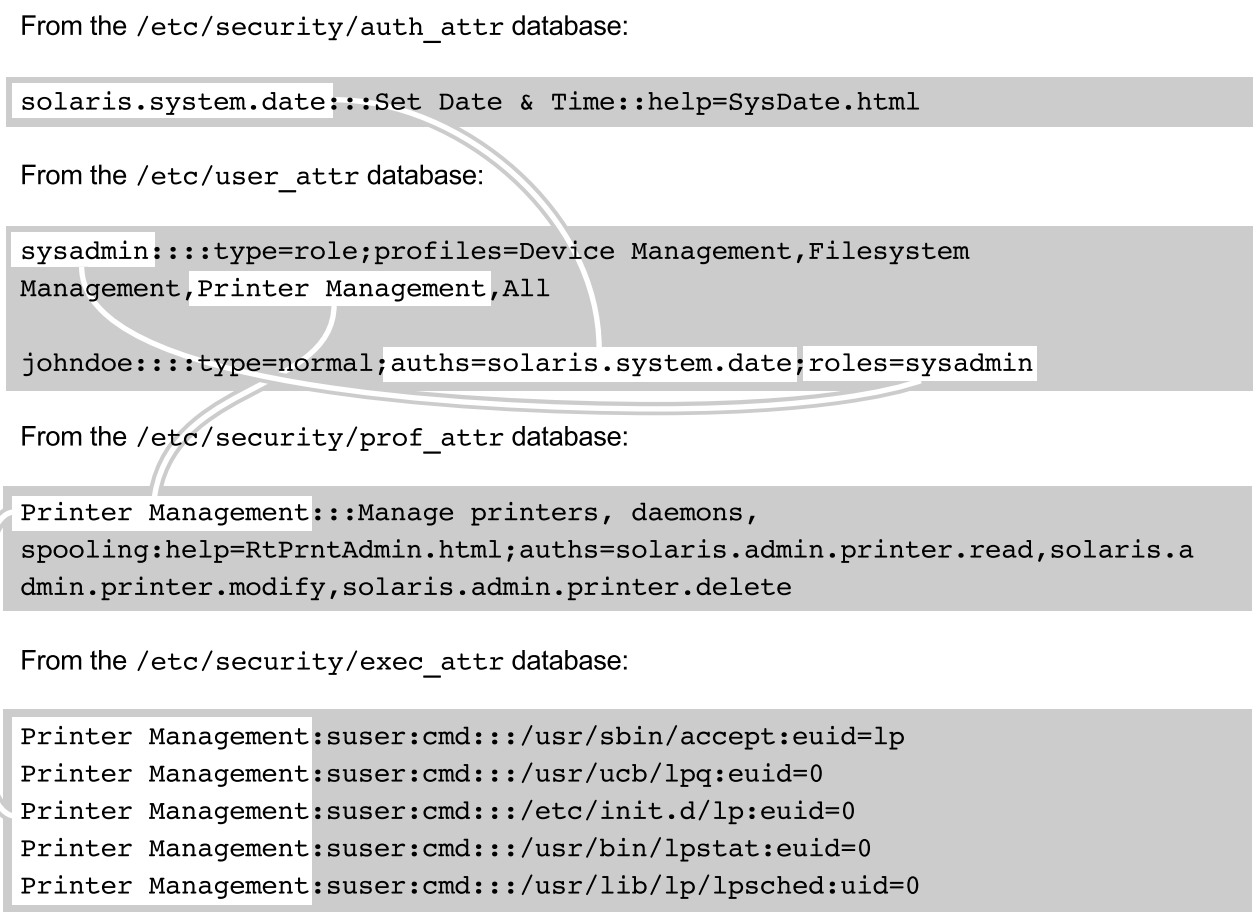


Figure 11-12 Relationship Between the Four RBAC Databases

The /etc/security/policy.conf File

The /etc/security/policy.conf file lets you grant specific rights profiles and authorizations to all users. The two types of entries in the file consist of key-value pairs, as follows:

- AUTHS_GRANTED=*authorizations*, where *authorizations* refers to one or more authorizations
- PROFS_GRANTED=*right_profiles*, where *right_profiles* refers to one or more rights profiles

Some typical values from an /etc/security/policy.conf file are shown in the following example.

```
# cat policy.conf
#
# Copyright (c) 1999-2001 by Sun Microsystems, Inc. All rights reserved.
#
# /etc/security/policy.conf
#
# security policy configuration for user attributes. see policy.conf(4)
#
#ident  "@(#)policy.conf        1.5     01/03/26 SMI"
#
AUTHS_GRANTED=solaris.device.cdrw
PROFS_GRANTED=Basic Solaris User
```

The solaris.device.cdrw authorization provides access to the cdrw command.

```
# grep 'solaris.device.cdrw' /etc/security/auth_attr
solaris.device.cdrw:::CD-R/RW Recording Authorizations::help=DevCDRW.html
```

The `Basic Solaris User` profile grants users access to all listed authorizations. The `profiles=All` field grants unrestricted access to all Solaris OE commands that have not been restricted by a definition in a previously listed authorization.

```
# grep 'Basic Solaris User' /etc/security/prof_attr
Basic Solaris User:::Automatically assigned rights:
auths=solaris.profmgr.read,solaris.jobs.users,solaris.mail.mailq,
solaris.admin.usermgr.read,solaris.admin.logsvc.read,
solaris.admin.fsmgr.read,solaris.admin.serialmgr.read,
solaris.admin.diskmgr.read,solaris.admin.procmgr.user,
solaris.compsys.read,solaris.admin.printer.read,
solaris.admin.prodreg.read,solaris.admin.dcmgr.read,
solaris.snmp.read,solaris.project.read,solaris.admin.patchmgr.read,
solaris.network.hosts.read,solaris.admin.volmgr.read;profiles=All;
help=RtDefault.html
```

Managing RBAC

You can configure RBAC features using the Solaris Management Console or the command line.

Managing RBAC Using the Solaris Management Console

The Solaris Management Console 2.1 in the Solaris 9 OE enables you to configure RBAC features using a GUI console. The GUI provides a point-and-click method of configuring RBAC rights and roles. The GUI wizards prompt you for any necessary configuration parameters.

Note – Using the GUI assumes knowledge of the underlying dependencies that are built into the RBAC feature.

Fundamentals of Managing RBAC

To set up privileged access using the RBAC GUI, follow these steps:

1. Build the user accounts that will be assigned the RBAC rights.

Note – Step 1 is not required if the designated rights and roles are being made available to existing users.

2. Build the rights profiles needed to support the superuser access requirements.
3. Build the role that will provide access to the rights profiles for designated users.

The following example grants an ordinary user access to administrative rights for package commands that require superuser access:

Figure 11-13 shows that access to the RBAC features begins with the Solaris Management Console.

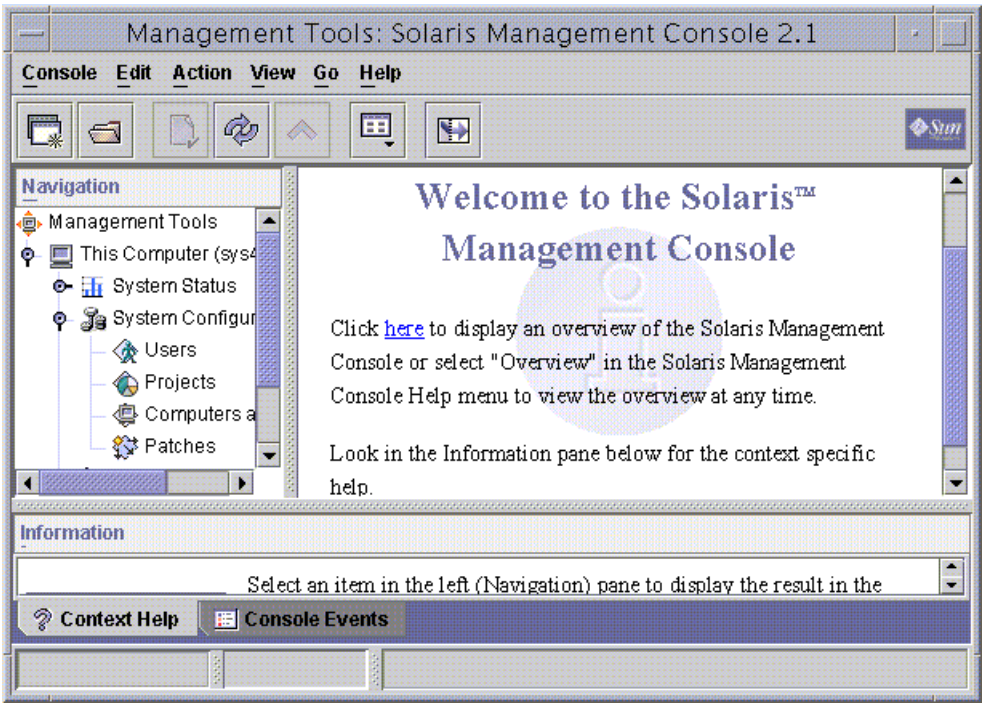


Figure 11-13 Solaris Management Console 2.1 – Users Window

To access RBAC features, perform the following steps:

1. Select Management Tools.
2. Click This Computer.
3. Click System Configuration
4. Double-click the Users icon.

5. Log in as `root`, as shown in the Log In: User Name Window in Figure 11-14.

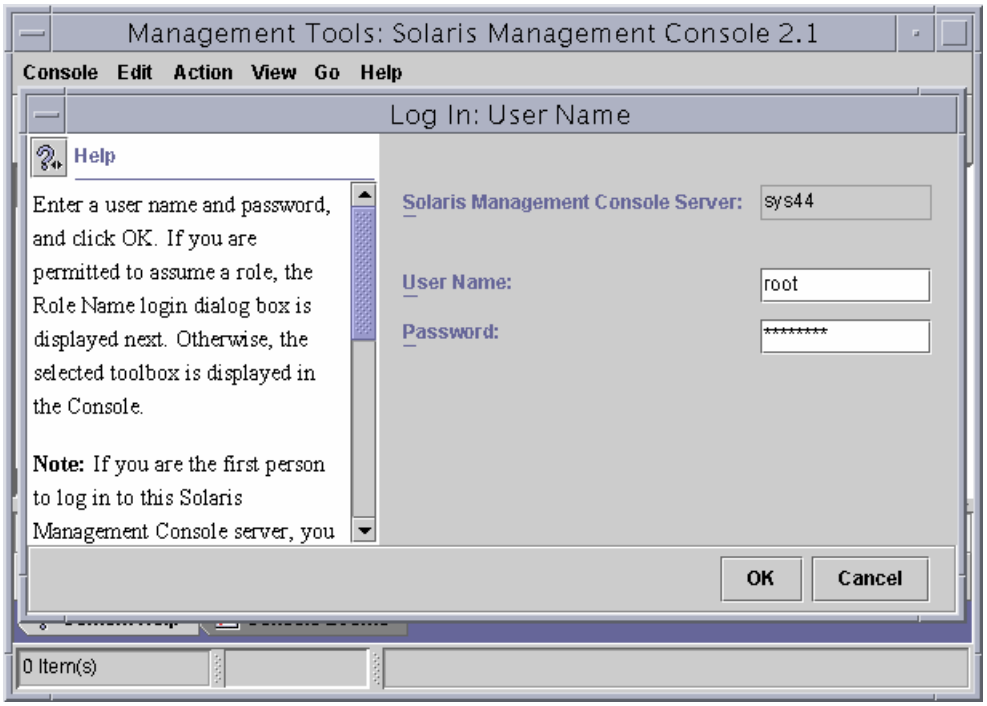


Figure 11-14 Log In: User Name Window

From this login, you have the necessary permissions to set up users, work with name services, and assign rights and roles to other users.

Note – After other users have been granted the necessary access permissions, you can log in with those user login names on subsequent sessions.

After you log in, the View pane displays the set of tools used to perform traditional user administration tasks and the RBAC tasks, as shown in Figure 11-15.

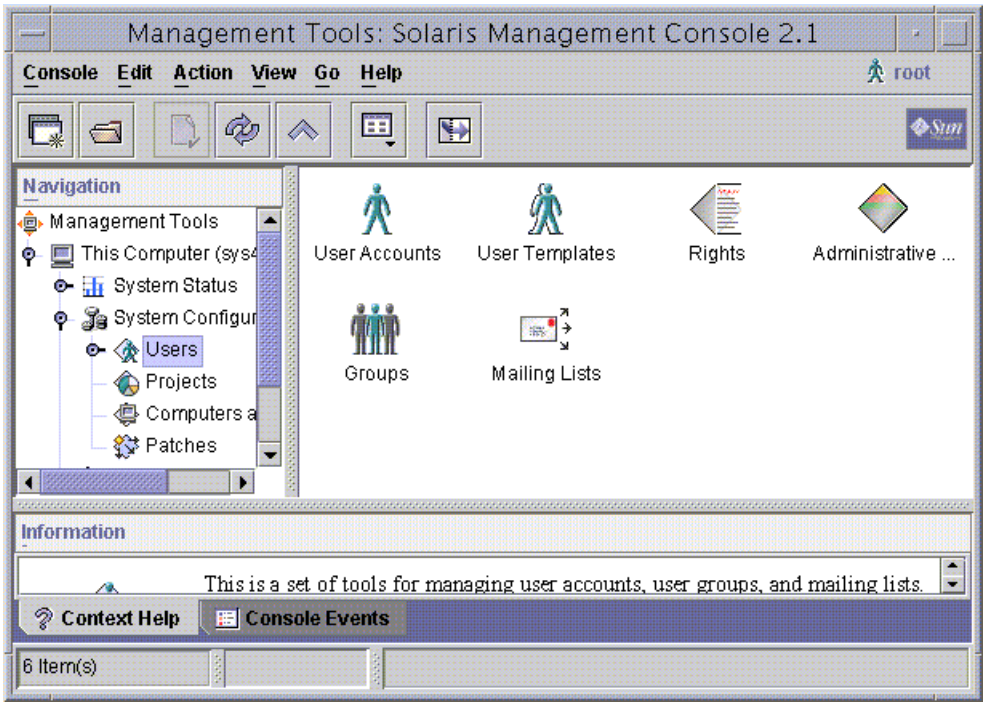


Figure 11-15 Solaris Management Console 2.1 – Users Tools Window

Table 11-4 defines the tools in the Users toolbox.

Table 11-4 Users Tools

Title	Description
User Accounts	Add (or modify) user accounts in several ways: individually, in multiples, or starting from a template.
User Templates	Create a template. If you need to create multiple users with similar attributes, you can first create a template for that type of user.
Rights	Configure a named collection that includes three components: commands, authorizations, and other previously created rights.

Table 11-4 Users Tools (Continued)

Title	Description
Administrative Roles	Configure a user account with a specific set of administrative rights. You must use the su command to access a role, because you cannot log in to a role.
Groups	Manage access to groups.
Mailing Lists	Add a new mailing list. You can also use this tool to view, add, or delete recipients in a mailing list.

6. Double-click the User Accounts icon to select the User Accounts functions.

   The existing users appear in the View pane, as shown in Figure 11-16.

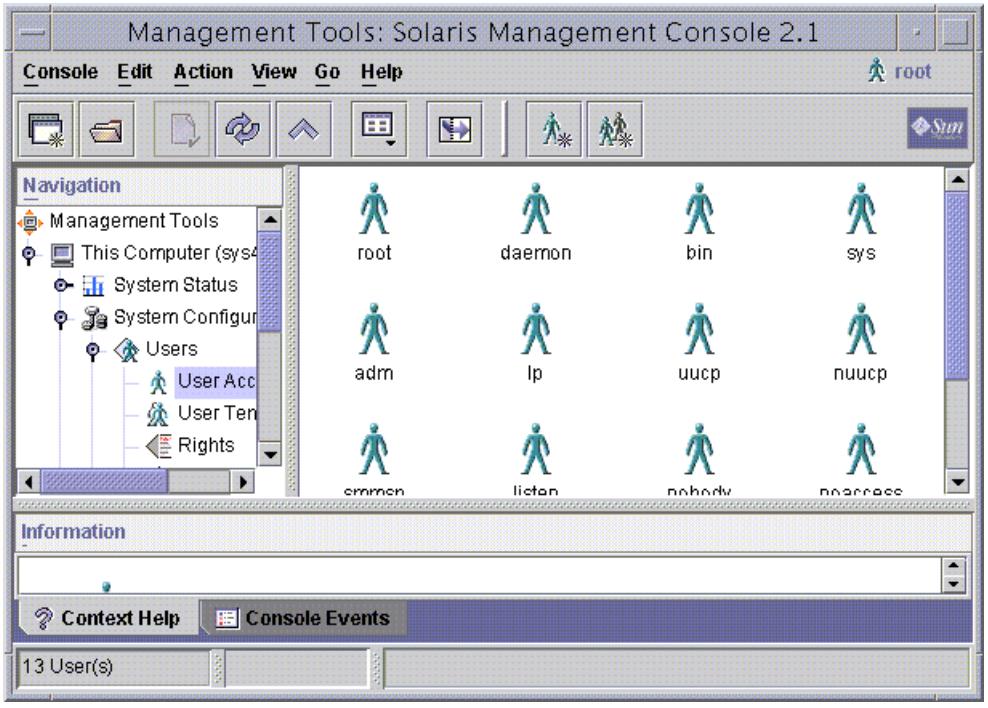


Figure 11-16 Solaris Management Console 2.1 – User Accounts Window

Building User Accounts

You can build a new user account that will be assigned access to all the package administration commands. Perform the following steps:

1. Select Add User from the Action menu, as shown in Figure 11-17.

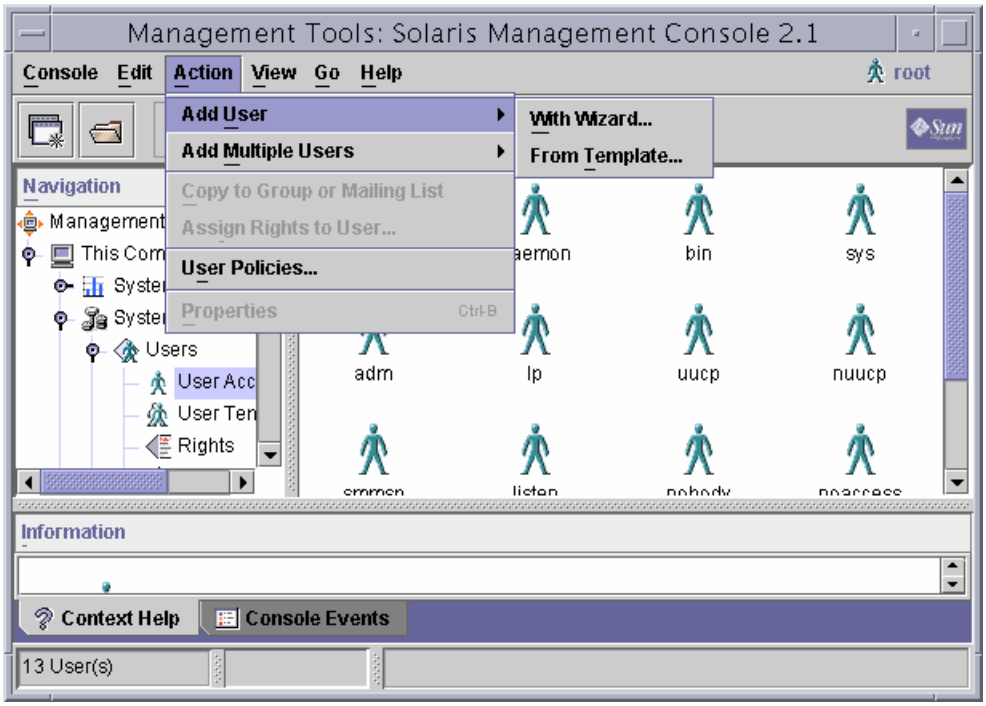


Figure 11-17 Action Menu – Add User

2. Select With Wizard from the Add User submenu.

Note – The Add User Wizard works the same as the `useradd` command and earlier GUI tools, such as AdminTool.

The Add User Wizard – Step 1 window appears, as shown in Figure 11-18.

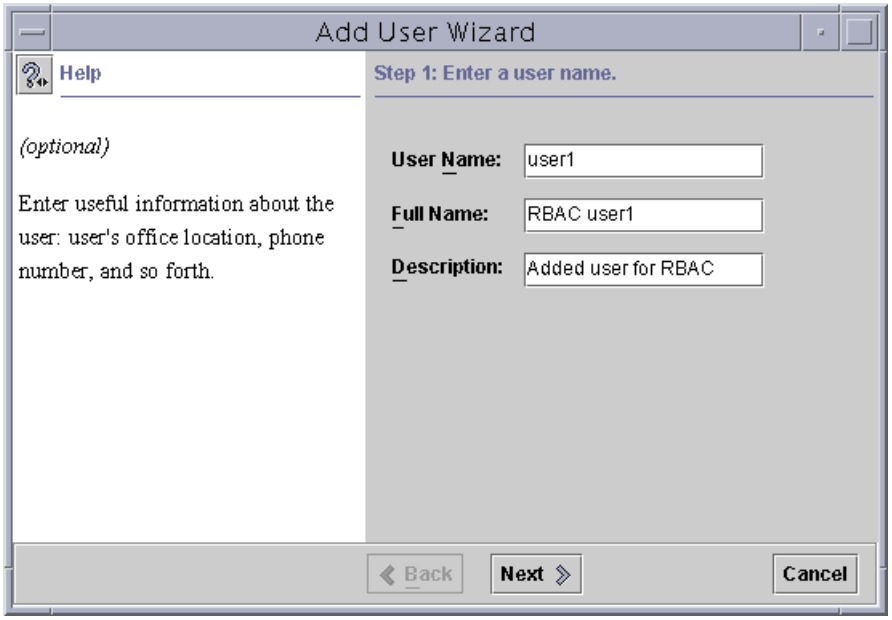


Figure 11-18 Add User Wizard – Step 1 Window

3. Enter the following information:

User Name	The login name for this user account. Enter `user1` as the user name.
Full Name	A descriptive entry identifying the owner of this account. Enter `RBAC user1` as the full name.
Description	Similar to the full name, this field further identifies the owner of this account. This entry populates the `gecos` field in the `/etc/passwd` file. Enter `Added user for RBAC` as the description.

4. Click Next to continue.

   The user ID number is the user's unique numerical ID for the system. The displayed number is the next available UID for the system. If this user account is accessible across multiple standalone systems, the UID should remain consistent to avoid file ownership problems between those systems.

5. Accept the default user ID number, as shown in the Add User Wizard – Step 2 window in Figure 11-19.

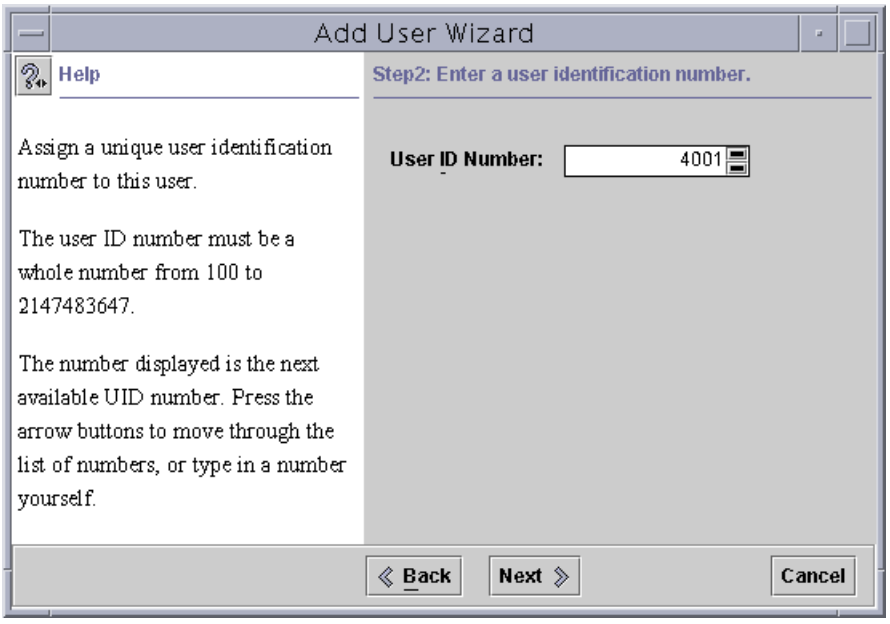


Figure 11-19 Add User Wizard – Step 2 Window

6. Click Next to continue.

There are two password options in the Add User Wizard – Step 3 window, as shown in Figure 11-20. With the first option, the new user will be prompted to set the password when logging in for the first time. Alternatively, with the second option, you can immediately assign the account password.

Figure 11-20 Add User Wizard – Step 3 Window

7. Enter and confirm `123pass` as the password, as shown in Figure 11-20.
8. Click Next to continue.

Group membership allows this user to share access permissions with other users within the same group, as shown in the Add User Wizard – Step 4 window in Figure 11-21. You can add this user to additional groups' common characteristics after account creation. Each user can belong to 15 additional groups that are also known as secondary groups.

9. When prompted with a choice for the new user's primary group membership, accept the default group assignment, as shown in Figure 11-21.

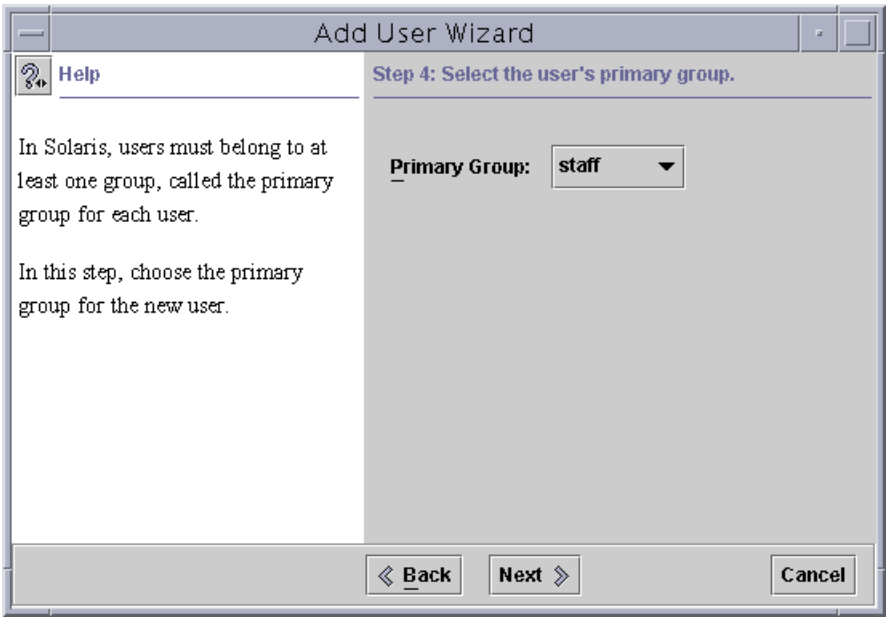


Figure 11-21 Add User Wizard – Step 4 Window

10. Click Next to continue.

The home directory path defines where this user's personal files are stored, as shown in the Add User Wizard – Step 5 window in Figure 11-22. When the account is created, the new user name appends to the home directory path that is defined in this field. For example, if this user is named `user1`, then the home directory becomes `/export/home/user1`.

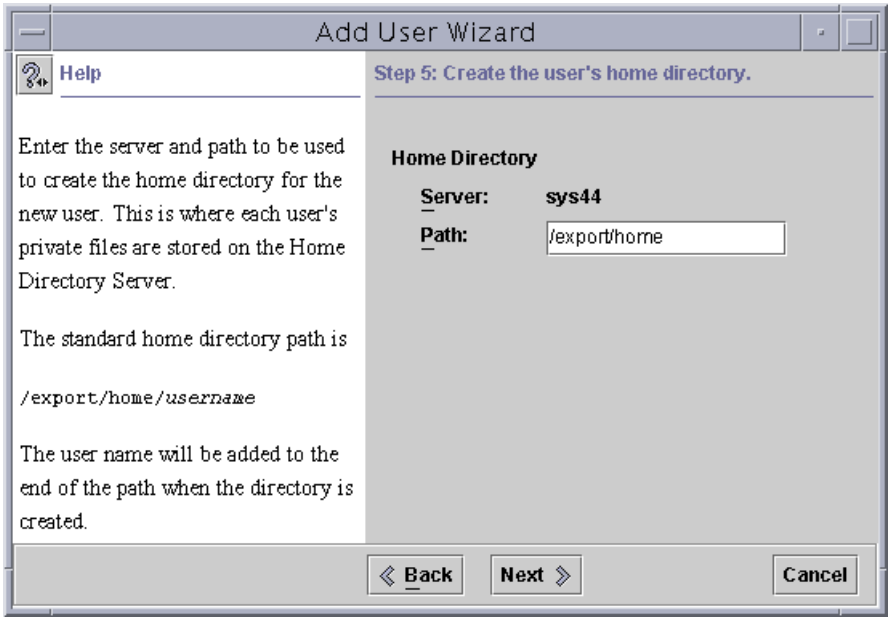


Figure 11-22 Add User Wizard – Step 5 Window

11. Enter the name of the directory in which the user's home directory will be created (`/export/home`), as shown in Figure 11-22.
12. Click Next to continue.

When you create a new user account, it is customary to also create a mail account, as shown in the Add User Wizard – Step 6 window in Figure 11-23. You provide the user with a mailbox that is a file on the mail server (also known as the inbox) that holds all newly received mail.

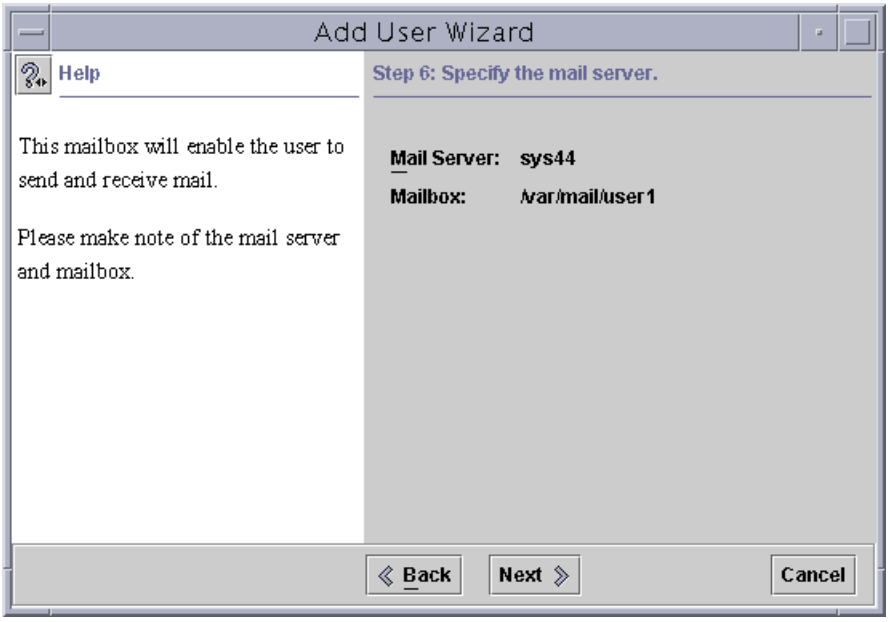


Figure 11-23 Add User Wizard – Step 6 Window

13. Click Next to accept the defaults, as shown in Figure 11-23.

14. Check each field for inadvertent errors, as shown in the Add User Wizard – Step 7 window in Figure 11-24. If you see any errors, step back through the windows to correct them, and then step forward again to the confirmation window.

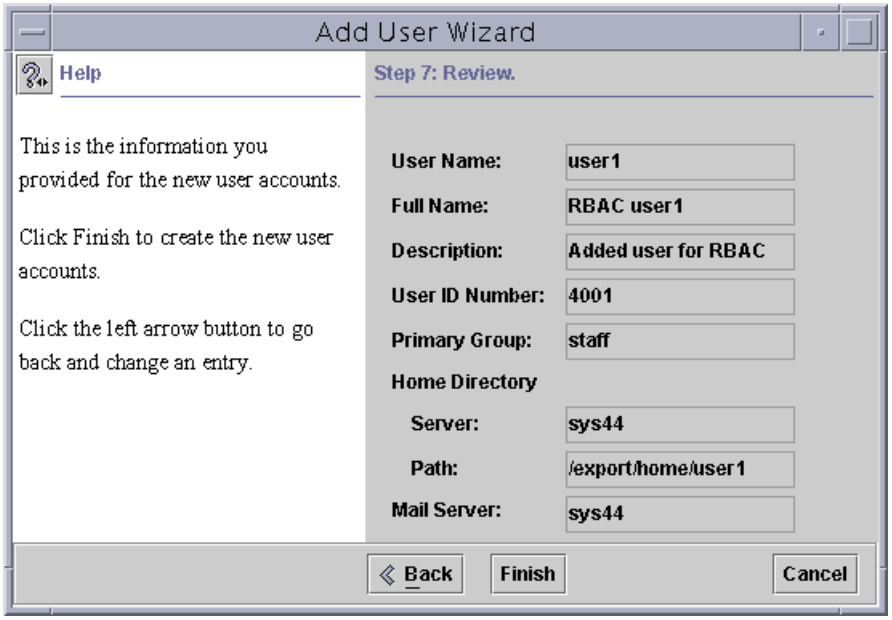


Figure 11-24 Add User Wizard – Review Window

15. When you are satisfied with the field inputs, click Finish to complete building the new user account.

After the new account is created, you are returned to the Solaris Management Console Window, which displays the new account, as shown in Figure 11-25.

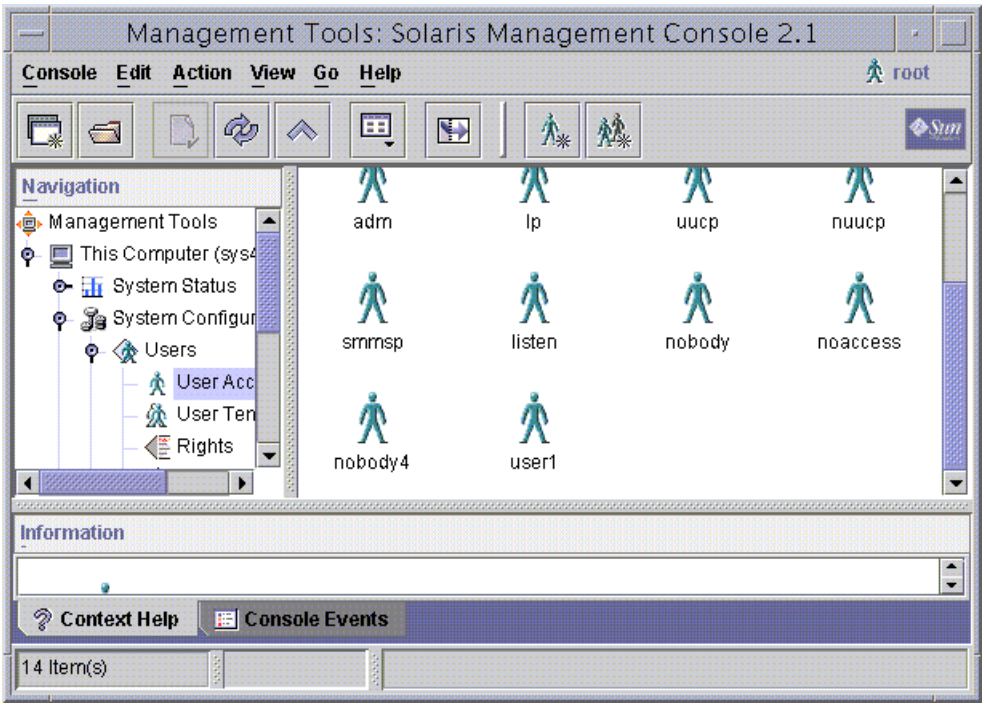


Figure 11-25 Solaris Management Console 2.1 – User Accounts Window

To test the user account, perform the following steps:

1. Log in with the user name that was just created.

Note – The host name in this example is `sys44`, and the user name is `user1`.

```
# telnet sys44
Trying 127.0.0.1...
Connected to sys44.
Escape character is '^]'.


SunOS 5.9

login: user1
Password:
Sun Microsystems Inc.   SunOS 5.9       Generic May 2002
```

2. Execute a few commands to verify that the new account functions as created.

```
$ who
root       console      Feb 28 13:45    (:0)
root       pts/4        Mar  2 09:29    (:0.0)
user1      pts/5        Mar  6 14:32    (sys44)


$ id
uid=4001(user1) gid=10(staff)
$ ls -a
.          ..         .cshrc    .login    .profile
$
```

3. Now that you have verified that the basic Solaris OE commands are functioning within the new user account, try executing more specialized commands within this account. Use the `pkginfo` (package information) command and the `pkgrm` (package removal) command. These examples use the `SUNWpppg` package.

```
$ pkginfo -l SUNWpppg
   PKGINST:  SUNWpppg
      NAME:  GNU utilities for PPP
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  11.9.0,REV=2002.02.12.18.33
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Optional GNU utilities for use with PPP
    PSTAMP:  crash20020212184313
  INSTDATE:  Feb 28 2002 08:32
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:       12 installed pathnames
                   8 shared pathnames
                   8 directories
                   3 executables
                 146 blocks used (approx)

$ pkgrm SUNWpppg
pkgrm: not found
```

Note – The `pkginfo` command is stored in the `/usr/bin` directory, which is in the default `PATH` variable for regular user accounts. The `pkgrm` is stored in the `/usr/sbin` directory, which is not in the default `PATH` for regular user accounts. You can modify the `PATH` variable to include the command's path, or you can enter the absolute path of the command on the command line.

```
$ /usr/sbin/pkgrm SUNWpppg
pkgrm: ERROR: You must be "root" for pkgrm to execute properly.
$
```

The `user1` account can execute the `pkginfo` command because no special privileges are required to get information on installed packages. However, to remove a software package requires `root` permissions; therefore, you must give `user1` superuser access to the system or give the user access to a restricted role account that has these specific rights. You should first create the specific set of rights, and then create a role to which you can assign the rights.

Building Rights Profiles

The Solaris 9 OE includes many default sets of rights. These rights profiles include the sets of tasks that system administrators are required to perform. In a large enterprise, you might have separate administrators for each of these rights, whereas, in a smaller company, a single administrator could be responsible for one or more of these task categories.

As a primary administrator, you must decide between two scenarios when using profiles:

- The default collections of task sets fit your Information Technology (IT) organization; in which case, you can move directly to creating roles for your users to assume when these task sets are required.
- A task set collection must be defined to further subdivide the default task sets. In this case, you must first create new rights profiles before creating roles.

In the earlier example, `user1` required access permissions to the full set of package administration commands. You can create a rights profile called Package Administration to add to the default rights profiles supplied with the Solaris 9 OE release.

To add or build a rights profile, perform the following steps:

1. Double-click on Rights in the Navigation pane.

   The View pane of the Solaris Management Console displays some of the categories for these collections of system administrator tasks, as shown in the Solaris Management Console 2.1 – Rights window in Figure 11-26.

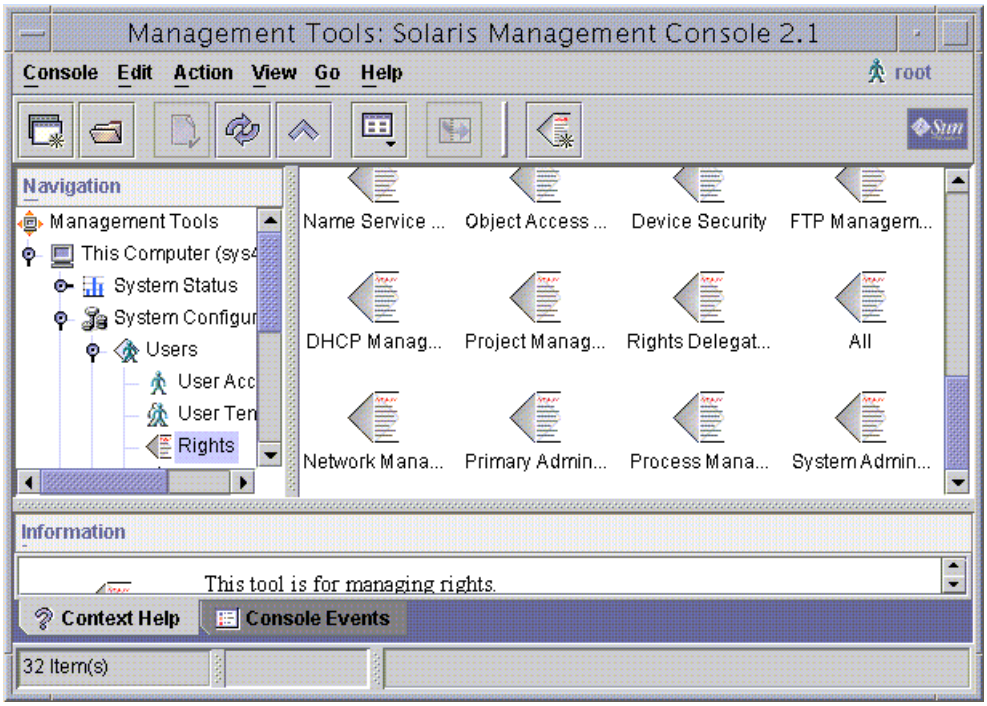


Figure 11-26 Solaris Management Console 2.1 – Rights Window

2. Select Add Right from the Action menu, as shown in Figure 11-27.

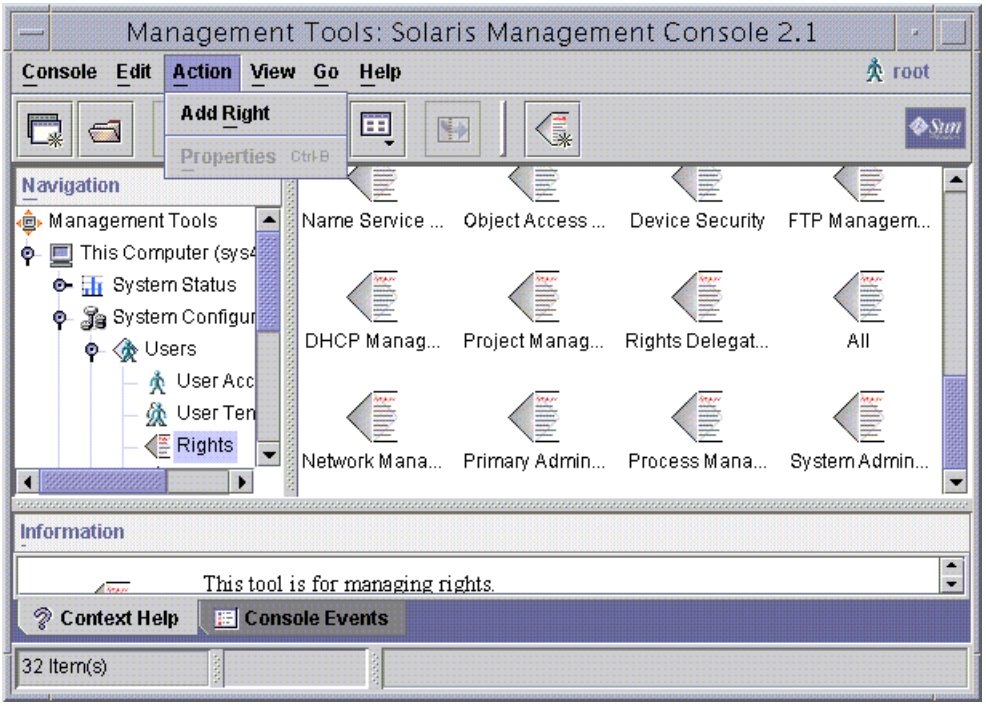


Figure 11-27 Action Menu – Add Right

The Add Right window – General tab appears. As shown in Figure 11-28, the window contains four tabs. Each tab configures one or more aspects of a rights profile.

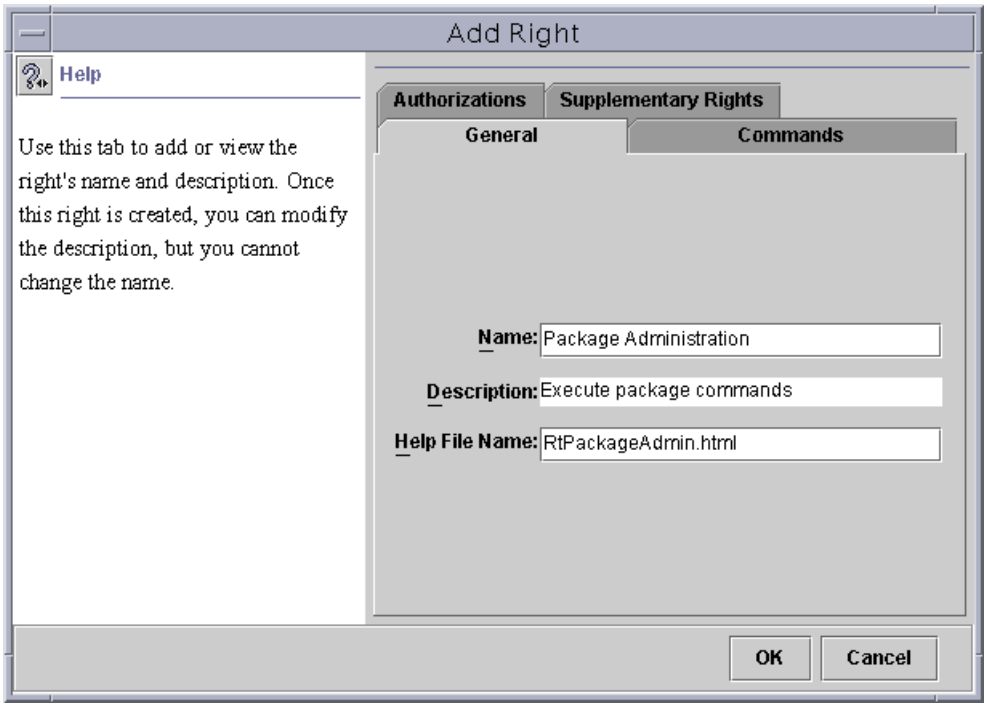


Figure 11-28 Add Right Window – General Tab

3. Select the General tab, and fill in the fields as follows:

Name	The name that identifies the rights profile in the rights window. This name corresponds to the line entry in the `/etc/security/prof_attr` database.
Description	This description is also presented in the `/etc/security/prof_attr` database as a definition of the rights profile.
Help File Name	This is a required field. It points to an HTML file in the `/usr/lib/help/profiles/locale/C` directory. You can copy and edit an existing file to satisfy this requirement.

Note – You should create the help file before referencing the help file in this window.

4. Select the Commands tab, as shown in Figure 11-29, and select the commands that your rights profile will include as follows:

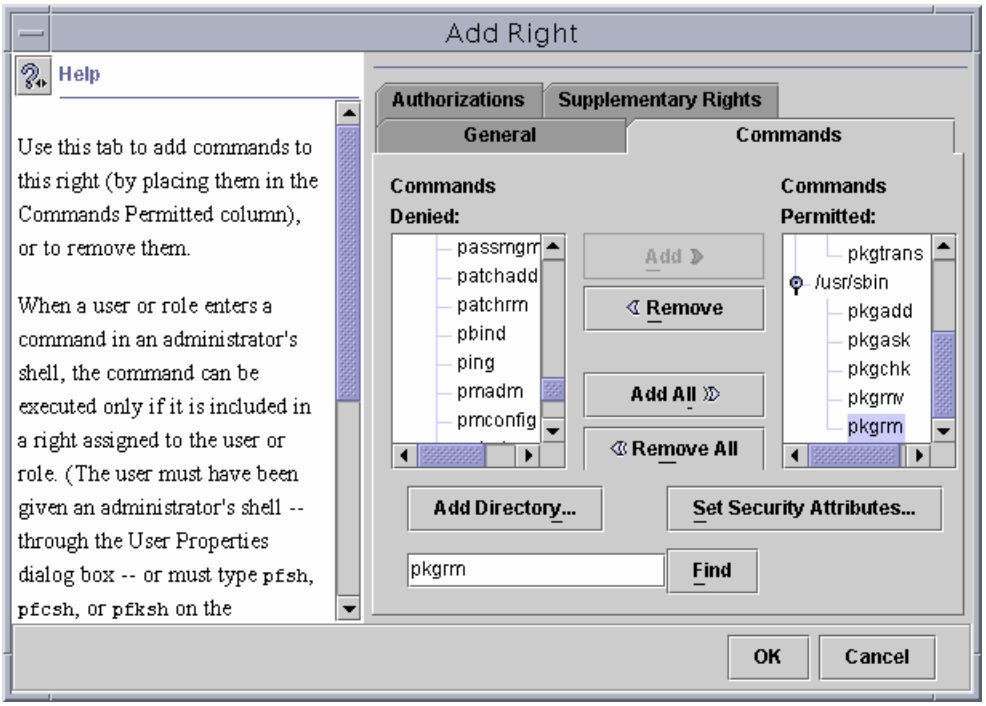


Figure 11-29 Add Right Window – Commands Tab

a. For each command that you want the rights profile to be able to run, select it, and click Add.

   The command moves to the Commands Permitted list.

b. Click Set Security Attributes.

The Set Security Attributes window, as shown in Figure 11-30, appears. This window also appears when you double-click any of the commands in the Permitted Commands field.

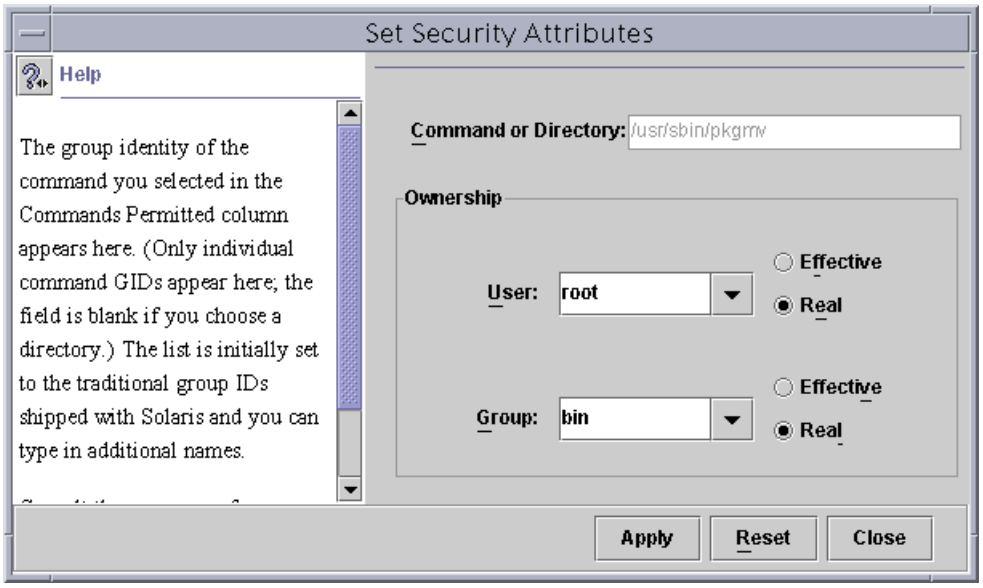


Figure 11-30 Set Security Attributes Window

c. Define the security attributes for each permitted command; you must assign the UID, EUID, GID, and EGID permissions.

Note – The online man pages do not always define the required execution permissions. However, the `/etc/security/exec_attr` database is a good source for the proper execution permissions for most commands.

5. Search the `/etc/security/exec_attr` database for the `pkgrm` command, and set the ownership accordingly.
6. Click Apply.
7. Click Close to continue.

The View pane in the Solaris Management Console is updated to include the Package Administrator rights profile, as shown in Figure 11-31.

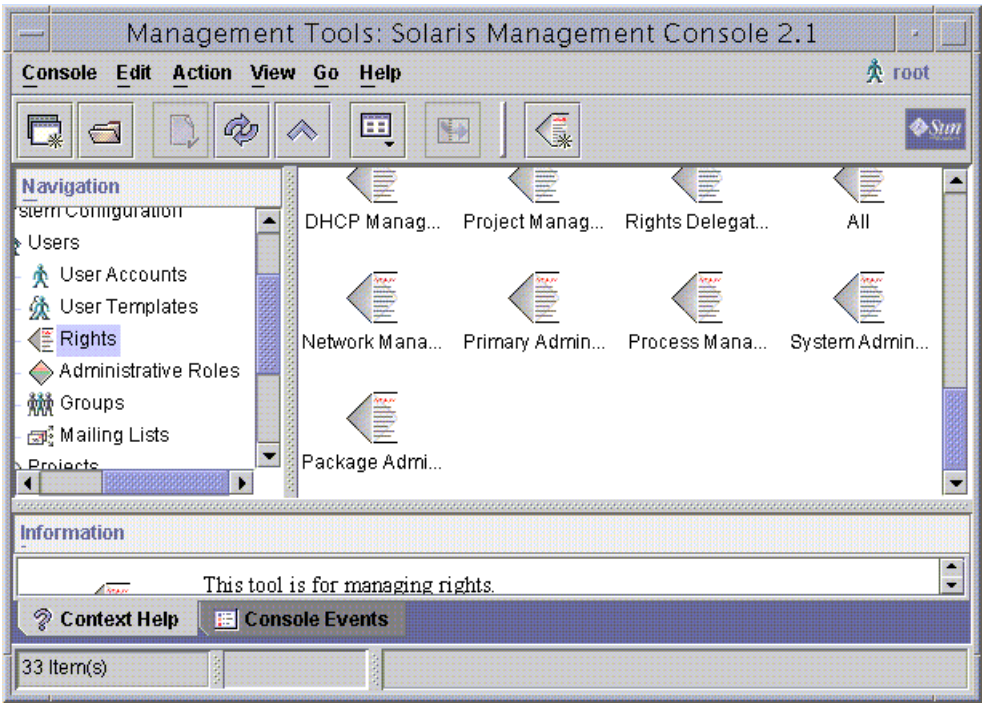


Figure 11-31 Solaris Management Console 2.1 – Rights Window

8. If you need to make modifications to this rights profile, double-click the newly created Package Administrator entry to return to the rights creation windows.

   After the rights profile is completed, it can be assigned to either an existing user or to a role.

Note – A user must be running a profile shell to execute the commands in an assigned rights profile.

Building the Role

Administrative *roles* run administrator shells, also known as profile shells. Because of the profile shell, you cannot log in to a *role* account. You must log in as a regular user, and then assume the role by using the su command.

To build an administration role, complete the following steps:

1. To display existing roles, double-click Administrative Roles in the Navigation pane, as shown in Figure 11-32.

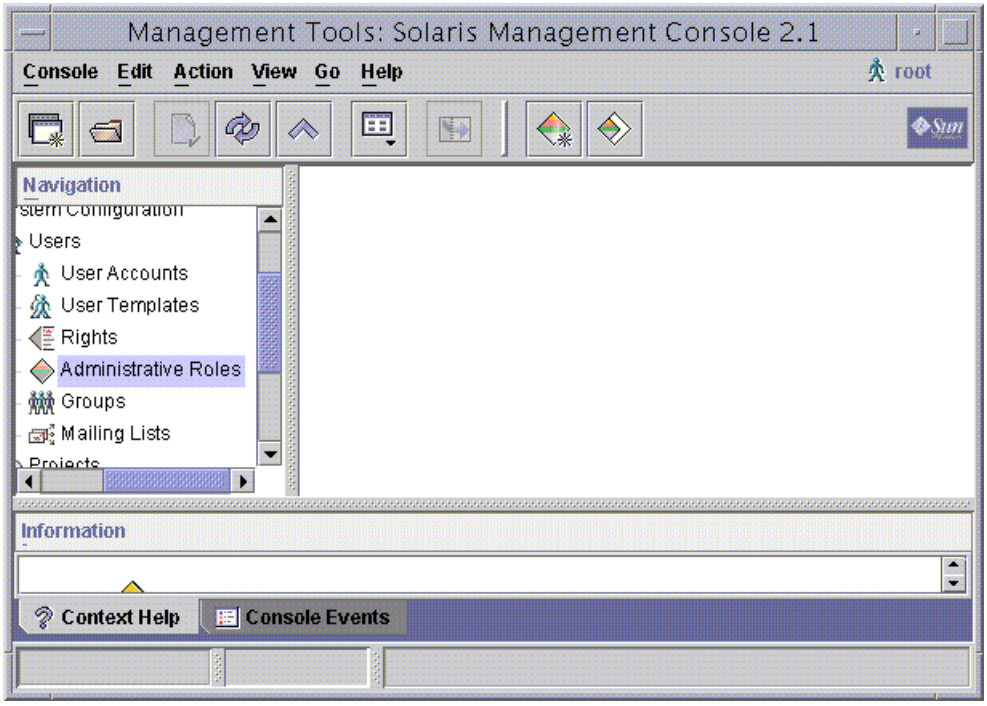


Figure 11-32 Solaris Management Console 2.1 – Administrative Roles Window

Note – By default, the Solaris 9 OE does not have any roles defined.

2. To create a role, select Add Administrative Role from the Action menu, as shown in Figure 11-33.

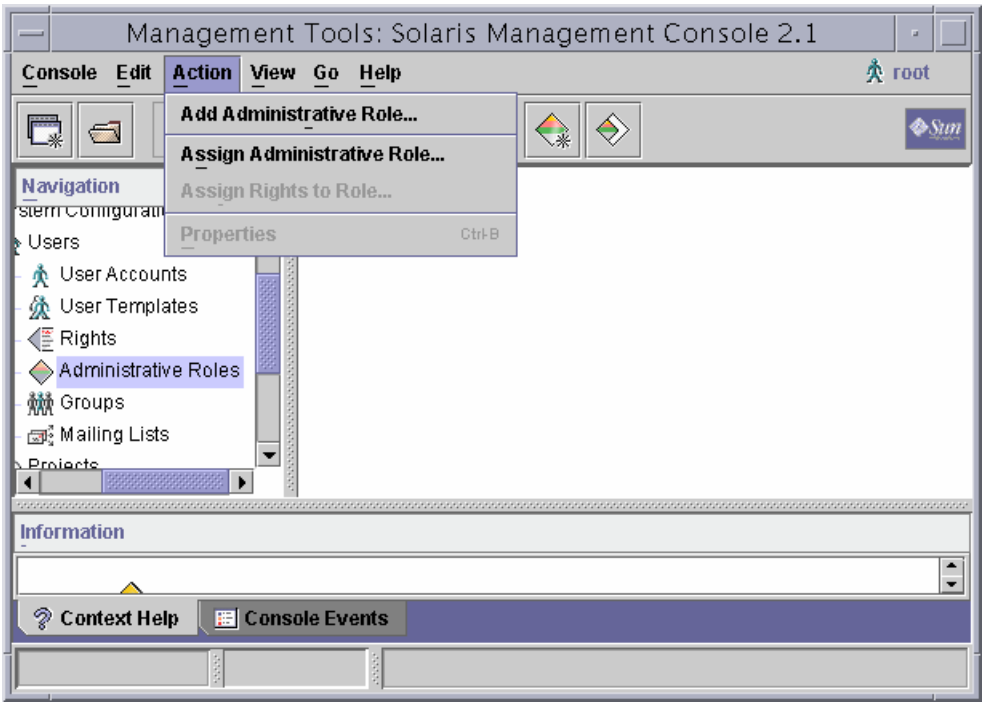


Figure 11-33 Action Menu – Add Administrative Role

The Add Administrative Role – Step 1 window appears, as shown in Figure 11-34.

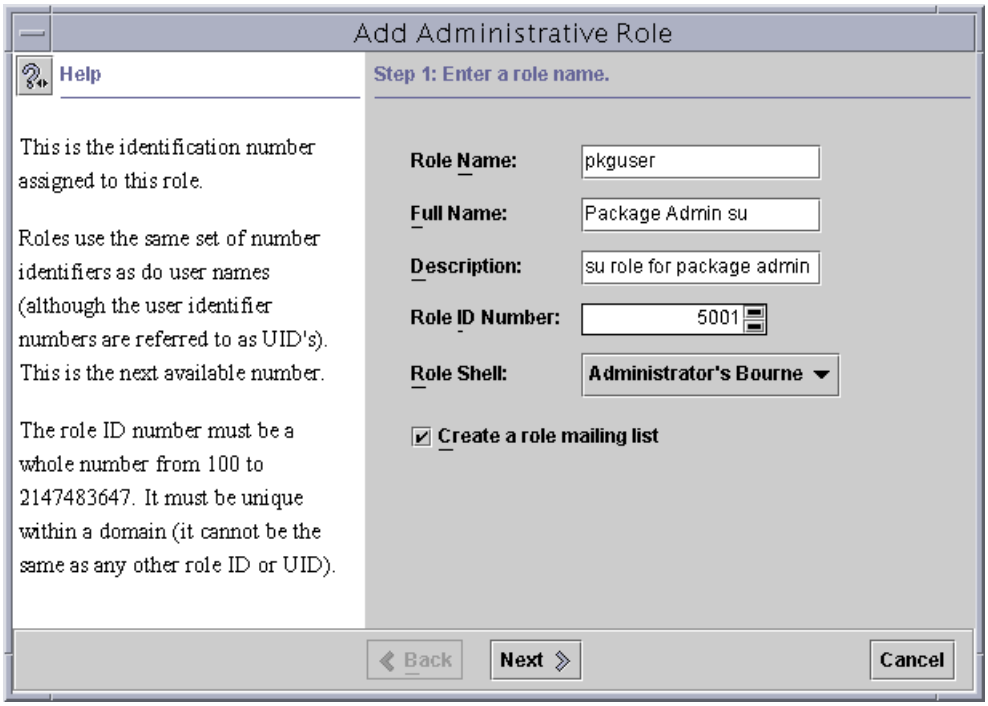


Figure 11-34 Add Administrative Role – Step 1 Window

3. Complete the fields in Figure 11-34 as follows:

Role Name	This is the name that you use to assume a specific role with the `su` command. This name identifies entries in the `/etc/passwd` and `/etc/shadow` files and in the `/etc/user_attr` database.
Full Name	This is an optional entry. If used, make this value unique to this role.
Description	This should clearly state the intent of this role. This entry populates the `gcos` field in the `/etc/passwd` file.
Role ID Number	This number, like the UID in user accounts, numerically identifies the role to the system.
Role Shell	These shells allow the `pfexec` command to execute specified commands with predefined process attributes, such as a specific user or group IDs.

4. Click Next to continue.

   The Add Administrative Role – Step 2 window appears, as shown in Figure 11-35.

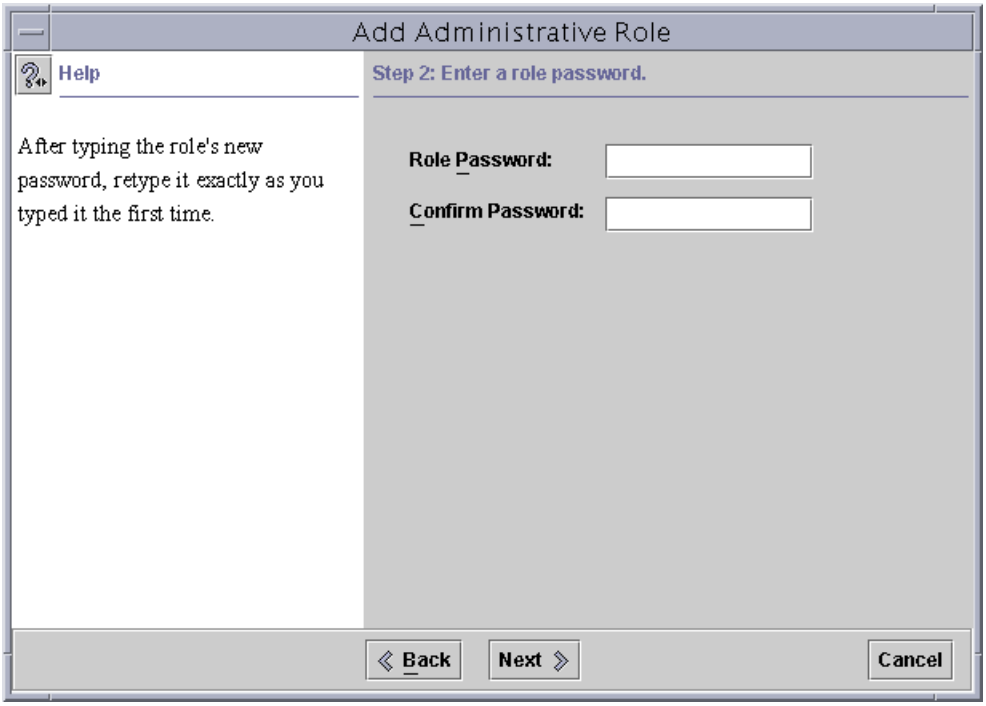


Figure 11-35 Add Administrative Role – Step 2 Window

   The role password follows the same characteristics as a regular user account password. A password must consist of between 6 and 15 characters (case-sensitive letters, numbers, and special characters). Only the first 6 characters are used during authentication, but 15 are available for those users who want longer passwords.

5. Enter and confirm the password.
6. Click Next to continue.

7. To build the administrative rights for this role, click the Package Administrator rights profile in the left column, as shown in the Add Administrative Role – Step 3 window in Figure 11-36.

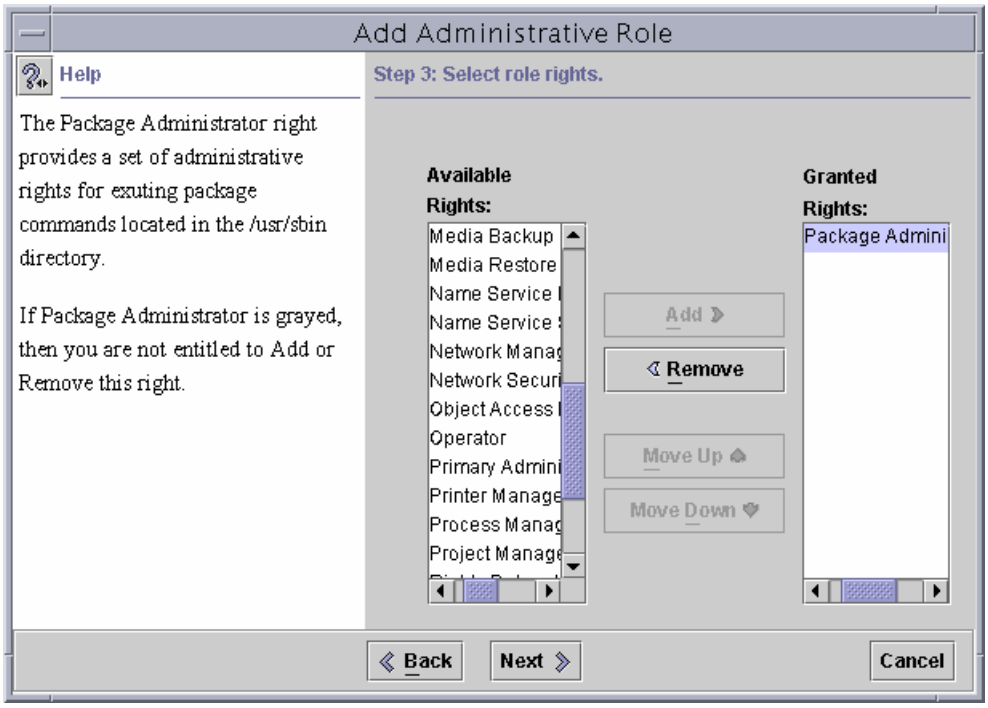


Figure 11-36 Add Administrative Role – Step 3 Window

8. Click Add.

   The rights are added to the Granted Rights in the right column.

Note – The help that is available on this screen is derived from the help files that are indicated in the Right Properties: Package Administration window.

9. Click Next to continue.

The Add Administrative Role – Step 4 window enables you to define the server and directory locations for the administrative role's home directory, as shown in Figure 11-37.

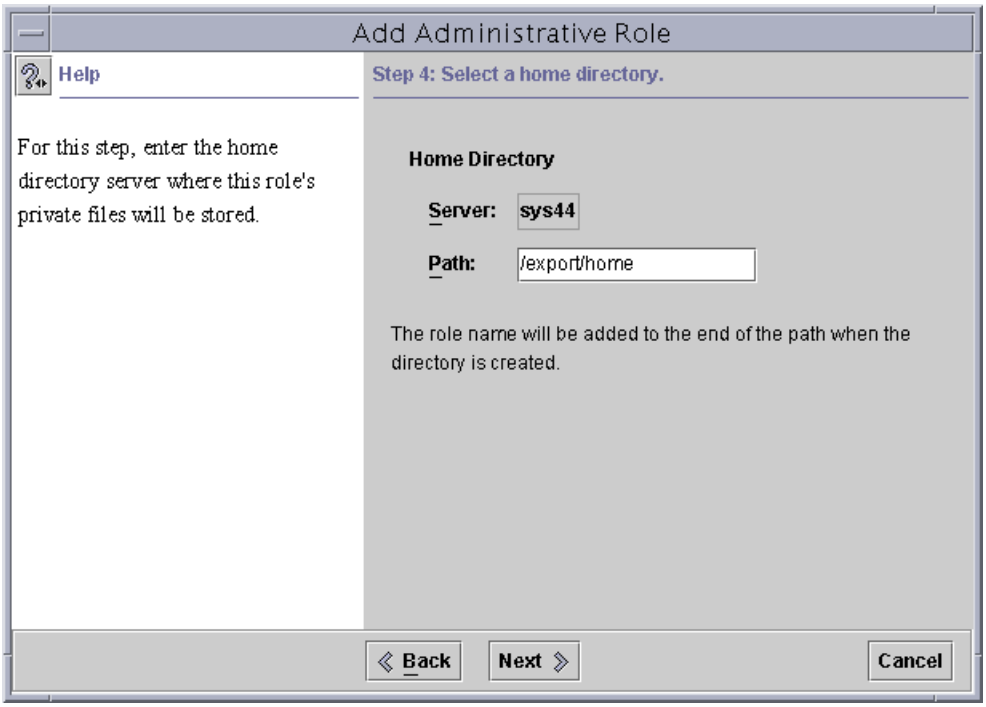


Figure 11-37 Add Administrative Role – Step 4 Window

10. Click Next to accept the default values, which creates a home directory based on the role name.

In Add Administrative Role – Step 5 window, you can provide access for this administrative role to a specific list of users, as shown in Figure 11-38. These are the users that will be allowed to assume this role with the su command.

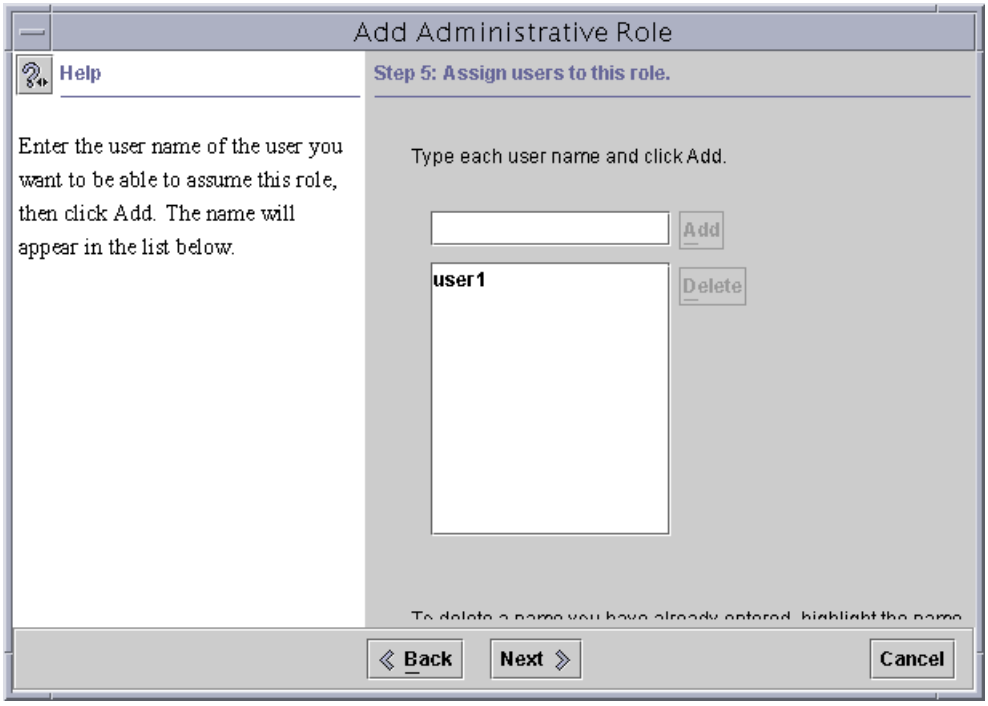


Figure 11-38 Add Administrative Role Window – Assign Users

11. Perform one of the following steps:
    - To add a user, enter a valid user name, and click Add.
    - To delete a user, click on the user's name in the lower box, and click Delete.
12. Click Next to continue.

13. Check each field in the Add Administrative Role – Review window for inadvertent errors. If you discover any errors, step back through the windows to correct them, and then step forward again to this confirmation window, as shown in Figure 11-39.

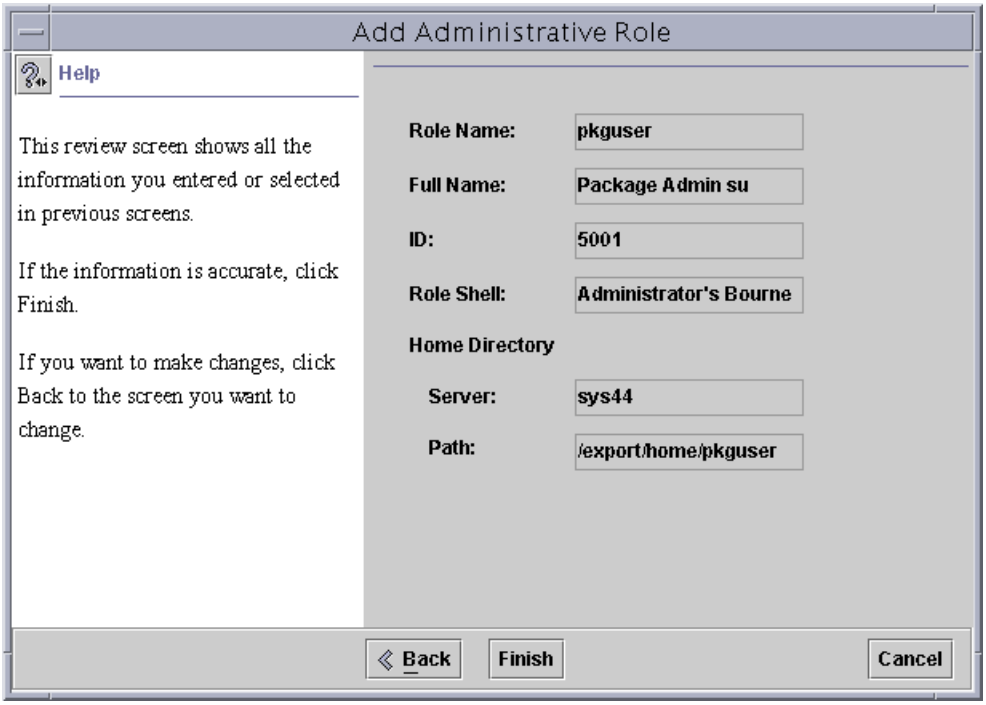


Figure 11-39 Add Administrative Role Window – Review

14. When you are satisfied with the field inputs, click Finish to complete building the new role account.

The new role is listed in the View pane of the Solaris Management Console, as shown in Figure 11-40. Subsequent role modifications can be made by double-clicking the role entry, stepping through the modification windows, and making the appropriate corrections.

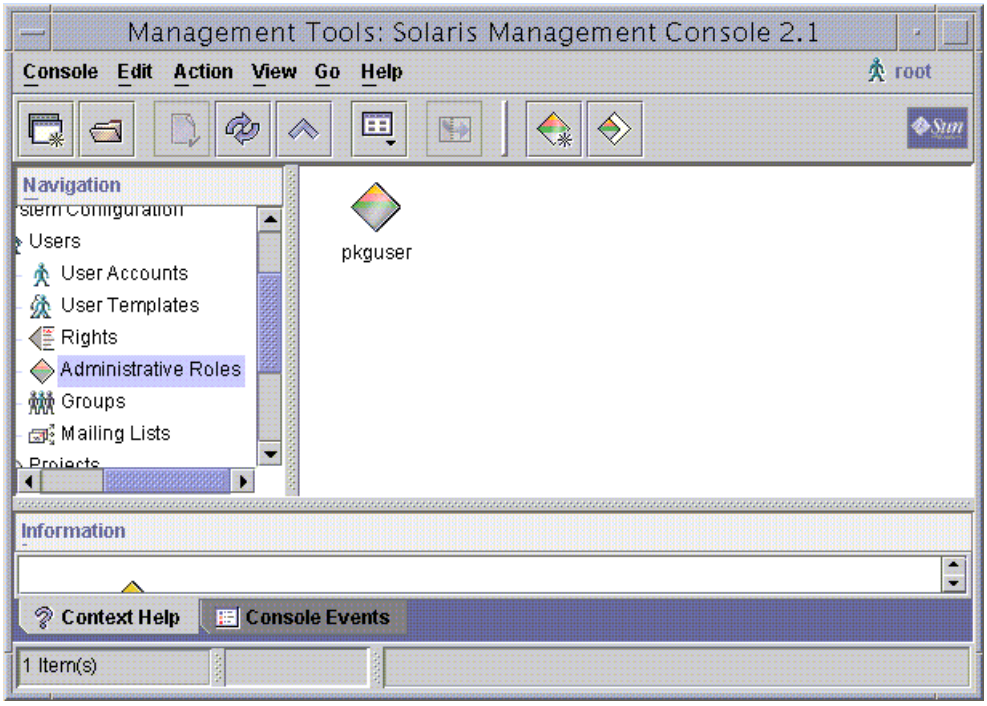


Figure 11-40 Solaris Management Console 2.1 – Administrative Role Window

To test the role, perform the following steps:

1. Log in as `user1`.

```
# telnet sys44
Trying 127.0.0.1...
Connected to sys44.
Escape character is '^]'.


SunOS 5.9

login: user1
Password:
Sun Microsystems Inc.   SunOS 5.9       Generic May 2002
```

2. Execute a few commands to verify the login.

```
$ who
root       console      Feb 28 13:45    (:0)
root       pts/6        Mar  6 14:49    (:0.0)
user1      pts/7        Mar  6 15:47    (sys44)
$ id
uid=4001(user1) gid=10(staff)
$ ls
$ ls -a
.          ..          .cshrc    .login    .profile
```

3. Remove the SUNWpppg package using the pkgrm (package removal) command.

```
$ /usr/sbin/pkgrm SUNWpppg
pkgrm: ERROR: You must be "root" for pkgrm to execute properly.
```

4. To remove a software package requires root permissions. You must give user1 access to the pkguser role account that has these specific rights.

```
$ su - pkguser
Password:
```

5. Verify that you have switched to the role account.

```
$ whoami
pkguser
$ id
uid=5001(pkguser) gid=14(sysadmin)
$ echo $SHELL
/bin/pfsh
$
```

6. Perform the pkgrm command using the pkguser role account.

```
$ /usr/sbin/pkgrm SUNWpppg

The following package is currently installed:
   SUNWpppg        GNU utilities for PPP
                   (sparc) 11.9.0,REV=2002.02.12.18.33

Do you want to remove this package? [y,n,?,q] y

## Removing installed package instance <SUNWpppg>
## Verifying package dependencies.
## Processing package information.
## Removing pathnames in class <none>
/usr/share/man/man1m <shared pathname not removed>
/usr/share/man <shared pathname not removed>
/usr/share <shared pathname not removed>
```

```
/usr/lib/inet/ppp/passprompt.so
/usr/lib/inet/ppp/minconn.so
/usr/lib/inet/ppp <shared pathname not removed>
/usr/lib/inet <shared pathname not removed>
/usr/lib <shared pathname not removed>
/usr/bin/pppdump
/usr/bin <shared pathname not removed>
/usr <shared pathname not removed>
## Updating system information.

Removal of <SUNWpppg> was successful.
$
```

Note – One final test of role account access is to perform a privileged command that the role cannot perform.

7. Execute the `date` command.

```
$ date
Wed Mar  6 15:52:33 MST 2002
```

8. Change the system time using the `date` command.

```
$ date
Wed Mar  6 15:52:33 MST 2002
$ date 03061600
date: Not owner
usage:  date [-u] mmddHHMM[[cc]yy][.SS]
        date [-u] [+format]
        date -a [-]sss[.fff]
```

In summary, you built a regular user account named `user1`. This account has access to perform regular user commands. However, when it is necessary to perform a software package removal that requires `root` access, `user1` must switch to a role that is configured with the required execution profile.

In the role of `pkguser` and using the Package Administrator rights profile, `user1` acquires the rights to remove a software package. However, it is the `pkguser` role that has the rights to remove the software package.

This pkguser role is not configured with full superuser access. Therefore, when you attempt to change the system date using this role, you are unsuccessful. The inability to access all superuser commands demonstrates the advantage of using RBAC instead of granting this access through the superuser. You can configure each administrator to perform only those tasks required in their job, and access to other tasks can remain secure.

Managing RBAC Using the Command Line

Command-line tools for configuring RBAC are also available. The dependencies that demand strict adherence to a defined configuration sequence when using the GUI are not present when using the command line. However, for RBAC to function as designed, the dependencies between the files and databases are still present and must be adhered to. The tools include:

- `roleadd` – Adds a role account on the system.
- `rolemod` – Modifies a role's login information.
- `useradd` – Adds a user account on the system.

Adding a Role

The `roleadd` command adds a role entry to the `/etc/passwd`, `/etc/shadow`, and `/etc/user_attr` files. Some common options include:

`-c` *`comment`*	A text string that provides a short description of the role.
`-d` *`dir`*	Specifies the home directory of the new role.
`-m`	Creates the new role's home directory if it does not already exist.
`-A` *`authorization`* and `-P` *`profile`*	Assigns authorizations and profiles, respectively, to the role.

```
# roleadd -m -d /export/home/tarback -c "Privileged tar Backup Role" \
-P "Media Backup,Media Restore" tarback
```

In this example, the `roleadd` command creates a new role called `tarback`, builds the required directory structures (including the home directory), and assigns the role with a profile of `Media Backup` and `Media Restore`.

Note – This command does not work on its own; the profile is not yet defined. You must edit the `/etc/security/exec_attr` database and the `/etc/security/prof_attr` database to include the `Media Backup` and `Media Restore` profiles.

Modifying a Role

To modify the login information of a role on a system, use the `rolemod` command. The `rolemod` command changes the definition of the specified role and makes the appropriate login-related changes to the system file and file system. The fields in the `rolemod` command are:

`-A` *`authorization`*	Specifies one or more comma-separated authorizations, as defined in the `/etc/security/auth_attr` database.
`-e` *`expire`*	Specifies the expiration date for a role.
`-l` *`new_logname`*	Specifies the new login name for the role.
`-P` *`profile`*	Specifies one or more comma-separated authorizations, as defined in the `/etc/security/prof_attr` database.
`-s` *`shell`*	Specifies the full path name of the program that is used as the role's shell when logging in.

This example modifies authorizations and profiles, respectively to the role.

```
# rolemod -A auth1,auth2 -P profile1,profile2 role1
```

In this example, the `rolemod` command assigns authorizations *`auth1`* and *`auth2`* and profiles *`profile1`* and *`profile2`* to the role named `role1`. The named authorizations must be previously defined in the `/etc/security/auth_attr` database, and the named profiles must be previously defined in the `/etc/security/prof_attr` database.

Adding a User

The useradd command adds a new user with authorizations and profiles to the /etc/passwd, /etc/shadow, and /etc/user_attr files. Some options of the useradd command are:

-c *comment*	Contains a text string for the user's full name and is stored in the user's /etc/passwd entry.
-d *dir*	Specifies the home directory of the new user.
-m	Creates the new user's home directory if it does not already exist.
-s *shell*	Specifies the full path name of the program used as the user's shell on login.
-R *role*	Specifies one or more comma-separated execution profiles defined in the user_attr database.
-A *authorization*	Specifies one or more comma-separated authorizations, as defined in the /etc/security/auth_attr database.
-P *profile*	Specifies one or more comma-separated authorizations, as defined in the /etc/security/prof_attr database.

```
# useradd -m -d /export/home/usera -c "User Account usera" \
-s /usr/bin/ksh -R tarback usera
```

In this example, the useradd command creates a new user account for a user named usera. This user has access to the role tarback, and the user's shell has been changed from the default Bourne shell to the Korn shell.

Additional Commands Used to Perform RBAC Functions

Table 11-5 describes some additional commands that you can use with RBAC operations.

Table 11-5 RBAC Commands

Command	Description
`auths`	Displays authorizations for a user.
`makedbm`	Makes a `dbm` file.
`nscd`	Identifies the name service cache daemon, which is useful for caching the `user_attr`, `prof_attr`, and `exec_attr` databases.
`pam_roles`	Identifies the role account management module for the Password Authentication Module (PAM). Checks for authorization to assume a role.
`pfexec`	Identifies the profile shells used to execute commands with the attributes specified in the `exec_attr` database.
`policy.conf`	Identifies the configuration file for the security policy. Lists granted authorizations.
`profiles`	Displays profiles for a specified user.
`roles`	Displays roles granted to a user.
`roleadd`	Adds a role account to the system.
`roledel`	Deletes a role's account from the system.
`rolemod`	Modifies a role's account information in the system.
`useradd`	Adds a user account to the system. Use the `-R` option to assign a role to a user's account.
`userdel`	Deletes a user's login from the system.
`usermod`	Modifies a user's account information in the system.

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Configuring RBAC (Level 1)

In this exercise, you configure RBAC by using the command line in the first task and by using the Solaris Management Console in the second task.

Preparation

During the lab, you are directed to execute commands that do not work to demonstrate how the RBAC facility must be used by logged in users.

Discuss how to use the `auths`, `profiles`, and `roles` RBAC commands to determine user privileges.

Task Summary

Perform the following tasks:

- Using the command-line tools, create a role that can shut down the system, and create a user named `user9`. Assign the role to `user9` to enable `user9` to shut down the system.
- Using the Solaris Management Console, create a user named `user11`, and create a role called `tarback` that can back up the `/etc/shadow` file; make the `tarback` role accessible to `user11`.

If you have any problems that you cannot fix, see your instructor.

Exercise: Configuring RBAC (Level 2)

In this exercise, you configure RBAC by using the command line in the first task and by using the Solaris Management Console in the second task.

Preparation

During the lab, you are directed to execute commands that do not work to demonstrate how the RBAC facility must be used by logged in users.

Discuss how to use the `auths`, `profiles`, and `roles` RBAC commands to determine user privileges.

Task Summary

Perform the following tasks:

- Using the command-line tools, create a role that can shut down the system, and create a user named `user9`. Assign the role to `user9` to enable `user9` to shut down the system.
- Using the Solaris Management Console, create a user named `user11`, and create a role called `tarback` that can back up the `/etc/shadow` file; make the `tarback` role accessible to `user11`.

If you have any problems that you cannot fix, see your instructor.

Tasks

Perform the following tasks.

Task 1– Creating a User and a Role Using the Command-Line Tools

Complete the following steps:

1. Create a role named `sdown`. Give it a user ID of `5000` and a group ID of `10`.
2. Create the profile named `Shut`.
3. Add the profile to the role.
4. Verify that the role is included in the `/etc/user_attr` file.
5. Create a user named `user9` and assign it access to the `sdown` role. Give this user a user ID of `4009` and a group ID of `10`.
6. Check the roles attributes for `user9`.

   _____

7. Assign the `shutdown` command to the profile.
8. To test the configuration, log in as `user9`.
9. From this login, shut down the system.

   What is the result of this shutdown attempt? Why?

   _____

10. Execute the `profiles` command to determine which RBAC profiles are associated with `user9`.
11. Execute the `roles` command to determine which RBAC roles are associated with `user9`.
12. Assume the role `sdown`.
13. Shut down the system by using the `init` command.

    What is the result of this shutdown attempt? Why?

    _____

    _____

14. List the commands that the `sdown` profile can execute.

    _____

    _____

15. Shut down the system using the `shutdown` command.
    What is the result of this shutdown attempt? Why?

    _____

Task 2 – Creating a User and a Role Using the Solaris Management Console

Complete the following steps:

1. Create a new user account with the following specifications:
   - Name: `user11`
   - User ID number: next available
   - Password: Set it now to `user11`
   - Group ID number: use the default
   - Home directory: `/export/home/user11`
   - Mailbox: `/var/mail/user11` (on this system)
2. Confirm user attributes by double-clicking the `user11` entry and stepping through the attribute windows.
3. From the command line, check for user creation.

   Why does `user11` appear in the `/etc/passwd` file, but not in the `/etc/user_attr` database?

   _____
4. Create an administrative role named `tarback` with the following specifications:
   - Name: `tarback`
   - Role ID number: Next available
   - Role shell: Any of the administrator shells
   - Password: `abc123`
   - Rights: As appropriate

Note – A backup administrator must perform all backups of the media as well as any necessary restores.

   - Home directory: `/export/home/tarback`
   - Assign users: `user11`
5. Confirm role attributes by double-clicking the `tarback` entry and stepping through the attribute windows.

6. From the command line, check for user and role creation.

   Why does `user11` now appear in the `/etc/user_attr` database?

   _____

   Does the `tarback` role appear in both the `/etc/passwd` file and the `/etc/user_attr` database?

   _____

7. To test the role, log in as `user11`.
8. Execute several commands to verify that the account is functional.
9. Execute the `tar` command to back up the `.profile` file.

   Can you back up this file?

10. Execute the `tar` command to back up the `/etc/shadow` file.

    Can you back up this file?

11. Switch to the `tarback` role.
12. Execute several commands to verify that the account is functional.
13. Execute the `tar` command to back up the `/etc/shadow` file.

    Can you back up this file?

    _____

    What is the difference, if any, between the executing the `tar` command as `user11` and executing the `tar` command after assuming the `tarback` role?

    _____

14. List the RBAC commands that can be executed using the `tarback` role.

Exercise: Configuring RBAC (Level 3)

In this exercise, you configure RBAC by using the command line in the first task and by using the Solaris Management Console in the second task.

Preparation

During the lab, you are directed to execute commands that do not work to demonstrate how the RBAC facility must be used by logged in users.

Discuss how to use the `auths`, `profiles`, and `roles` RBAC commands to determine user privileges.

Task Summary

Perform the following tasks:

- Using the command-line tools, create a role that can shut down the system, and create a user named `user9`. Assign the role to `user9`to enable `user9` to shut down the system.
- Using the Solaris Management Console, create a user named `user11`, and create a role called `tarback` that can back up the `/etc/shadow` file; make the `tarback` role accessible to `user11`.

If you have any problems that you cannot fix, see your instructor.

Tasks and Solutions

The following section describes the tasks you must perform, along with the solutions to these tasks.

Task 1– Creating a User and a Role Using the Command-line Tools

Complete the following steps:

1. Create a role named `sdown`. Give it a user ID of `5000` and a group ID of `10`.

```
# roleadd -u 5000 -g 10 -m -d /export/home/sdown sdown
# passwd sdown
```

2. Create the profile named `Shut`.

```
# vi /etc/security/prof_attr
Shut:::Able to shutdown the system:
```

3. Add the profile to the role.

```
# rolemod -P Shut,All sdown
```

4. Verify that the role is included in the `/etc/user_attr` file.

```
# more /etc/user_attr
```

5. Create a user named `user9` and assign it access to the `sdown` role. Give this user a user ID of `4009` and a group ID of `10`.

```
# useradd -u 4009 -g 10 -m -d /export/home/user9 -s /bin/ksh \
-R sdown user9
# passwd user9
```

6. Check the roles attributes for `user9`.

```
# grep user9 /etc/user_attr
```

7. Assign the `shutdown` command to the profile.

```
# vi /etc/security/exec_attr
Shut:suser:cmd:::/usr/sbin/shutdown:uid=0
```

8. To test the configuration, log in as `user9`.

9. From this login, shut down the system.

```
$ /usr/sbin/shutdown -i 6 -g 0
/usr/sbin/shutdown:  Only root can run /usr/sbin/shutdown
```

What is the result of this shutdown attempt, and why?

This shutdown attempt fails because, as a regular user, `user9` does not have the rights profile to execute the `shutdown` command.

10. Execute the `profiles` command to determine which RBAC profiles are associated with `user9`.

```
$ profiles
Basic Solaris User
All
```

11. Execute the `roles` command to determine which RBAC roles are associated with `user9`.

```
$ roles
sdown
```

12. Assume the role `sdown`.

```
$ su sdown
Password:$
```

13. Shut down the system by using the `init` command.

```
$ /usr/sbin/init 0
Must be super-user
```

What is the result of this shutdown attempt? Why?

This shut down attempt fails because, even after assuming the `sdown` role, because `user9` does not have the execution attribute to execute the `init` command.

14. List the commands that the `sdown` profile can execute.

```
$ profiles -l

      Shut:
          /usr/sbin/shutdown    uid=0
      All:
          *
```

15. Shut down the system using the `shutdown` command.

```
$ /usr/sbin/shutdown -i 6 -g 0
Shutdown started.    Fri Mar  8 09:51:18 MST 2002

Do you want to continue? (y or n): n
```

What is the result of this shutdown attempt? Why?

This command succeeds because the `sdown` role has execute permission when issuing the `shutdown` command.

16. Log out of the sdown role.

```
$ <Control-D>
```

17. Log out as user9.

```
$ <Control-D>
```

Task 2 – Creating a User and a Role Using the Solaris Management Console

Complete the following steps:

1. Create a new user account with the following specifications:

   Use the Add User Wizard in the Solaris Management Console.

   - Name: user11
   - User ID number: Next available.
   - Password: Set it now to user11
   - Group ID number: Use the default
   - Home directory: /export/home/user11
   - Mailbox: /var/mail/user11 (on this system)

2. Confirm user attributes by double-clicking the user11 entry and stepping through the attribute windows.
3. From the command line, check for user creation.

```
# grep user11 /etc/passwd
user11:x:4011:10:user for tarback role:/home/user11:/bin/sh
# grep user11 /etc/user_attr
#
```

Why does user11 appear in the /etc/passwd file, but not in the /etc/user_attr database?

When a user account is created, a record of the user appears in the /etc/passwd *file and the* /etc/shadow *file. The user record does not appear in the* /etc/user_attr *database until the user has been associated with a role.*

4. Create an administrative role named `tarback` with the following specifications:

   Use the Add Administrative Role wizard in the Solaris Management Console.

   - Name: `tarback`
   - Role ID number: Next available
   - Role shell: Any of the administrator shells
   - Password: `abc123`
   - Rights: As appropriate

   The appropriate rights include Media Backup, Media Restore, and All

Note – A backup administrator must perform all backups of the media as well as any necessary restores.

   - Home directory: `/export/home/tarback`
   - Assign users: `user11`

5. Confirm role attributes by double-clicking the `tarback` entry and stepping through the attribute windows.
6. From the command line, check for user and role creation.

```
# grep user11 /etc/passwd
user11:x:4011:10:user for tarback role:/home/user11:/bin/sh
# grep user11 /etc/user_attr
user11::::roles=tarback;type=normal
# grep tarback /etc/passwd
user11:x:4011:10:user for tarback role:/home/user11:/bin/sh
tarback:x:100:14:can tar the shadow file:/home/tarback:/bin/pfksh
# grep tarback /etc/user_attr
tarback::::profiles=Media Backup,Media Restore,All;type=role
user11::::roles=tarback;type=normal
```

Why does `user11` now appear in the `/etc/user_attr` database?

After associating `user11` *with the* `tarback` *role, an entry that records this relationship should appear in the* `/etc/user_attr` *database.*

Does the `tarback` role appear in both the `/etc/passwd` file and the `/etc/user_attr` database?

Because it is a role, `tarback` *appears in both locations.*

7. To test the role, log in as `user11`.

```
$ telnet sys44
Trying 192.168.30.44...
Connected to sys44.
Escape character is '^]'.


SunOS 5.9

login: user11
Password:
Last login: Thu May  2 14:56:46 from sys44
Sun Microsystems Inc.   SunOS 5.9       Generic May 2002
```

8. Execute several commands to verify that the account is functional.

```
$ id -a
uid=4011(user11) gid=10(staff) groups=10(staff)
$ pwd
/home/user11
```

9. Execute the `tar` command to back up the `.profile` file.

```
$ tar cvf .profile.bak .profile
a .profile 1K
```

Can you back up this file?

The `.profile` *file can be backed up.*

10. Execute the `tar` command to back up the `/etc/shadow` file.

```
$ tar cvf /etc/shadow.bak /etc/shadow
tar: /etc/shadow.bak: Permission denied
```

Can you back up this file?

The `/etc/shadow` *file cannot be backed up by a regular user.*

11. Switch to the `tarback` role.

```
$ su - tarback
Password:
$
```

12. Execute several commands to verify that the account is functional.

```
$ id -a
uid=100(tarback) gid=14(sysadmin) groups=14(sysadmin)
$ pwd
/home/tarback
```

13. Execute the `tar` command to back up the `/etc/shadow` file.

```
$ tar cvf /etc/shadow.bak /etc/shadow
a /etc/shadow 1K
```

Can you back up this file?

Yes.

What is the difference, if any, between the executing the `tar` command as `user11` and executing the `tar` command after assuming the `tarback` role?

The `tarback` *role has the System Administrator rights for media backup and media restore. These rights enable any user that assumes the* `tarback` *role to backup or restore any file.*

14. List the RBAC commands that can be executed using the `tarback` role.

```
$ profiles -l

      Media Backup:
          /usr/bin/mt    euid=0
          /usr/sbin/tar    euid=0
          /usr/lib/fs/ufs/ufsdump    euid=0, gid=sys
      Media Restore:
          /usr/sbin/tar    euid=0
          /usr/bin/cpio    euid=0
          /usr/bin/mt    euid=0
          /usr/lib/fs/ufs/ufsrestore    euid=0
      All:
          *
```

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Performing Smartcard Authentication

Objectives

This module describes the fundamentals of Solaris Smartcard, how to administer Smartcard, and how to use Smartcard for securing a login to the Solaris OE desktop.

Upon completion of the module, you should be able to:

- Describe Smartcard concepts
- Perform Smartcard administration
- Troubleshoot Smartcard operations

The following course map shows how this module fits into the current instructional goal.

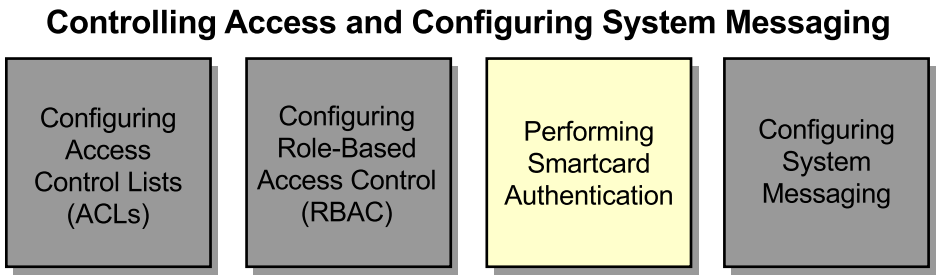


Figure 12-1 Course Map

Introducing Smartcard Concepts

A Smartcard is a card with an embedded microprocessor, a memory chip, or both. Smartcards, unlike magnetic stripe cards, can carry all necessary functions and information on the card. Smartcards are widely used for electronic commerce, communications, computer security, and network applications.

Solaris Smartcard Features

Solaris Smartcard provides an alternate method for logging in to the Solaris desktop environment than is provided by the standard UNIX login. Information stored on the Smartcard verifies the identity of the user during log in. A user who cannot provide the login information that is on the Smartcard is denied access to the desktop. The Solaris Smartcard software:

- Implements the Smartcard framework, which is based on the Open Card Framework (OCF) 1.1 standard
- Supports a variety of card readers
- Supports three widely used Smartcards
- Allows administration management from the Solaris Smartcard console or from the Solaris OE command line
- Protects secure desktop environments through the use of the personal identification number (PIN) authentication method
- Allows a user to store security credentials directly on the card (Java™ technology cards only)

Solaris Smartcard Requirements

To use the Solaris Smartcard software, you need:

- A SPARC system running the Solaris 8 OE or Solaris 9 OE
- A supported internal or external card reader
- Smartcards

Solaris Smartcard supports the following Smartcards and card readers:

- Payflex card

  The Payflex card, which has a microprocessor and up to 8K of memory, is functionally similar to the Micropayflex cards that are used to authenticate login sessions using the Sun Ray™ appliances. Using the Payflex card, you must enter a login PIN, whereas the Micropayflex automatically recalls the current session running within the SunRay cluster.

- Cyberflex card

  The Cyberflex Smartcard has a larger memory capacity (16K or 32K) that enables larger applications or multiple applications to be stored on this Smartcard.

- iButton card

  An iButton card is a microchip that is similar to those used in a Smartcard, but it is housed in round stainless steel button. The iButton supports Java Card™ 2.0/OpenCard standards and uses a special reader. The iButton can be worn as jewelry, such as the "Java™ Ring." More information on the iButton can be found at: `http://www.ibutton.com`.

- Sun SCRI External Serial Card Terminal reader

  The Sun SCRI External Serial Card Terminal reader enables Smartcard functionality to be added to systems that were not originally configured with the Smartcard feature.

- Sun SCRI Internal Card Terminal reader

  The Sun SCRI Internal Card Terminal reader provides an embedded Smartcard reader on workstations, such as the Sun Blade™ product line.

- Dallas Semiconductor iButton Serial Card Terminal reader

  Refer to the `http://www.ibutton.com` Web site.

- GIS Smart Mouse Serial Card Terminal reader

Solaris Smartcard Login

You can secure desktop environments by requiring users to log in with a configured Solaris Smartcard. A desktop that is configured to use Smartcards for login authentication would have a login screen, as shown in Figure 12-2.

Figure 12-2 Smartcard Login Screen

After the user inserts the Smartcard, a personalized login screen is displayed using the user name stored on the card. Figure 12-3 shows that the user is then prompted to enter a PIN. The Solaris Smartcard software compares the typed PIN with the PIN stored on the card.

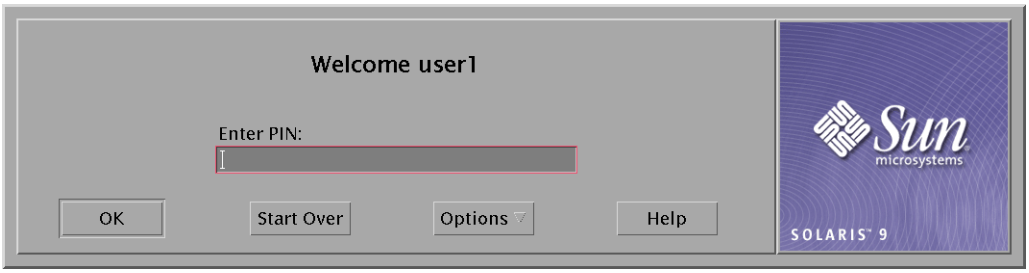


Figure 12-3 User Login Screen

If the PIN matches, the password database specified in the `/etc/nsswitch.conf` file is searched for the password stored on the Smartcard. If the password is found in the system's password database, the Solaris Smartcard software considers the user authenticated, and the user can log in to the desktop.

The OCF Server

The OCF server daemon, `ocfserv`, is a per-host daemon that acts as the central point of communication with all Smartcards connected to the host. Any application that needs to use a Smartcard communicates with the Smartcard through this server, which is responsible for handling all traffic to the Smartcards. At startup time, the server reads the properties file, `/etc/smartcard/opencard.properties`, to determine the readers and cards currently registered.

Note – The OCF server daemon is dynamically started by the `inetd` daemon, using an entry in the `/etc/inet/inetd.conf` file.

Performing Smartcard Administration

Smartcard administration includes the following tasks, which are listed in the order in which they should be performed:

1. Starting the Smartcard console
2. Enabling card readers
3. Activating card services
4. Adding support for a new Smartcard
5. Loading the Smartcard applet on a Smartcard
6. Creating user information on a Smartcard
7. Activating Smartcard operations
8. Configuring Smartcard removal options

Starting the Smartcard Console

The Smartcard console is the graphical user interface (GUI) that you use to manage the Smartcard software.

To start the Smartcard console:

1. Perform one of the following actions:
   - From the command line, log in as `root`, and type:

```
# /usr/dt/bin/sdtsmartcardadmin &
```

   - To start the Smartcard console from CDE, open the Workspace menu by clicking the right mouse button on the desktop.
2. Select the Tools submenu from the Workspace menu, and then select Smart Card.
3. To choose the correct reader installed on your system, select Card Readers from the main SmartCard Console window, as shown in Figure 12-4.

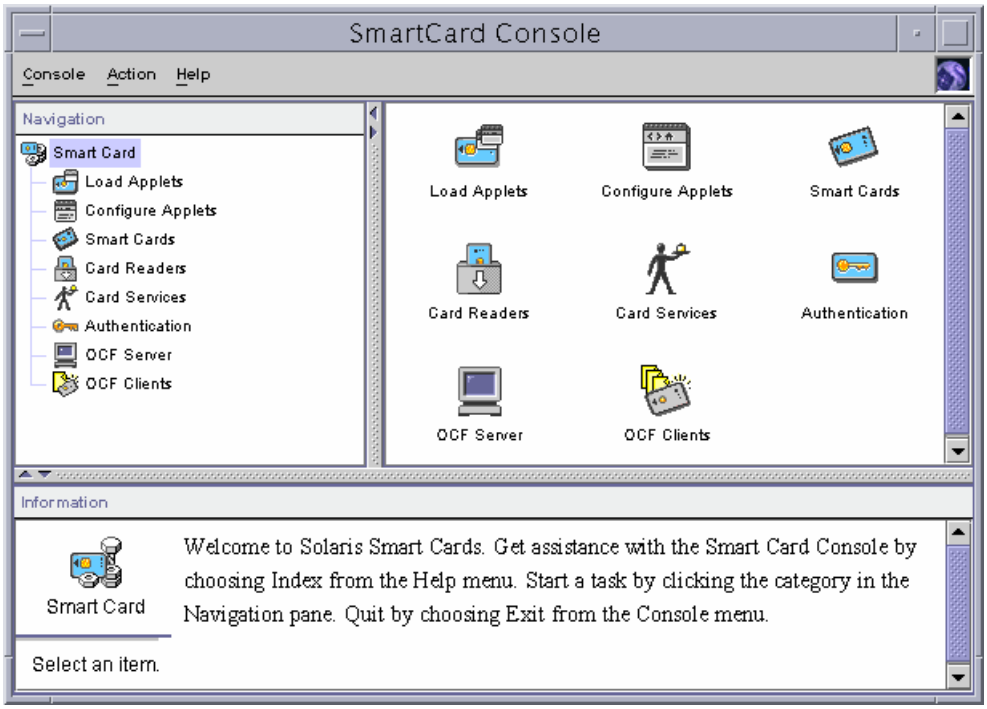


Figure 12-4 SmartCard Console Window

Note – A non-`root` user can run the Smartcard Console, but that user can only perform two tasks: load applets and configure applets.

4. Double-click Card Readers.

   The Card Readers window appears, as shown in Figure 12-5.

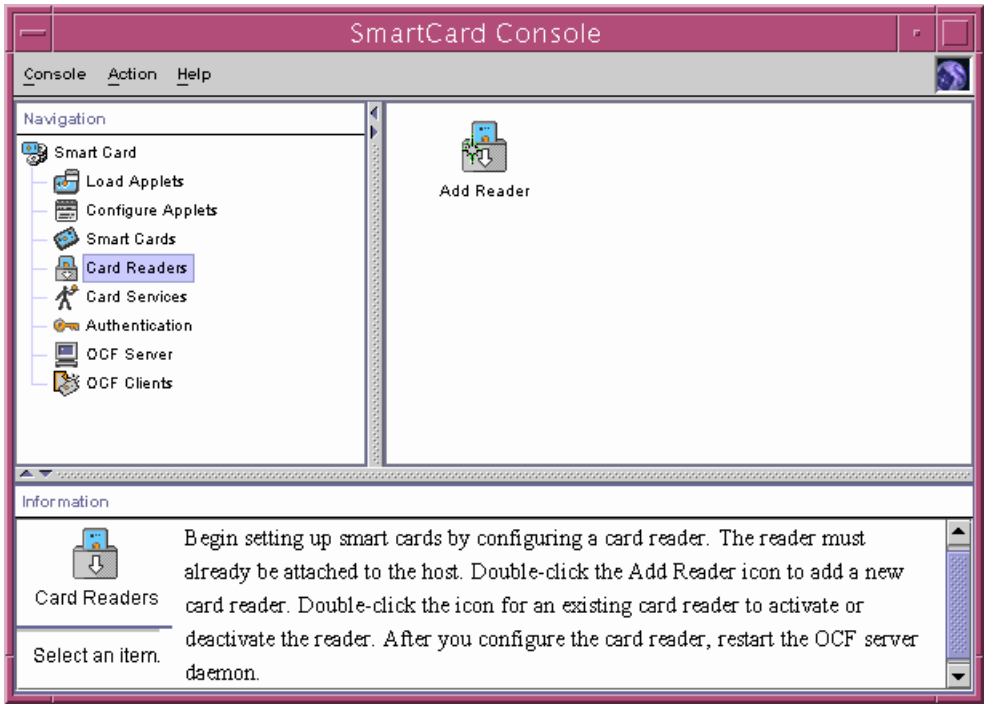


Figure 12-5 Card Readers Window

5. To add a reader, double-click Add Reader.

Enabling a Card Reader

You must enable a card reader before you can use it. The Internal card reader is available with the Sun Blade systems. The external card reader is connected to a serial port and has a power connection through the keyboard connector.

1. Assuming an external card reader is connected, select Sun SCRI External Card Terminal Reader from the list of supported card readers, as shown in Figure 12-6.

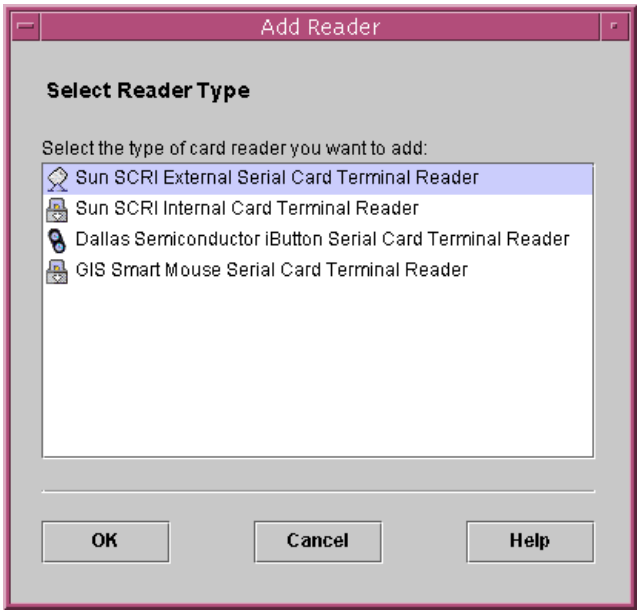


Figure 12-6 Add Reader Window

2. Click OK to continue.

3. In the Card Readers: SunCardReader window:

   a. Choose a device port from the Device drop-down menu, as shown in Figure 12-7.

   b. Select an Activation status.

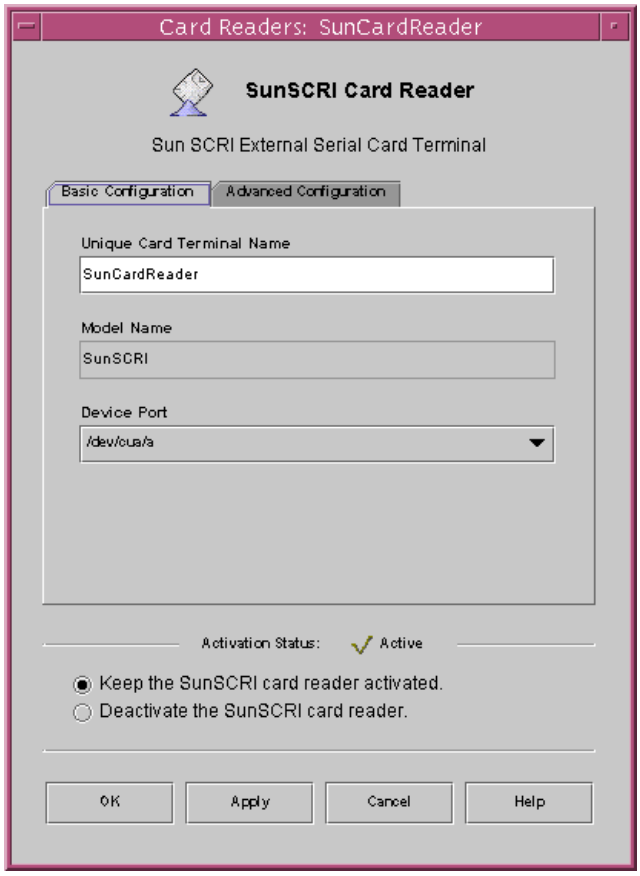


Figure 12-7 Card Readers: SunCardReader Window

4. Click OK to continue.

   When you select the activation status for the card reader, the Intervention Required window appears, as shown in Figure 12-8.

Figure 12-8 Intervention Required Window

5. Click Restart OCF Now.

After you select the restart option for the OCF server, another window appears with the message `OCF Server Restarted`, as shown in Figure 12-9.

Figure 12-9 Informational Message Window

6. Click OK to return to the Smartcard Console window.

   When you enable the card reader, its icon appears in the Card Readers Window's View pane, as shown in the in Figure 12-10, and the Smartcard reader is successfully enabled.

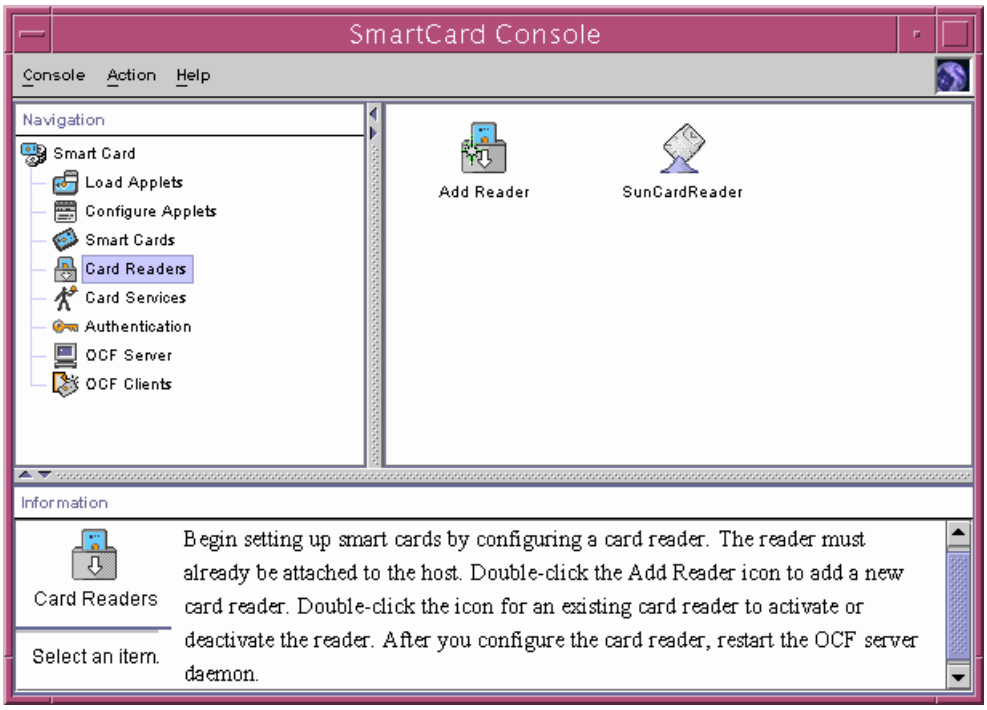


Figure 12-10 Card Readers Window – View Pane

Activating Card Services

To activate Smartcard services, perform the following steps:

1. From the Navigation pane in the Card Services window, click Card Services.

   The Smartcards known to the server are displayed, as shown in Figure 12-11.

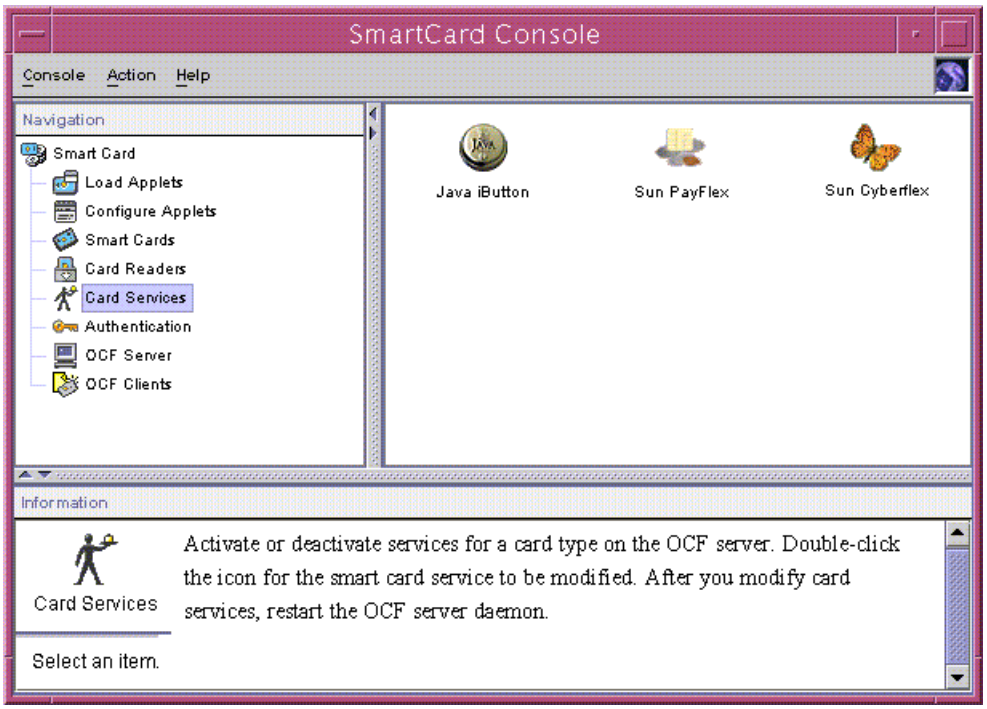


Figure 12-11 Card Services Window

2. Double-click one of the Smartcard icons in the View pane. For example, click the Sun PayFlex icon.

   The Card Services window launches, as shown in Figure 12-12 on page 12-13.

3. In the Card Services window, either:

   - Deactivate the set of services by selecting Deactivate Sun PayFlex services and then clicking OK.
   - Keep the Sun PayFlex services activated by clicking OK to continue.

   The services that are currently supported by the selected card type are displayed, as shown in Figure 12-12. The services are either all on or all off.

Figure 12-12 Card Services: Sun PayFlex Window

4. Click OK.

Adding Support for a New Smartcard

The answer-to-reset (ATR) property contains numeric values that identify the Smartcard version. When the Smartcard is first inserted into the reader, it is powered-on or reset. The Smartcard must respond with a recognized ATR for communications with the server to continue. Smartcard manufacturers supply the ATR property. When you set up Smartcards, you must identify the ATR on the Smartcard to the OCF server.

To add support for a new Smartcard, perform the following steps:

1. Insert your PayFlex Smartcard into the card reader.
2. Double-click the Smart Cards icon.
3. Double-click the PayFlex icon in the View pane, as shown in Figure 12-13.

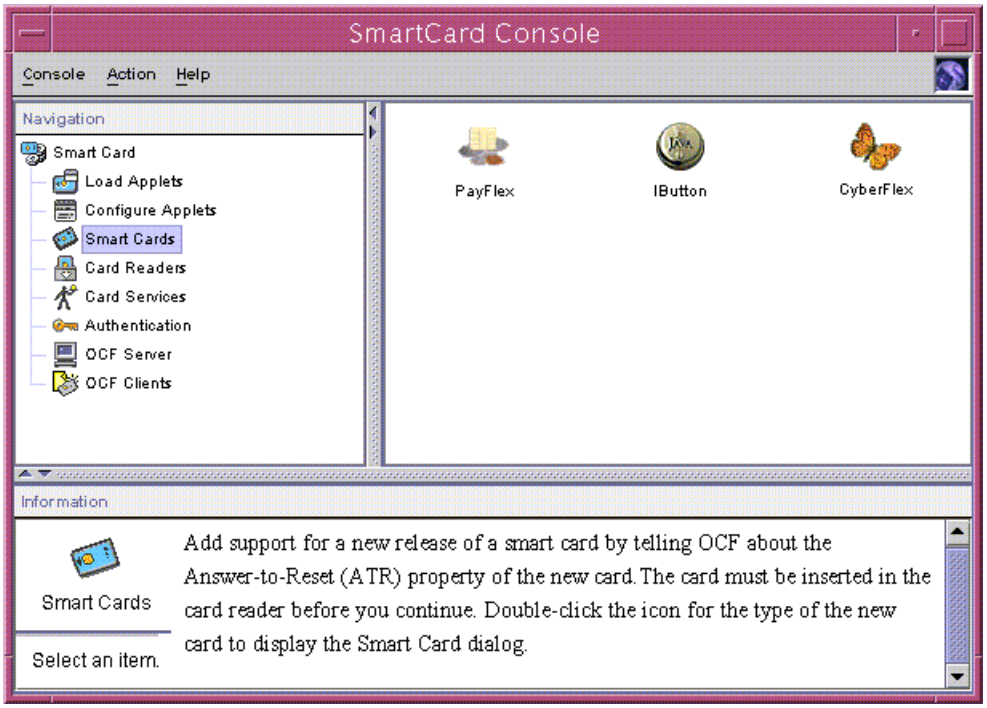


Figure 12-13 Smart Cards Window

Note – You must change the ATR on a system if the manufacturer of the Smartcard that you are using issues a new card type with a different ATR.

The Smart Card: PayFlex window appears and displays the known ATRs for that type of card. For example, the known PayFlex models are displayed, as shown in Figure 12-14.

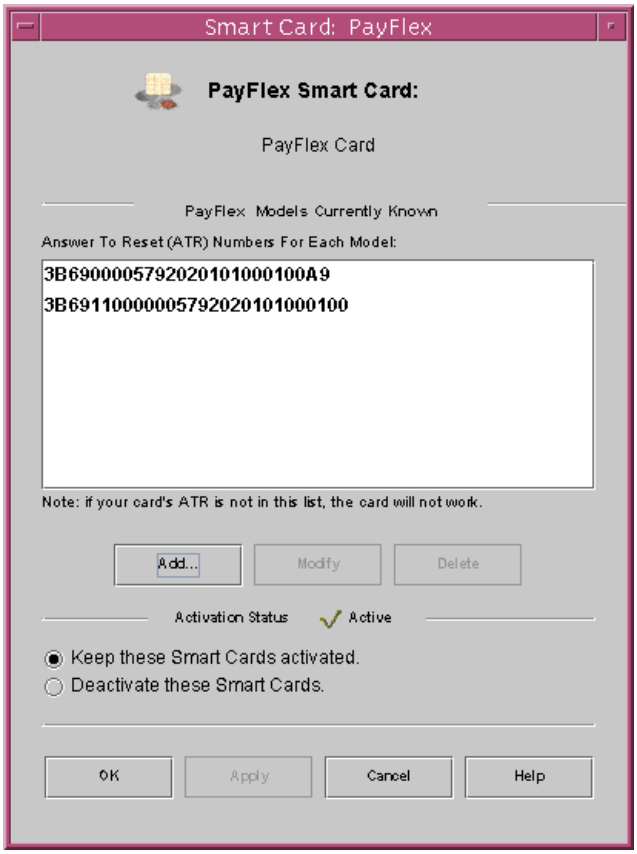


Figure 12-14 Smart Card: PayFlex Window

Because the PayFlex Smartcard currently in use has an ATR that is not known to the default OCF server configuration, you must add this ATR to the ATR list recognized by the server.

4. Click Add to add the ATR.

The Add ATR window appears, as shown in Figure 12-15.

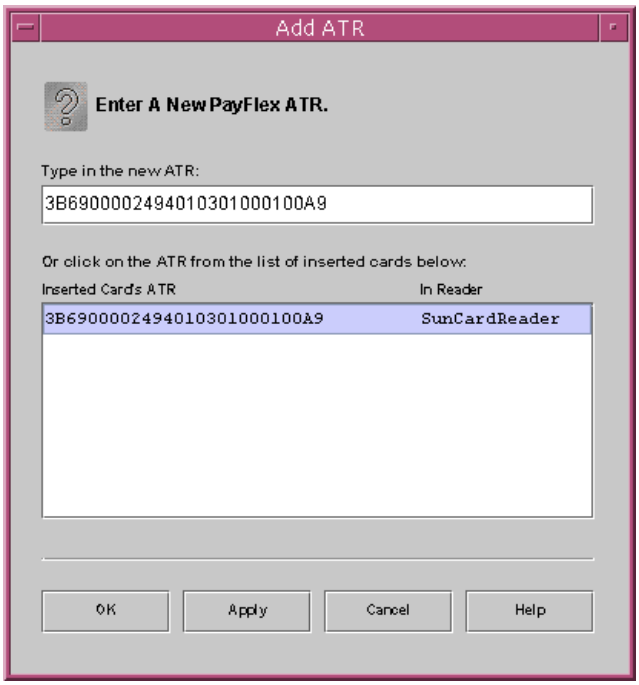


Figure 12-15 Add ATR Window

5. Select the ATR that is highlighted in blue to move the new ATR information into the input field at the top of the window.

Note – If the ATR does not appear in the Add ATR window, verify that the Smartcard is inserted with the correct side up. If the ATR still does not appear, contact the card manufacturer for the ATR number, and manually enter the number.

6. Click OK to add this ATR to the list of currently known ATRs.

The new ATR is displayed, as shown in Figure 12-16.

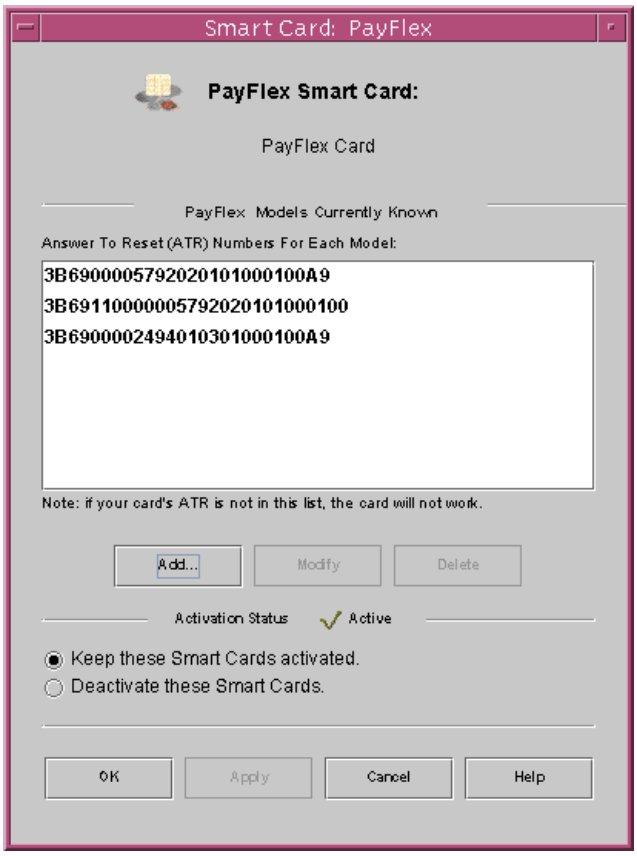


Figure 12-16 Answer to Reset (ATR) Numbers for Each Model List Box

7. After the OCF server recognizes the Smartcard, select the activation status.
8. Click OK to continue.

Loading the Smartcard Applet to a Smartcard

The SolarisAuthApplet applet contains the functions needed to store and use a user's profile information. You must load the applet onto all card types supported by Solaris Smartcards.

To load the Smartcard applet to a Smartcard, perform the following steps:

1. From the SmartCard Console, click the Load Applets icon.

   The View pane displays the available applets, as shown in Figure 12-17. In this example, the SolarisAuthApplet is the only applet supported at this time.

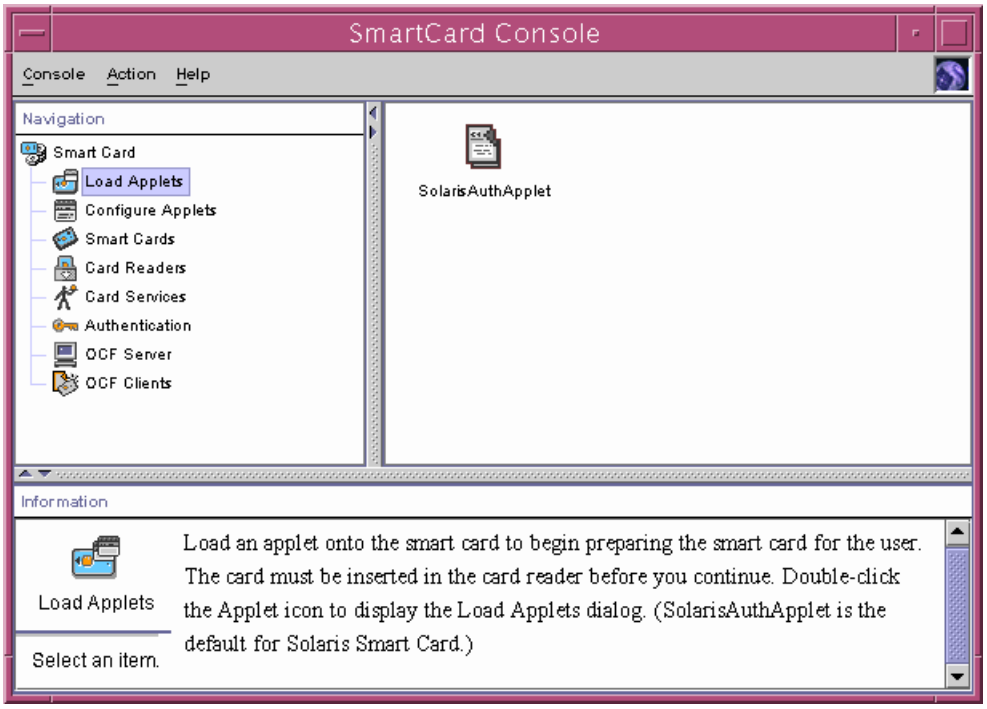


Figure 12-17 SmartCard Console Window – Load Applets

2. Double-click the applet icon.

The Load Applets window appears, as shown in Figure 12-18.

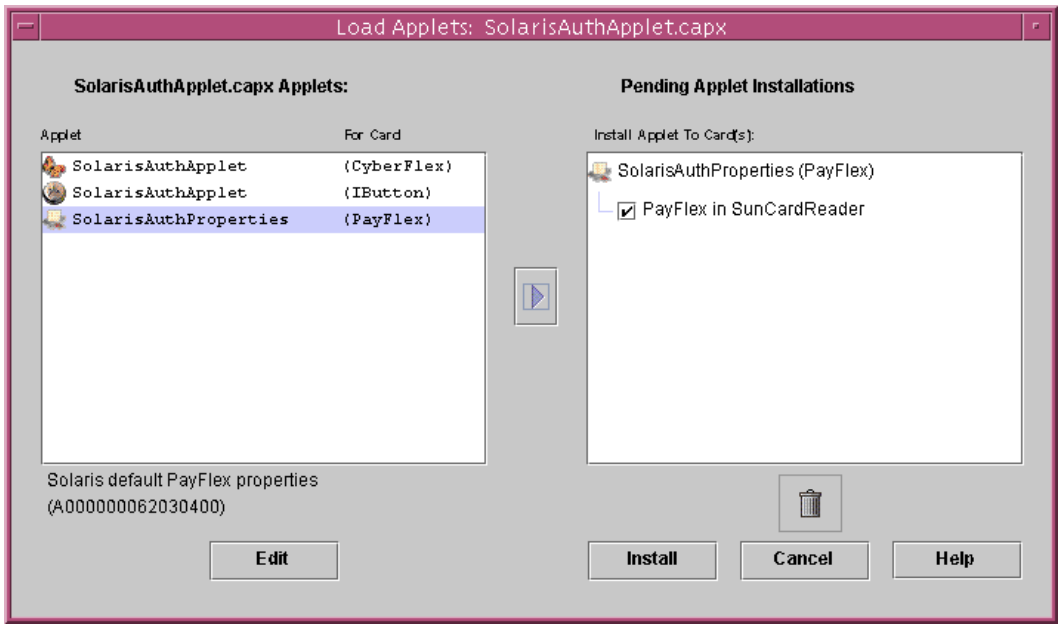


Figure 12-18 Load Applets Window

3. Select the correct SolarisAuth applet for the card currently in the reader.
4. Click the arrow to move the applet to the right box, and click Install.

   The Loading Applet to Device window appears, as shown in Figure 12-19.

Figure 12-19 Loading Applet to Device Window

5. Click OK to continue loading the applet.

Note – It might take 30 seconds or more to load the applet. During this time, nothing appears on the screen.

When the applet is loaded, the Applet Installation Successful window appears, as shown in Figure 12-20.

Figure 12-20 Applet Installation Successful Window

Note – The Payflex Smartcards do not have the capability to delete or reload applets.

6. Click OK to continue.

Creating User Information on a Smartcard

After the SolarisAuth applet is loaded onto the Smartcard, you configure the Smartcard for the user.

To create user information on a Smartcard:

1. In the Smartcard Console window, click the Configure Applets icon in the Navigation pane.

   The available Cards and Readers appear, as shown in Figure 12-21.

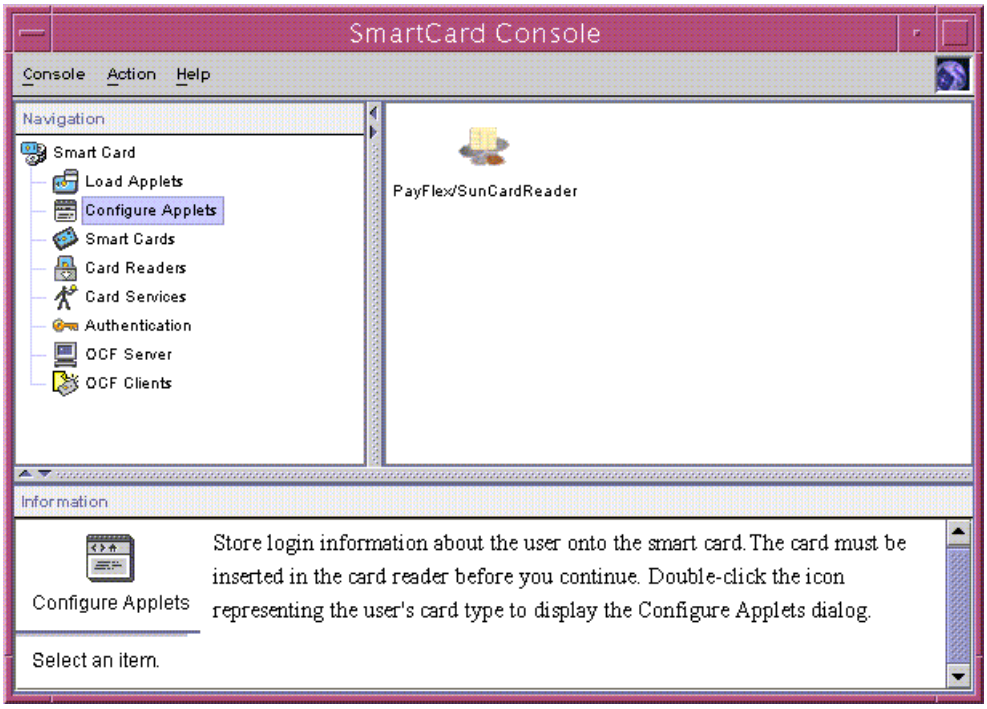


Figure 12-21 Smartcard Console

2. Double-click the PayFlex/SunCardReader icon.

The Configure Applets: Payflex window appears, as shown in Figure 12-22.

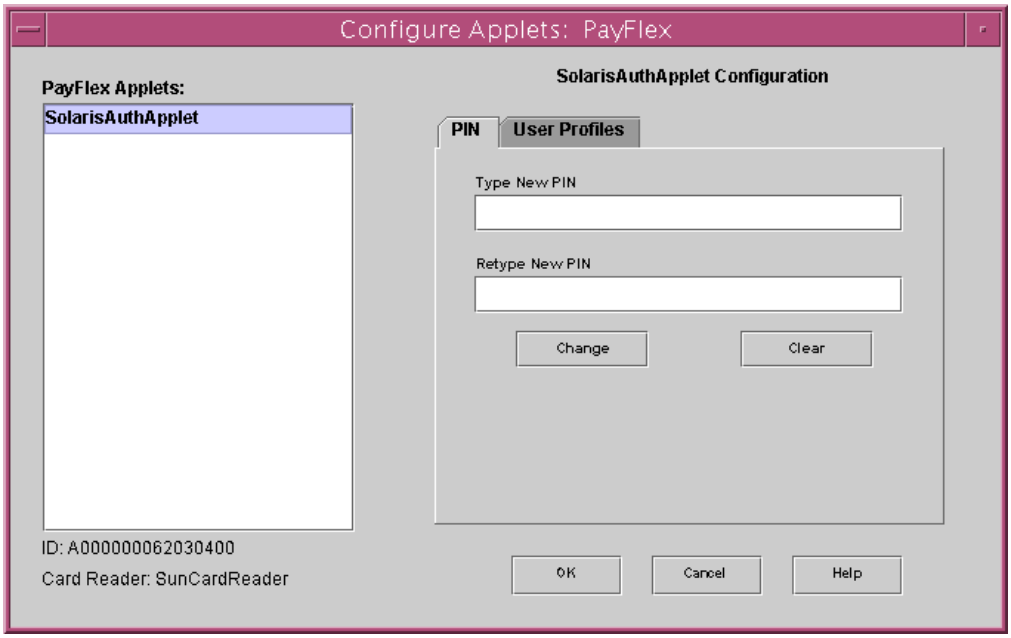


Figure 12-22 Configure Applets: PayFlex Window

3. In the left field, click the applet in the applets list.

   There is only one applet to configure, as shown in Figure 12-22.

4. To set a new PIN:
   a. Select the PIN Configuration tab.
   b. Enter the new PIN, and click Change.

      The Change PIN: Enter PIN window appears, as shown in Figure 12-23.

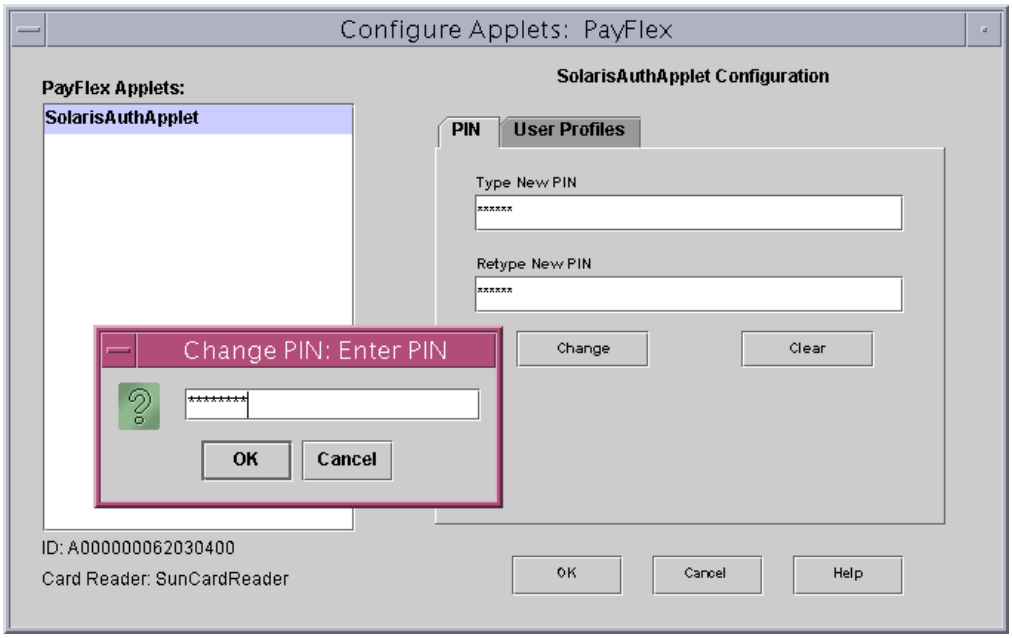


Figure 12-23 Change PIN: Enter PIN

   c. Enter the current PIN. The default (current) PIN is set to `$$$$java`.
   d. Click OK.

5. To modify the user profiles:
   a. Click the User Profiles tab.

      Currently the `dtlogin` application is the only available and supported application. Therefore, the profile name must be `dtlogin`.

   b. Type **`dtlogin`** in the User Profile Name field.
   c. Add a valid user name and password for this card.
   d. Click Set to update the user profile.

Note – Users can change their own PIN using the SmartCard Console.

The Set User Profile: Enter PIN window appears, as shown in Figure 12-24.

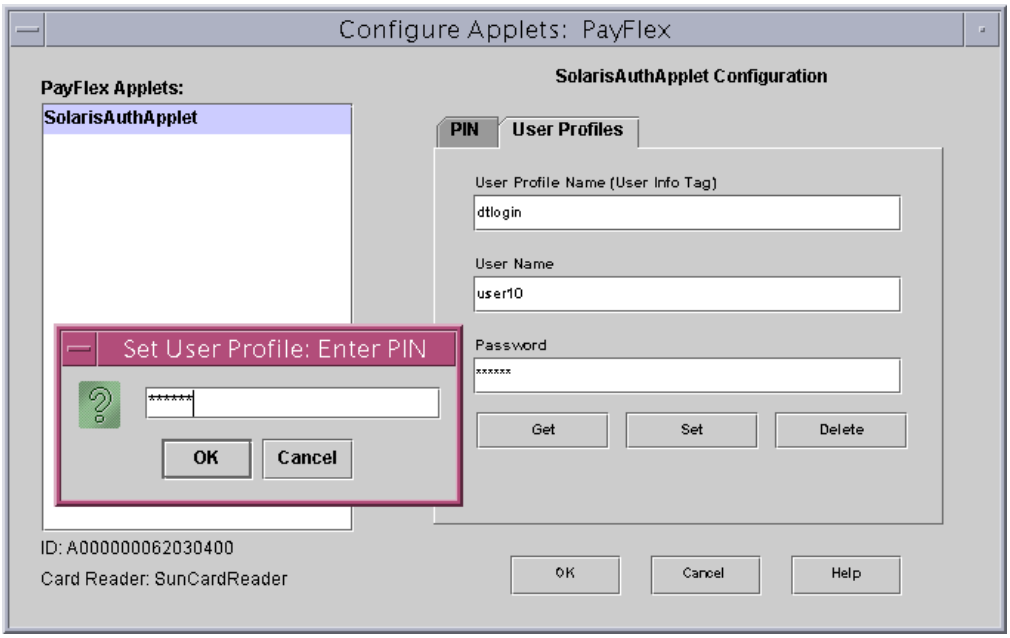


Figure 12-24 Set User Profile: Enter PIN Window

e. Enter a PIN for the user profile.

Caution – Do not forget the new PIN. You cannot modify the current information on the card without the PIN.

f. Click OK in the Set User Profile: Enter PIN window.

6. Click OK.

Activating Smartcard Operations

The Smartcard is now configured and ready to use. Next, you must activate the application configured for that Smartcard on the client.

When you activate a Smartcard, you use The Desktop Configuration Dialog window and its four tabs:

- Cards/Authentications – Displays the current cards and the authentication scheme used by the desktop.
- Defaults – Lets you set defaults from a list of available resources for the desktop. These resources include the Smartcards, Card Reader, and type of Authentication.
- Timeouts – Modify functionality
- Options – Modify functionality

To activate Smartcard operations:

1. In the SmartCard Console window, click the OCF Clients icon.

   The available clients appear, as shown in Figure 12-25.

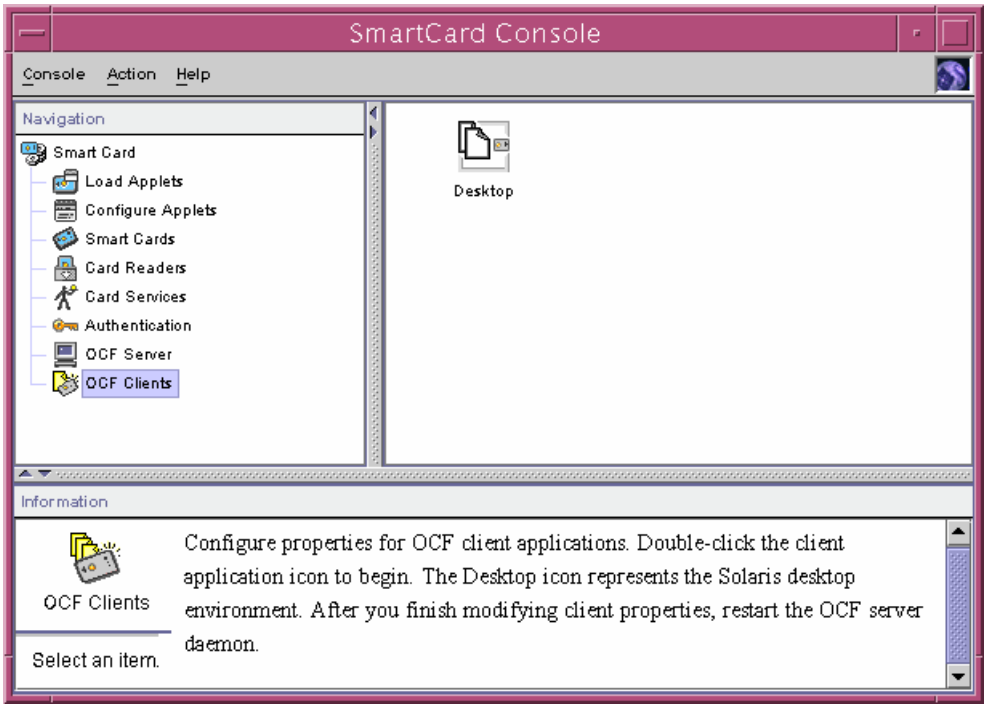


Figure 12-25 SmartCard Console Window

2. Double-click the Desktop icon.

The Cards/Authentications Used by Desktop window appears, as shown in Figure 12-26

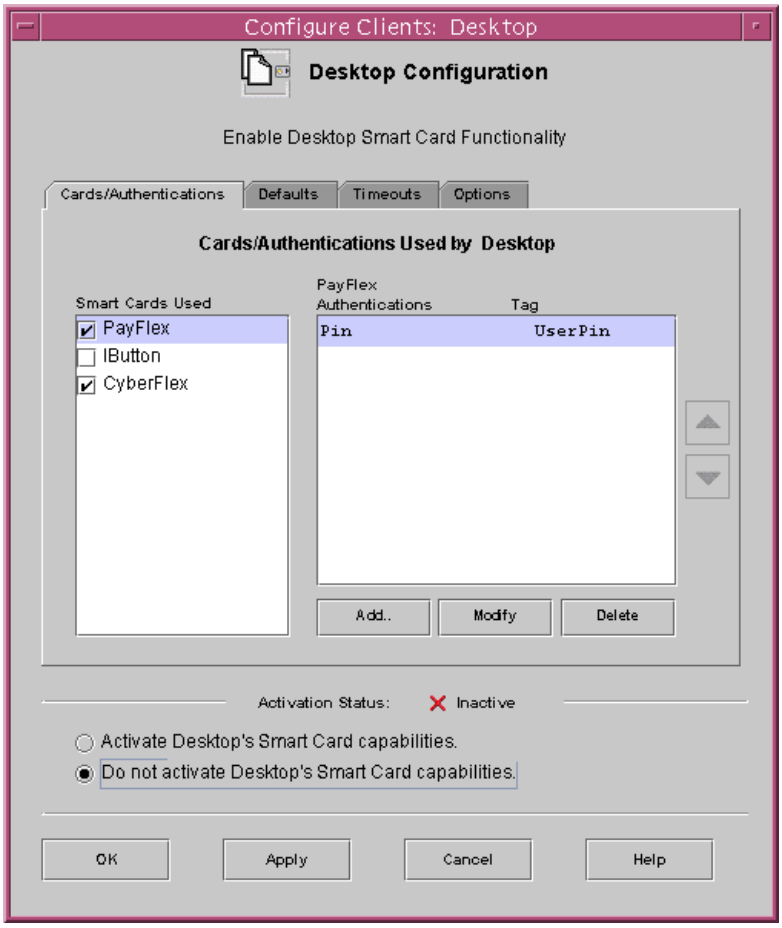


Figure 12-26 Cards/Authentications Used by Desktop Window

3. Select PayFlex in the Smart Cards Used field.

Note – When you click PayFlex, two fields, `Pin` and `User Pin`, appear in the right pane. Do not modify these fields.

4. Click Add.
5. Because the current status of the Desktop's Smartcard capabilities is shown as Inactive, select Activate Desktop's SmartCard capabilities.

6. Select the Defaults tab.

   The Default Resources for Desktop window appears, as shown in Figure 12-27. In this window, you can specify a specific card and reader or select the default that is set for the OCF Server.

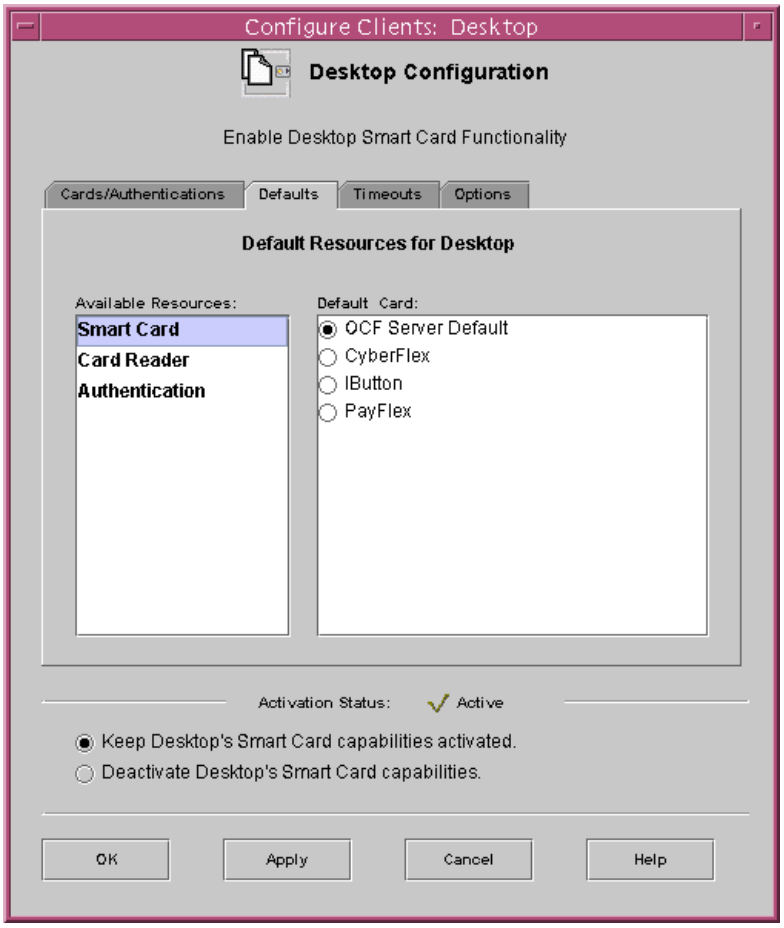


Figure 12-27 Default Resources for Desktop Window

7. Click OK to continue.

Configuring Smartcard Removal Options

You use the Timeouts and Options tabs of the Desktop Configuration window to modify the desktop Smartcard functionality. In other words, you are configuring the behavior of the desktop when the card is removed from the reader.

In the Timeouts tab, as shown in Figure 12-28, there are three sliders:

- Card Removal Timeout – The number of seconds that the desktop waits after a Smartcard is removed before locking the screen.
- Reauthentication Timeout – The number of seconds that the Reauthentication Screen is displayed.
- Card Removal Logout Wait Timeout – The number of seconds that the desktop waits for a Smartcard to be reinserted before the desktop displays the Reauthentication screen. If the card is not reinserted in that amount of time, the user is logged out.

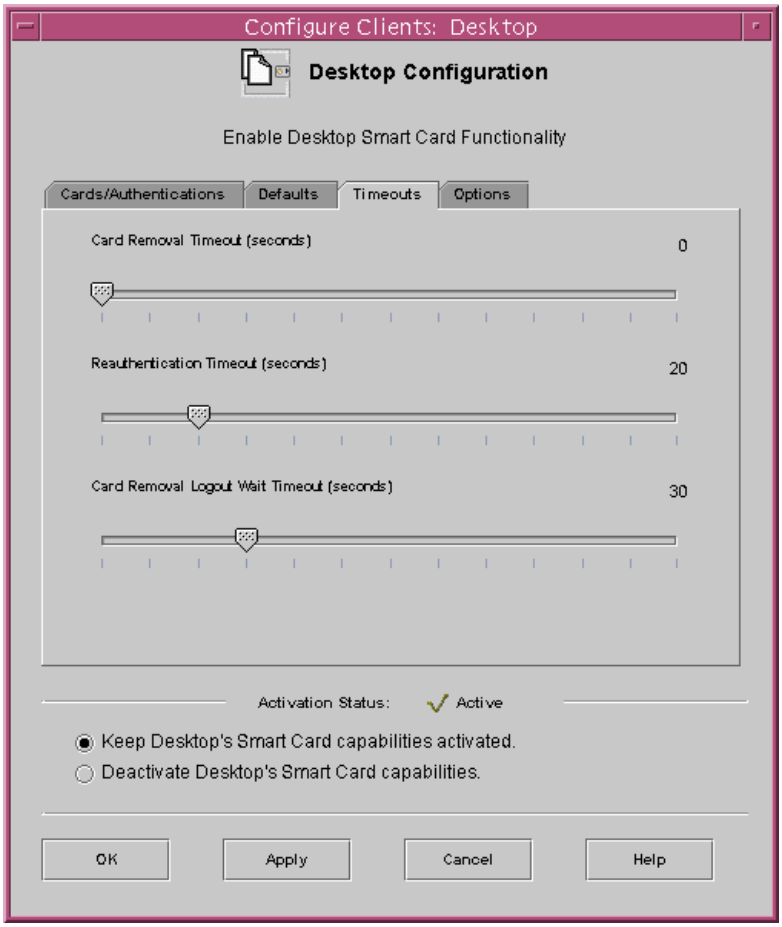


Figure 12-28 Desktop Timeouts Configuration Tab

The Options tab, as shown in Figure 12-29, has two options:

- Ignore Card Removal – When selected, removing the Smartcard does not invoke a lock screen or logout.
- Reauthenticate After Card Removal – When selected, the Reauthentication Screen is immediately launched when the Smartcard is removed. When not selected, the Reauthentication Screen is controlled by the Card Removal Logout Wait parameter set in the Timeouts tab.

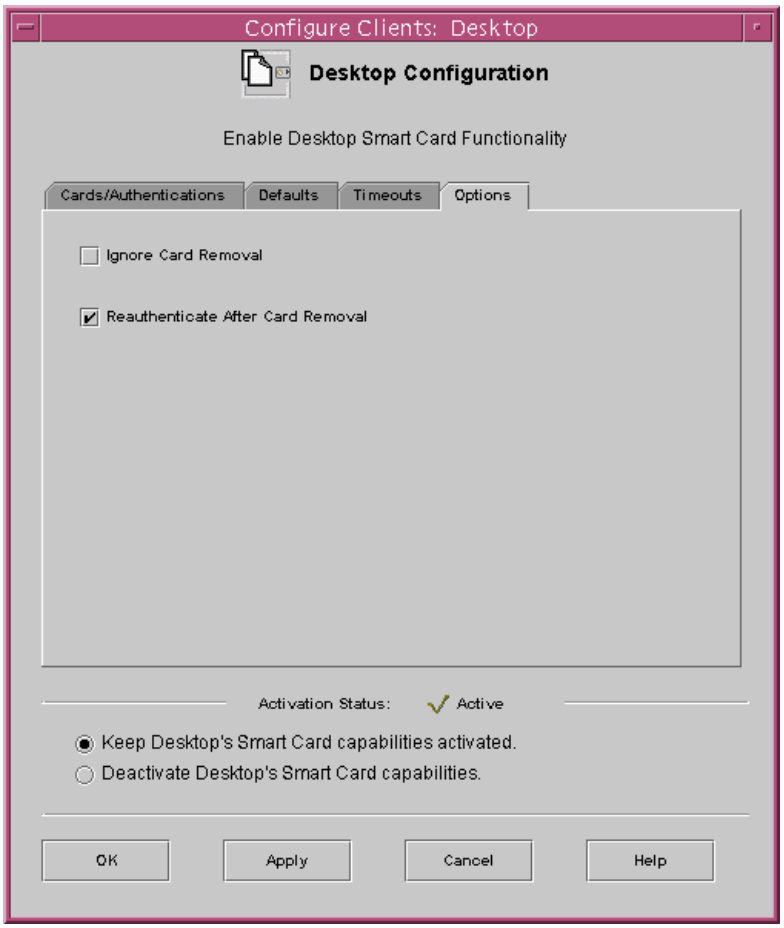


Figure 12-29 Options Tab

To test whether you have successfully configured and activated the Smartcard, complete the following steps:

1. Remove the card from the card reader.
2. Exit your current login session.

   The Display Locked Screen window, as shown in Figure 12-30, appears.

Figure 12-30 Display Locked Screen

3. Insert the card into the card reader.
4. Enter your login PIN.

   Your new session starts.

Troubleshooting Smartcard Operations

The following sections provide some procedures for troubleshooting Smartcard operations.

Enabling Debugging

The OCF Server in the SmartCard Console, shown in Figure 12-31, generates a text-formatted log file. You set server debug levels and the OpenCard tracing level to record the necessary information for debugging and reporting problems to technical support.

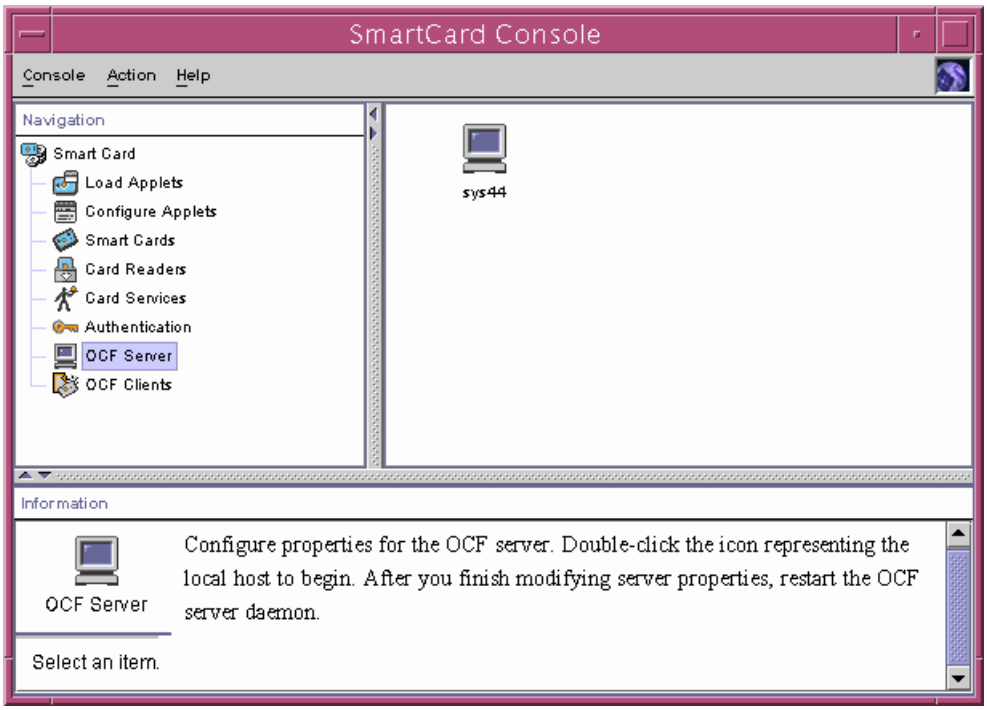


Figure 12-31 Smartcard Console

To enable optional debugging using the SmartCard Console:

1. Select the OCF Server from the Navigation pane.
2. Double-click the icon representing the local system.

The OCF Server Administration window appears, as shown in Figure 12-32.

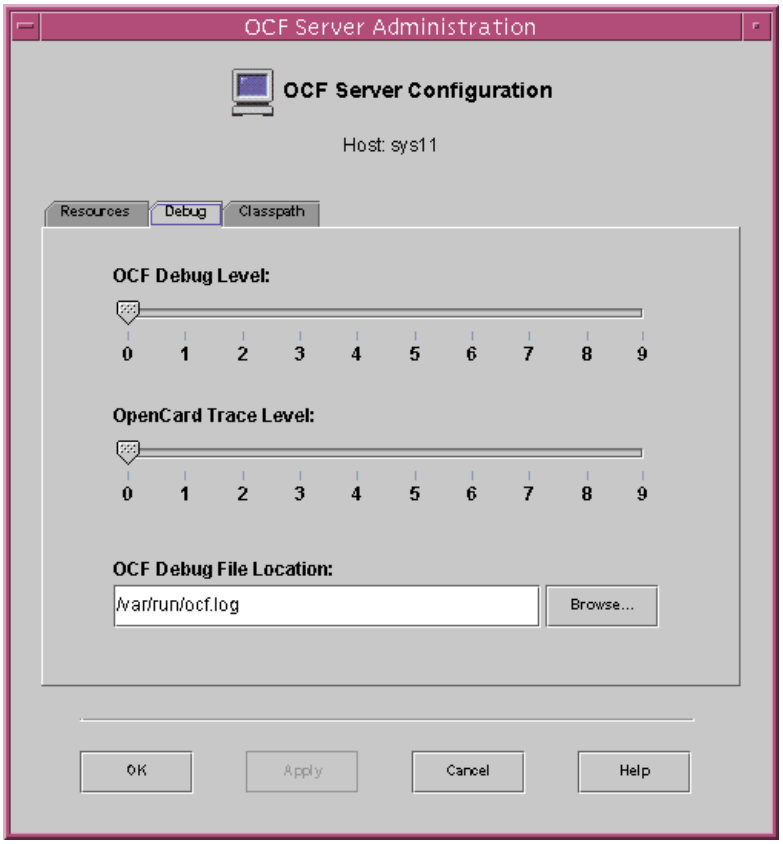


Figure 12-32 OCF Server Administration Window

3. Select the Debug tab.
4. To indicate the level of debugging you want, use the OCF Debug Level slider.
5. To indicate the trace level you want, use the OpenCard Trace Level slider.
6. If necessary, change the default debug file `/var/run/ocf.log` in the OCF Debug File Location field.
7. Click OK to make the changes.

Disabling Smartcard Operations

You might need to disable Smartcard operations if a Smartcard configuration error does not allow a user to log in with a Smartcard, or if a system no longer needs a Smartcard login. As the `root` user, type the following command to disable Smartcard operations:

```
# smartcard -c disable
```

Resolving Smartcard Configuration Problems

Smartcard configuration information is stored in the `/etc/smartcard/opencard.properties` file. This file does not require administration and should not be edited manually. However, if you inadvertently introduce a problem in your Smartcard configuration by using the Smartcard console or the command line, you can restore the previous version of the `opencard.properties` file.

Note – This procedure assumes you have previously copied the `opencard.properties` file to `opencard.properties.bak`.

As the `root` user, from the command line, perform the following steps:

1. Change to the `/etc/smartcard` directory.
2. Save the current version.

```
# cp opencard.properties opencard.properties.bad
```

3. Copy the previous version to the current version.

```
# cp opencard.properties.bak opencard.properties
```

You can display the current client and server configuration by typing the following on the command line:

```
# smartcard -c admin

Client Properties:
  ClientName.PropertyName          Value
  -----------------------          -----
  Desktop.IButton.authmechanism  = Pin=UserPin
  Desktop.validcards             = CyberFlex PayFlex
  Desktop.PayFlex.authmechanism  = Pin=UserPin
  default.validcards             = CyberFlex IButton PayFlex
  default.authmechanism          = Pin=UserPin
  default.defaultaid             = A000000062030400

Server Properties:

  PropertyName                     Value
  ------------                     -----
  authservicelocations           = com.sun.opencard.service.auth
  OpenCard.trace                 = com.sun:1 opencard.core:1
  initializerlocations           = com.sun.opencard.cmd.IButtonInit
  debugging                      = 1
  debugging.filename             = /var/run/ocf.log
  ocfserv.protocol               = rpc
  authmechanism                  = Pin Password
  language                       = en
  cardservicelocations           = com.sun.opencard.service.common
  IButton.ATR                    =
008F0E000000000000000000000004000034909000
  country                        = US
  CyberFlex.ATR                  = 3B169481100601810F 3B169481100601811F
  OpenCard.services              =
com.sun.opencard.service.cyberflex.CyberFlexS
erviceFactory com.sun.opencard.service.ibutton.IButtonServiceFactory
com.sun.ope
ncard.service.payflex.PayFlexServiceFactory
  PayFlex.ATR                    = 3B690000579202010100010A9
3B6911000000579202
0101000100 3B690000249401030100010A9
  OpenCard.terminals             =
com.sun.opencard.terminal.scm.SCMStc.SCMStcCa
rdTerminalFactory|SunCardReader|SunSCRI|/dev/cua/a
  keys.chkey.dtlogin.user1       = AD5BE9EAE0CD8C15
  keys.chkey.login.user          = 3132333435363738
  SunCardReader.0.ATR            = 3B690000249401030100010A9
```

Resolving Smartcard ATR Problems

When trying to download an applet to a Smartcard, an error message "SmartcardInvalidCardException" might indicate that the ATR of the Smartcard inserted in the card reader has not been added as a valid ATR for that card type. When selecting the card type from the Smartcards window in the Smartcard Console, if the ATR does not automatically appear in the Add ATR window, you must add the ATR manually. The card manufacturer will provide the ATR for you.

Resolving Smartcard Login Problems

When Smartcard operations are enabled, the Common Desktop Environment (CDE) login screen displays the prompt: `please insert Smartcard`. If you cannot log in to the system using a Smartcard because of Smartcard setup problems, log in remotely with the `rlogin` or `telnet` commands. You can also choose the CDE command-line login from the local system. Become the `root` user, and disable Smartcard operations from the command line:

```
# smartcard -c disable
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Configuring Smartcard for Desktop Authentication (Level 1)

In this exercise, you configure a Smartcard and configure the desktop to use a Smartcard for login authentication.

Preparation

To prepare for this exercise, refer to the material in the module.

Note – The `delete` command is not available on the Payflex Smartcards. Therefore, once an applet is loaded onto a Payflex Smartcard, it cannot be unloaded. CyberFlex Smartcards can unload applets.

Tasks

Using the SmartCard Console window, perform the following tasks:

- Enable the card reader
- Configure your Smartcard
- Activate Smartcard operations
- Test your Smartcard login

Exercise: Configuring Smartcard for Desktop Authentication (Level 2)

In this exercise, you configure a Smartcard and configure the desktop to use a Smartcard for login authentication.

Preparation

To prepare for this exercise, refer to the material in the module.

Note – The `delete` command is not available on the Payflex Smartcards. Therefore, once an applet is loaded onto a Payflex Smartcard it cannot be unloaded. CyberFlex Smartcards can unload applets.

Task Summary

Using the SmartCard Console window, perform the following tasks:

- Enable the card reader
- Configure your Smartcard
- Activate Smartcard operations
- Test your Smartcard login

Tasks

Complete the following steps:

1. As the `root` user, start the SmartCard Console.
2. Select and enable the correct card reader.
3. Activate Card Services for your card.
4. Add support for a new Smartcard.
5. Load the Smartcard applet to your Smartcard.
6. Configure the PIN and user profile.
7. Activate Smartcard operations on the desktop.
8. Log out, and verify the login using the Smartcard.

9. Log in as `user11`, and start the Smartcard Console.
10. Reset the PIN to the default value, and reset the user profile to blank.
11. Log out, and attempt to log in again with the Smartcard.
12. Use the `telnet` command to connect to the host with the Smartcard reader, and disable Smartcard from the command line.

Exercise: Configuring Smartcard for Desktop Authentication (Level 3)

In this exercise, you configure a Smartcard and configure the desktop to use a Smartcard for login authentication.

Preparation

To prepare for this exercise, refer to the material in the module.

Note – The `delete` command is not available on the Payflex Smartcards. Therefore, once an applet is loaded onto a Payflex Smartcard, it cannot be unloaded. CyberFlex Smartcards can unload applets.

Task Summary

Using the SmartCard Console window, perform the following tasks:

- Enable the card reader
- Configure your Smartcard
- Activate Smartcard operations
- Test your Smartcard login

Tasks and Solutions

Complete the following steps:

1. As the `root` user, start the SmartCard Console.

```
# /usr/dt/bin/sdtsmartcardadmin &
```

2. Select and enable the correct card reader.

   For more information, see Figure 12-6 on page 12-9.

3. Activate Card Services for your card.

   For more information, see Figure 12-12 on page 12-13.

4. Add support for a new Smartcard.

   For more information, see Figure 12-14 on page 12-15 and Figure 12-16 on page 12-17.

5. Load the Smartcard applet to your Smartcard.

   For more information, see Figure 12-18 on page 12-19.

6. Configure the PIN and user profile.

   For more information, see Figure 12-23 on page 12-23 and Figure 12-24 on page 12-24.

7. Activate Smartcard operations on the desktop.

   For more information, see Figure 12-25 on page 12-25.

8. Log out, and verify the login using the Smartcard.
9. Log in as `user11`, and start the Smartcard Console.

```
# /usr/dt/bin/sdtsmartcardadmin &
```

10. Reset the PIN to the default value, and reset the user profile to blank.

    For more information, see Figure 12-23 on page 12-23 and Figure 12-24 on page 12-24.

11. Log out, and attempt to log in again with the Smartcard.

    Login fails, you can no longer access the desktop.

12. Use the `telnet` command to connect to the host with the Smartcard reader, and disable Smartcard from the command line.

```
# telnet instructor
Trying 192.168.0.1...
Connected to instructor.
Escape character is '^]'.

SunOS 5.9

login: root
Password:
# smartcard -c disable
# exit
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 13

Configuring System Messaging

Objectives

The `syslog` system messaging features track system activities and events. You can manually generate log messages by using the `logger` command. The Solaris Management Console activity is tracked by using the messaging facilities available to the Solaris Management Console. Regardless of the type of information you want to record, a messaging feature exists to record it.

Upon completion of this module, you should be able to:

- Describe the fundamentals of the `syslog` function
- Configure the `/etc/syslog.conf` file
- Configure `syslog` messaging
- Use the Solaris Management Console log viewer

The following course map shows how this module fits into the current instructional goal.

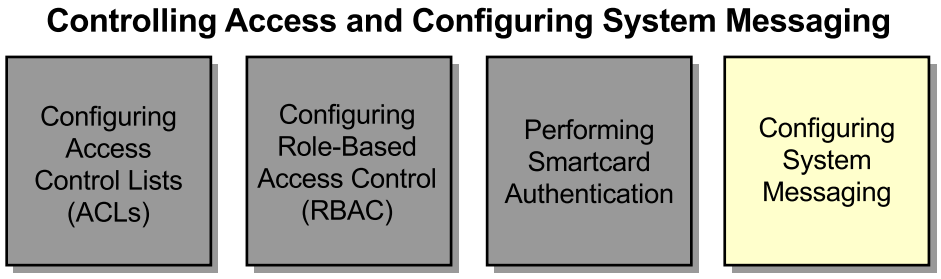


Figure 13-1 Course Map

Introducing the syslog Function

The syslog function, the syslogd daemon, and input from the /etc/syslog.conf file work together to facilitate system messaging for the Solaris 9 Operating Environment (Solaris 9 OE).

The syslog Concept

The syslog function sends messages generated by the kernel programs and system utilities to the syslogd daemon, as shown in the Figure 13-2. With the syslog function you can control message logging, depending on the configuration of the /etc/syslog.conf file. The daemon can:

- Write messages to a system log
- Forward messages to a centralized log host
- Forward messages to a list of users
- Write messages to the system console

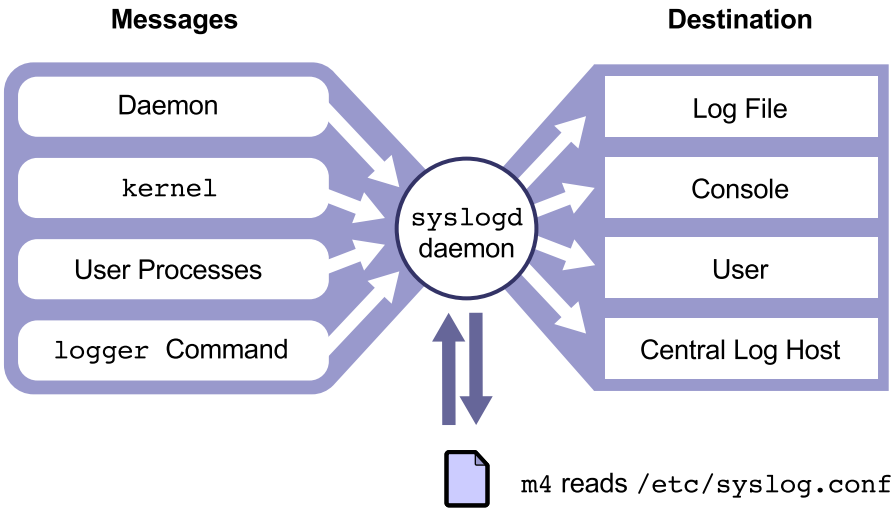


Figure 13-2 The syslog Structure

The /etc/syslog.conf File

A configuration entry in the /etc/syslog.conf file consists of two tab-separated fields*: selector* and *action*.

The selector field has two components, a *facility* and a *level* written as *facility.level*. Facilities represent categories of system processes that can generate messages. Levels represent the severity or importance of the message.

The action field determines where to send the message.

For example, when you place the following entry in the /etc/syslog.conf file, error messages for all facilities are sent to the /var/adm/messages file:

```
*.err      /var/adm/messages
```

where:

*.err	Is the selector field. The asterisk (*) is the *facility*, and the dot (.) is the delimiter. The err field is the *level* of the message.
/var/adm/messages	Is the action field.

Caution – Only use *tabs* as white space in the /etc/syslog.conf file.

The Solaris OE accesses the /usr/include/sys/syslog.h file to determine the correct *facility.level* sequencing order.

Selector Field

The selector field is a semicolon-separated list of priority specifications in the following format:

facility.level;facility.level

In the selector field syntax, *facility* is a system facility. Table 13-1 shows values that the selector field (*facility*) can contain.

Table 13-1 Selector Field (*facility*) Options

Field	Description
kern	Messages generated by the kernel.
user	Messages generated by user processes. This file does not list the default priority for messages from programs or facilities.
mail	The mail system.
daemon	System daemons, such as the in.ftpd and the telnetd daemons.
auth	The authorization system, including the login, su, and ttymon commands.
syslog	Messages generated internally by the syslogd daemon.
lpr	The line printer spooling system, such as the lpr and lpc commands.
news	Files reserved for the USENET network news system.
uucp	The UNIX-to-UNIX copy (UUCP) system does not use the syslog function.
cron	The cron and at facilities, including crontab, at, and cron.
local0-7	Fields reserved for local use.
mark	The time when the message was last saved. The messages are produced internally by the syslogd daemon.
*	All facilities, except the mark *facility*.

Note – You can use the asterisk (*) to select all facilities (for example `*.err`); however, you cannot use `*` to select all levels of a *`facility`* (for example, `kern.*`)

In the selector field syntax, *`level`* is the severity or importance of the message. Each *`level`* includes all the levels above (of a higher severity). Table 13-2 shows the levels in descending order of severity.

Table 13-2 Selector Field (*`level`*) Options

Level	Priority	Description
`emerg`	0	Panic conditions that are normally broadcast to all users
`alert`	1	Conditions that should be corrected immediately, such as a corrupted system database
`crit`	2	Warnings about critical conditions, such as hard device errors
`err`	3	Errors other than hard device errors
`warning`	4	Warning messages
`notice`	5	Non-error conditions that might require special handling
`info`	6	Informational messages
`debug`	7	Messages that are normally used only when debugging a program
`none`	8	Messages are not sent from the indicated *`facility`* to the selected file

Note – Not all levels of severity are implemented for all facilities in the same way. For more information, refer to the online manual pages.

Action Field

The action field defines where to forward the message. This field can have any one of the following entries:

/filename	The targeted file.
@host	The @ sign denotes that messages must be forwarded to a remote host. Messages are forwarded to the syslogd daemon on the remote host.
user1, user2	The user1 and user2 entries receive messages if they are logged in.
*	All logged in users will receive messages when they are logged in.

Note – You must manually create the */filename* file if it does not already exist.

Entries in the /etc/syslog.conf File

The standard /etc/syslog.conf configuration file is:

```
#ident  "@(#)syslog.conf        1.5     98/12/14 SMI"   /* SunOS 5.0 */
#
# Copyright (c) 1991-1998 by Sun Microsystems, Inc.
# All rights reserved.
#
# The syslog configuration file.
#
# This file is processed by m4 so be careful to quote (" ") names
# that match m4 reserved words. Also, within ifdef’s, arguments
# containing commas must be quoted.
#
*.err;kern.notice;auth.notice                   /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit        /var/adm/messages

*.alert;kern.err;daemon.err                     operator
*.alert                                         root

*.emerg                                         *

# If a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
#auth.notice ifdef(‘LOGHOST’,/var/log/authlog, @loghost)

mail.debug   ifdef(‘LOGHOST’,/var/log/syslog, @loghost)

#
# Non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(‘LOGHOST’,,
user.err                                        /dev/sysmsg
user.err                                        /var/adm/messages
user.alert                                       ‘root, operator’
user.emerg                                      *
)
```

The syslogd Daemon and the m4 Macro Processor

Figure 13-3 shows how the syslogd daemon, the m4 macro processor, and the /etc/syslog.conf file interact in conceptual phases to determine the correct message routing.

Process

These conceptual phases are described as:

1. The syslogd daemon runs the m4 macro processor.
2. The m4 processor reads the /etc/syslog.conf file, processes any m4 statements in the input, and passes the output to the syslogd daemon.
3. The syslogd daemon uses the configuration information output by the m4 processor to route messages to the appropriate places.

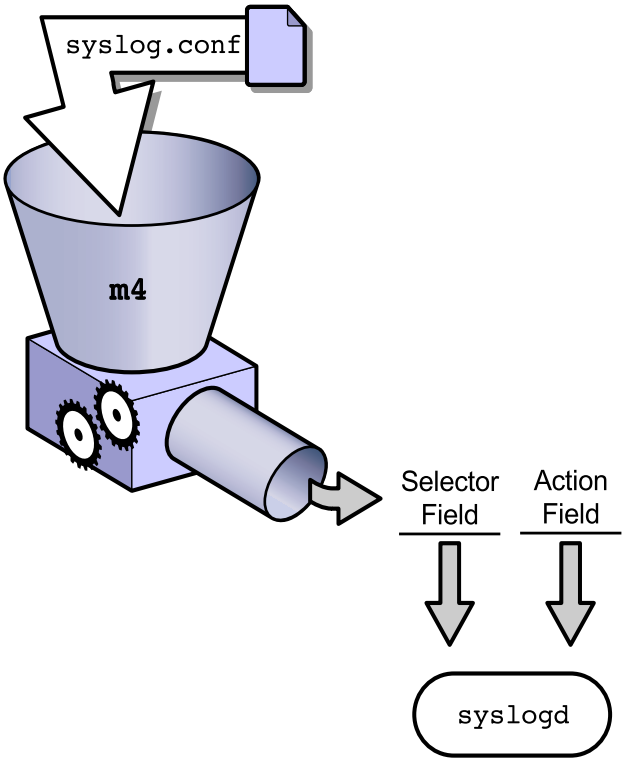


Figure 13-3 The m4 Macro Processor

The syslogd daemon does not read the /etc/syslog.conf file directly. The syslogd daemon obtains its information as follows:

1. The syslogd daemon starts the m4 processor, which parses the /etc/syslog.conf file for m4 commands that it can interpret.
2. If the m4 processor does not recognize any m4 commands on a line, it passes the output back to the syslogd daemon as a two-column output.
3. The syslogd daemon then uses the two-column output to route messages to the appropriate destination.

If the m4 processor encounters an ifdef statement within the /etc/syslog.conf file, the ifdef statement is evaluated for a True or False condition. The message routing then occurs relative to the output of the test.

Operation Phase 1

In the following examples, the syslogd daemon is running on the host1 system. This section contains two examples of the host1 system's /etc/hosts file.

These /etc/hosts file examples are excerpts of the /etc/hosts/ file.

Example A /etc/hosts:

```
192.9.200.1 host1 loghost
192.9.200.2 host2
```

Example B /etc/hosts:

```
192.9.200.1 host1
192.9.200.2 host2 loghost
```

When the syslogd daemon starts at system boot, the syslogd daemon evaluates the /etc/hosts file, and checks the Internet Protocol (IP) address associated with the *hostname* as compared to the IP address associated with loghost.

In Example A, host1 and loghost are both associated with IP address 192.9.200.1. Therefore, the syslogd daemon runs the first command line: /usr/ccs/bin/m4 -D LOGHOST, causing the m4 LOGHOST variable to be defined as TRUE during the parsing of the /etc/syslog.conf file.

In Example B, host1 is associated with IP address 192.9.200.1, while host2 and loghost are both associated with IP address 192.9.200.2. In this example, the syslogd daemon runs the second command line, /usr/ccs/bin/m4 (no -D LOGHOST), causing the m4 LOGHOST variable to be undefined during the parsing of the /etc/syslog.conf file.

Operation Phase 2

In the phase 2, the m4 macro processor parses the /etc/syslog.conf file. For each line that is parsed, the m4 processor searches the line for m4 statements, such as an ifdef statement. If no ifdef statement is encountered on the line, the m4 processor passes the line to the syslogd daemon.

If the m4 processor finds a line with an ifdef statement, the line is evaluated as follows:

- The ifdef ('LOGHOST', truefield, falsefield) command checks to see if the variable LOGHOST is defined.
- If the variable LOGHOST is defined, the entries from the truefield field are used; otherwise, entries from the falsefield field are used.

For example:

```
mail.debug          ifdef('LOGHOST', /var/log/syslog, @loghost)
```

If the variable LOGHOST variable is defined in phase 1, then the m4 processor returns:

```
mail.debug          /var/log/syslog
```

If the LOGHOST variable was evaluated as FALSE in phase 1, then the m4 processor returns:

```
mail.debug          @loghost
```

In either case, the output has an entry in the selector field and an entry in the action field. The m4 processor then passes the output to the syslogd daemon.

Operation Phase 3

For each line parsed in the `/etc/syslog.conf` file from phase 2, the `m4` processor produces output in a two-column field: a selector field and an action field. The output is sent to the `syslogd` daemon, which uses the information to route messages to their appropriate destinations. After the information is configured, the `syslogd` daemon continues to run with this configuration.

Configuring the /etc/syslog.conf File

The target locations for the syslog message files are defined within the /etc/syslog.conf file. You must restart the syslogd daemon whenever you make any changes to this file.

Message Routing

The following excerpt from the /etc/syslog.conf file shows how various events are logged by the system.

```
1   *.err;kern.notice;auth.notice                  /dev/sysmsg
2   *.err;kern.debug;daemon.notice;mail.crit       /var/adm/messages
3   *.alert;kern.err;daemon.err                    operator
4   *.alert                                        root
5   *.emerg                                        *
```

Note – Within the /etc/syslog.conf file, use a selector *level* of err to indicate that all events of priority error (and higher) are logged to the target defined in the action field.

In Line 1, every error event (*.err) and all kernel and authorization *facility* events of *level* notice, which are not error conditions but might require special handling, will write a message to the /dev/sysmsg file.

In Line 2, every error event (*.err), all kernel *facility* events of *level* debug, all daemon *facility* events of *level* notice, and all critical *level* mail events will record a message in the /var/adm/messages file. Therefore, errors are logged to both files.

Line 3 indicates that all alert *level* events, including the kernel error *level* and daemon error *level* events, are sent to the user operator if this user is logged in.

Line 4 indicates that all alert *level* events are sent to the root user if the root user is logged in.

Line 5 indicates that any event that the system interprets as an emergency will be logged to the terminal of every logged-in user.

To alter the event logging mechanism, edit the /etc/syslog.conf file, and restart the syslogd daemon.

Stopping and Starting the syslogd Daemon

The syslogd daemon can be started automatically during boot or manually from the command line.

Starting the syslogd Daemon During Boot Operation

The /etc/rc2.d/S74syslog file starts the syslogd process during each system boot.

The /etc/syslog.conf configuration file is read each time the syslogd daemon starts.

Manually Stopping and Starting the syslogd Daemon

If the configuration file has been modified, you can manually stop or start syslogd daemon, or send it a HUP signal, which causes the daemon to reread the /etc/syslog.conf file.

To stop the syslogd daemon, perform the command:

```
# /etc/init.d/syslog stop
```

To start the syslogd daemon, perform the command:

```
# /etc/init.d/syslog start
```

To send a HUP signal to the syslogd daemon, perform the command:

```
# pkill -HUP syslogd
```

Configuring syslog Messaging

The inetd daemon uses the syslog command to record incoming network connection requests made by using Transmission Control Protocol (TCP).

Enabling TCP Tracing

The inetd daemon is the network listener process for many network services. The inetd daemon listens for service requests on the TCP and User Datagram Protocol (UDP) ports associated with each of the services listed in the inetd configuration file. When a request arrives, the inetd daemon executes the server program associated with the service. You can modify the behavior of the inetd daemon to log TCP connections by using the syslogd daemon.

The following online manual page excerpt for the inetd daemon shows that only the daemon *facility* and the notice message *level* are supported:

```
# man inetd
Maintenance Commands                                    inetd(1M)

NAME
     inetd - Internet services daemon
...
...
-t       Instructs inetd to trace the incoming connections for all of
its TCP services. It does this by logging the client's IP address and TCP
port number, along with the name of the service, using the syslog(3)
facility. UDP services can not be traced. When tracing is enabled, inetd
uses the syslog facility code "daemon" and "notice" priority level.
```

Note – The Internet daemon inetd provides services for many network protocols, including the Telnet and File Transfer Protocol (FTP) protocols.

You must enable the trace option for the `inetd` daemon to send messages to the `syslogd` daemon. In other words, use the `-t` option as an argument to the `inetd` daemon to enable tracing of TCP services. When you enable the trace option for the `inetd` daemon, it uses the `syslog` *`facility`* to log the client's IP address and TCP port number, and the name of the service. To enable tracing TCP connections automatically at boot time, add the `-t` option to the entry which activates the `inetd` daemon in the `inetsvc` script located in the `/etc/init.d` directory.

The modified entry looks similar to the following:

```
# grep inetd /etc/init.d/inetsvc
/usr/sbin/inetd -s -t &
```

Note – You must restart the `inetd` daemon for the new option to take effect.

In the previous example, the `/etc/syslog.conf` file configures the `syslogd` daemon so that it selectively distributes the messages sent to it from the `inetd` daemon.

```
# grep daemon.notice /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit  /var/adm/messages
```

All daemon messages of *`level`* `notice` or higher are sent to the `/var/adm/messages` file due to the `daemon.notice` entry in the `/etc/syslog.conf` file.

Note – The `/var/adm/messages` file must exist. If it does not exist, create it, and then stop and start the `syslogd` daemon, or messages will not be written to the file.

Monitoring a syslog File in Real Time

You can monitor the designated `syslog` file, in the `/var/adm` directory, in real time using the command `tail -f`. The `tail -f` command holds the file open so that you can view messages being written to the file by the `syslogd` daemon.

Viewing Messages In Real Time

To view messages sent to the /var/adm/messages file, perform the command:

```
# tail -f /var/adm/messages
```

Figure 13-4 shows the log entry generated by a telnet request to system host1 from IP address 192.9.200.1 on Port 45800. Table 13-3 lists each field in this figure and its corresponding result.

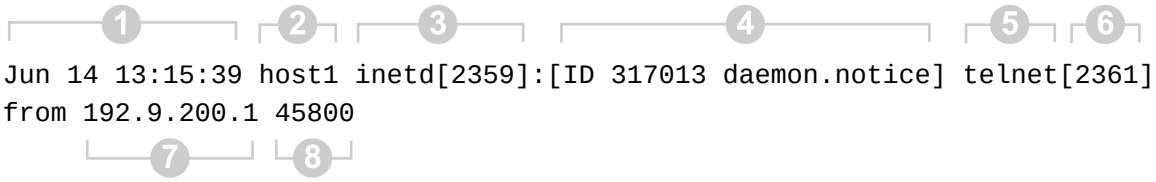


Figure 13-4 The syslogd Daemon Logged Entry

Table 13-3 The syslogd Logged Entry Description

Number	Field	Result
1	Date/time	Jun 14 13:15:39
2	Local host name	host1
3	Process name/PID number	inetd[2359]
4	MsgID number/ selector *facility.level*	[ID 317013 daemon.notice]
5	Incoming request	telnet
6	PPID number	[2361]
7	IP address	192.9.200.1
8	Port number	45800

To exit the /var/adm/messages file, press Control-C.

Note – Should any unusual activity occur, use scripts to automatically parse the log files, and then send the information to support personnel.

Adding One-Line Entries to a System Log File

The `logger` command enables you to send messages to the `syslogd` daemon.

The syntax of the `logger` command is:

```
logger [ -i ] [ -f file ] [ -p priority ] [ -t tag ] [ message ]
```

where:

`-i`	Logs the process ID of the `logger` command with each line
`-f` *`file`*	Uses the contents of *`file`* as the message to log (*`file`* must exist)
`-p` *`priority`*	Enters the message with the specified *`priority`*
`-t` *`tag`*	Marks each line added to the log file with the specified *`tag`*
`message`	Concatenates the string arguments of the message in the order specified, separated by single-space characters

You can specify the message priority as a *`facility.level`* pair. For example, `-p local3.info` assigns the message priority of the `info` *`level`* in the `local3` *`facility`*. The default priority is `user.notice`.

Therefore, the following example logs the message `System rebooted` to the `syslogd` daemon, using the default priority *`level`* `notice` and the *`facility`* `user`:

```
# logger System rebooted
```

If the user.notice selector field is configured in the /etc/syslog.conf file, the message is logged to the file designated for the user.notice selector field. If the user.notice selector field is not configured in the /etc/syslog.conf file, you can either add the user.notice selector field to the /etc/syslog.conf file, or you can prioritize the output as follows:

```
# logger -p user.err System rebooted
```

Changing the priority of the message to user.err routes the message to the /var/adm/messages file as indicated in the /etc/syslog.conf file.

A message priority can also be specified numerically. For example, logger -i -p2 "crit" creates an entry in the message log that identifies the user.crit-facility.level pair as follows:

```
Nov 3 09:49:34 hostname root[2838]: [ID 702911 user.crit] crit
```

Using the Solaris Management Console Log Viewer

You can use the Solaris Management Console Log Viewer application to view `syslog` message files. You can also use this application to view and capture information from the Management Tool logs.

Opening the Solaris Management Console Log Viewer

To open the viewer, perform the following steps:

1. Use the `smc` command to open the Solaris Management Console:

```
# smc &
```

   The Solaris Management Console application launches.

2. Select This Computer (*`hostname`*).
3. Select System Status.
4. Select Log Viewer.

   The initial Log Viewer is displayed, as shown in Figure 13-5.

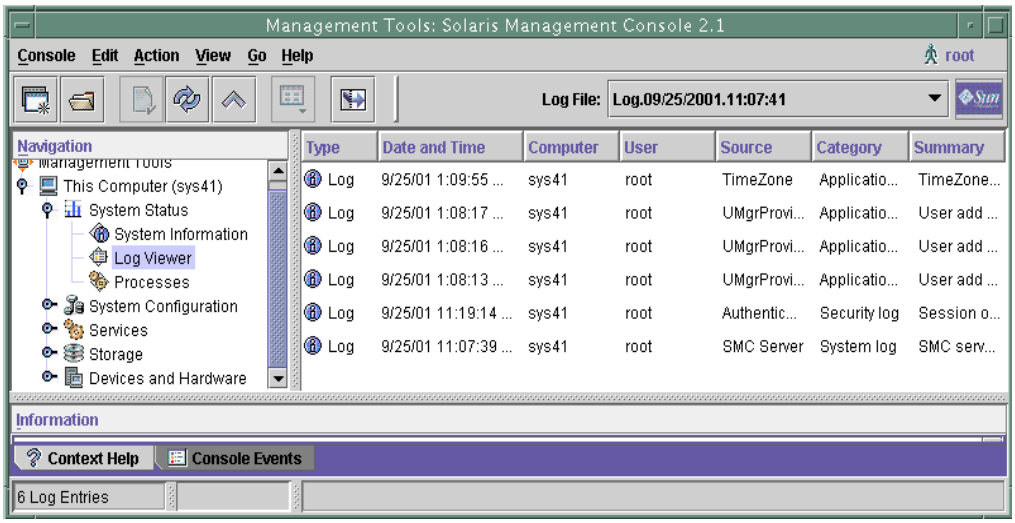


Figure 13-5 Solaris Management Console – Log Viewer

The initial Log Viewer display lists Management Tools log entries from the `/var/sadm/wbem/log` directory.

Viewing a syslog Message File

To select Log files, use the Log File pull-down menu located on the icon bar of the Log Viewer window. Figure 13-6 shows that the Log File pull-down menu lists both the `wbem_log` files that record Solaris Management Console activity and the `syslog` message logs named `/var/log/syslog` and `/var/adm/messages`.

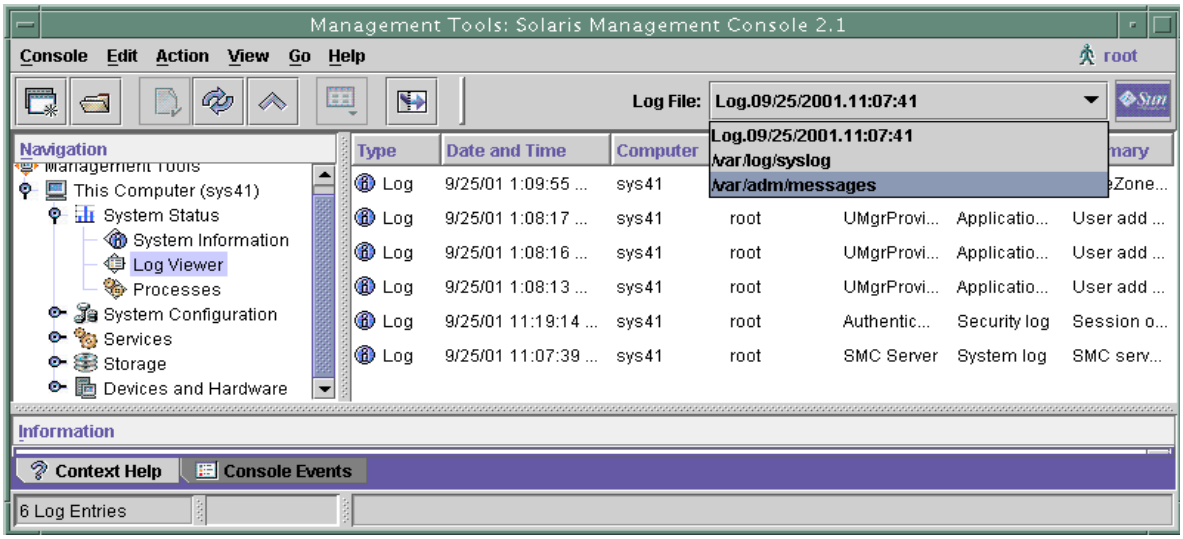


Figure 13-6 List of Log Files

To view a `syslog` messages log, perform the following steps:

1. Click the down arrow icon in the Log Files selection box.
2. Select the `/var/adm/messages` log that you want to view.

   The selected message log appears in the Solaris Management Console View pane, as shown in Figure 13-7 on page 13-21.

Note – You cannot manipulate the `syslog` message logs. You can only view them chronologically as they were created.

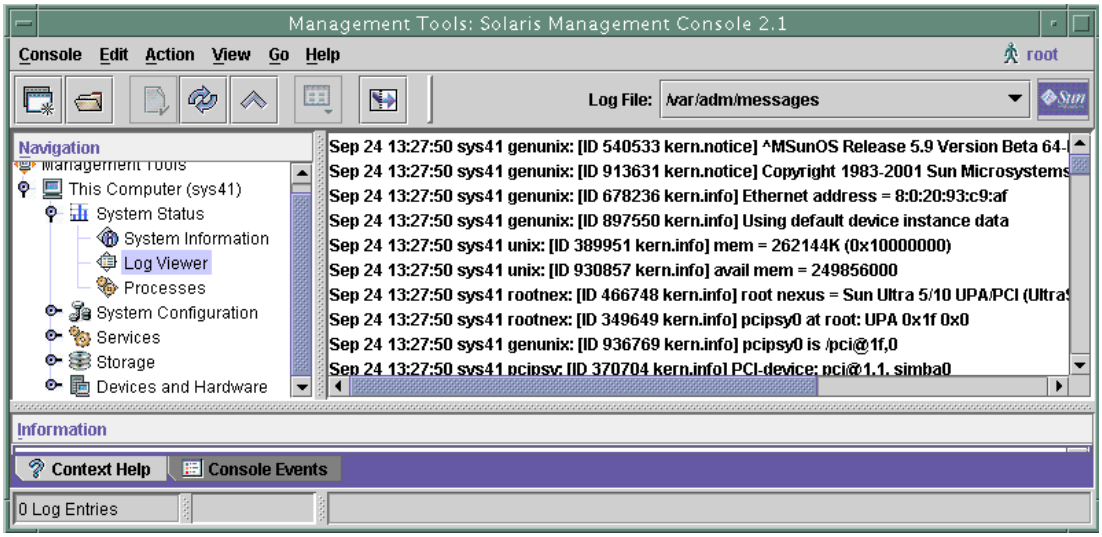


Figure 13-7 Display of the `syslog` Generated Message File

Note – You can sort and filter the message logs by using command-line sorting and filtering tools, such as the `sort` and `grep` commands.

Viewing a Management Tools Log File

When you view the `syslog` messages files, you can only use the Open Log Files or the Log Files Settings functions in the Action menu, as shown in Figure 13-8.

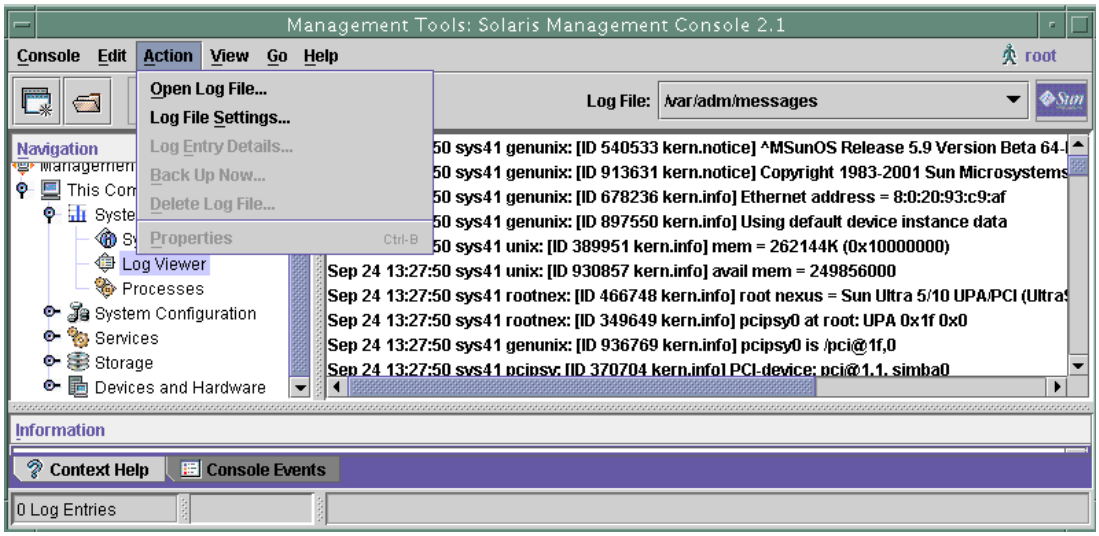


Figure 13-8 Action Menu

Select Open Log Files from the Action menu to display the Open Log Files window. The Open Log Files window contains the same list of log files displayed by the Log Viewer Log File pull-down menu. To view the log files associated with the Solaris Management Console, you must load one of the `wbem_log` files.

The `wbem_log` files exist, by default, in the `/var/sadm/wbem/log` directory. The most recent log is named `wbem_log`.

To open the `wbem_log` file, select the *`LogMM/DD/YEAR.HH:MM:SS`* file, and then click Open, as shown in Figure 13-9.

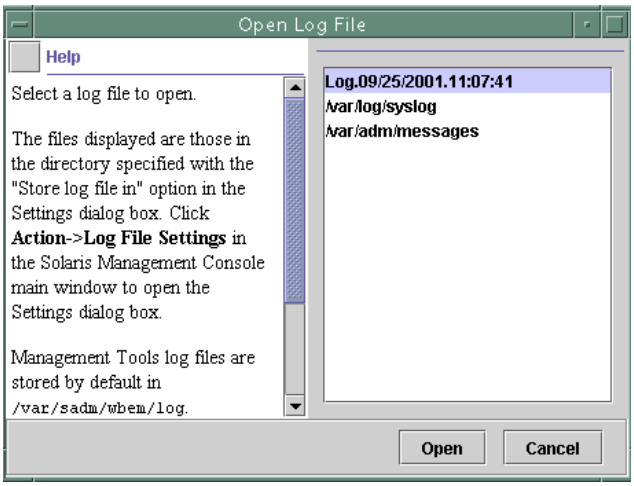


Figure 13-9 Action Menu Open Log File Window

The log file in Figure 13-9 is named `Log09/25/2001.11:07:41`, which indicates the log file creation date and time.

The Log Viewer lets you view and manage log files for Solaris Management Console tools and events. For example, log entries are generated for session open, session close, authentication success, and authentication failure events.

You can also use the log view to *select* specific events, as shown in Figure 13-10. To *view* specific events, select an option from the View menu.

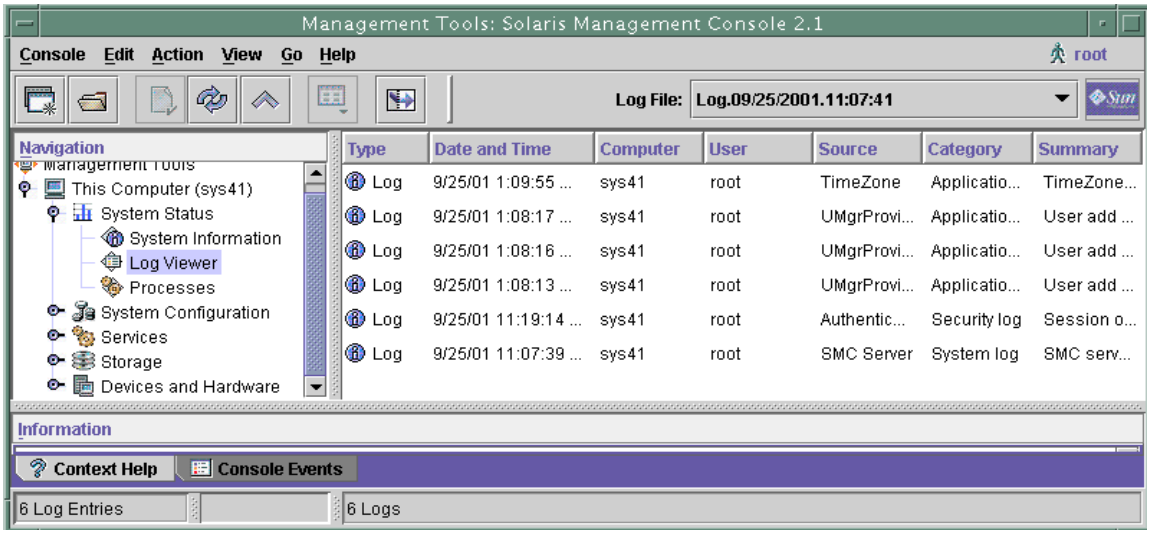


Figure 13-10 Display of `wbem_log` Generated Message File

Browsing the Contents of a Management Tools Log File

The Filter option in the View menu lets you filter out unwanted logged events to help you establish pattern recognition scenarios, which are helpful when troubleshooting system irregularities.

Select Filter from the View menu to open the Log Filter window, as shown in Figure 13-11.

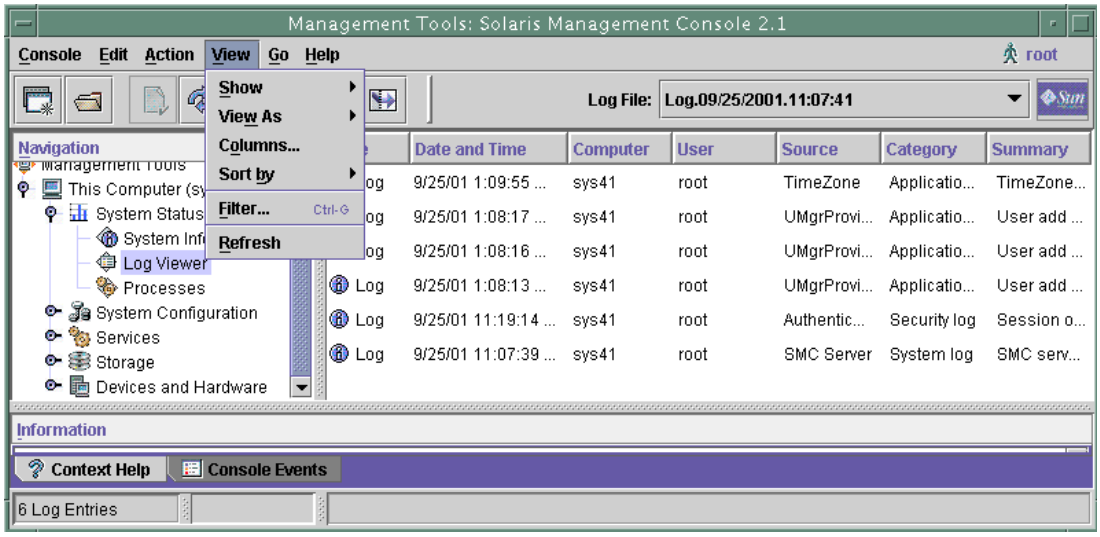


Figure 13-11 View Menu

The Log Filter window, as shown in Figure 13-12 on page 13-26, enables you to narrow the logged event report based on:

- The date and time that the log entries start and stop
- Log properties:
  - Type – Logged events, which include informational, warning, or error events
  - Identification – Logged events created by a specific user or system
  - Category – The event generation source, such as an application, the system, or security event

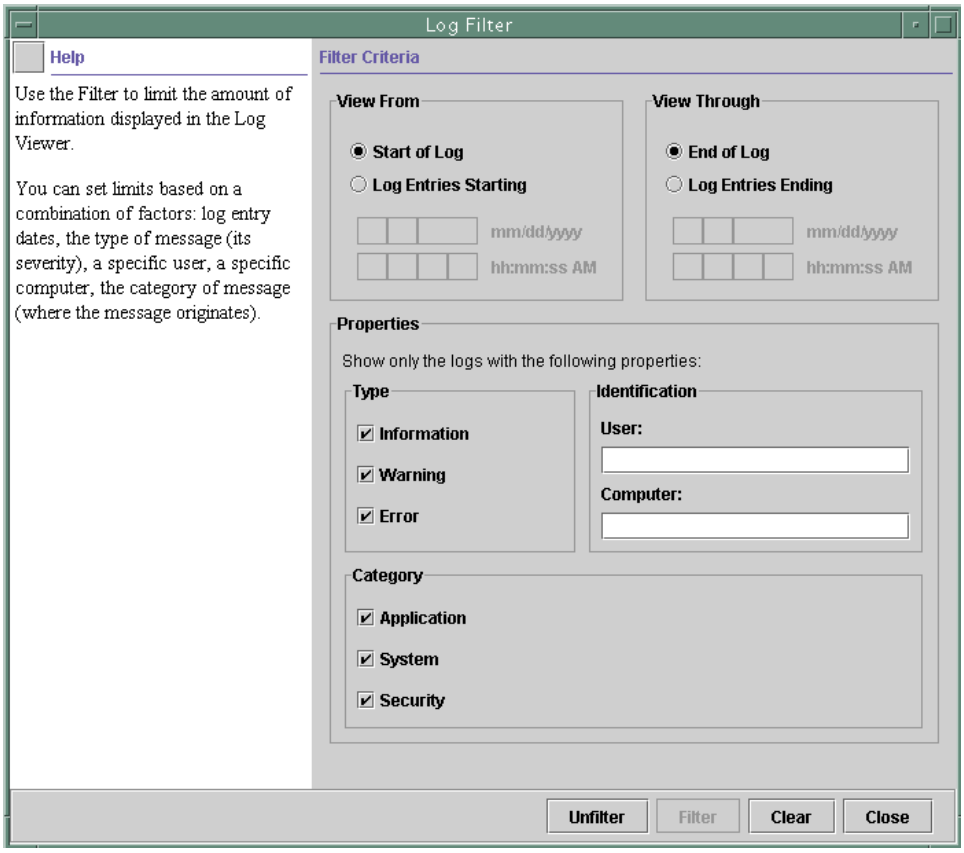


Figure 13-12 Log Filter Window

The Log Viewer then filters the selected log file. Figure 13-12 shows that the selected Log File is identified in the Log File box on the Log Viewer icon bar. You can reload your display to show only the events that fit the filtered criteria

To return to the Log Viewer, click Close.

Displaying Management Tools Log Entry Details

The Log Viewer shows an overview of the logged event's details. To view more specific details of the logged event, double-click a specific log entry in the list.

Figure 13-13 shows the Log Viewer window. The bold column headings in the View pane identify and display the contents of the fields that are contained in the log file.

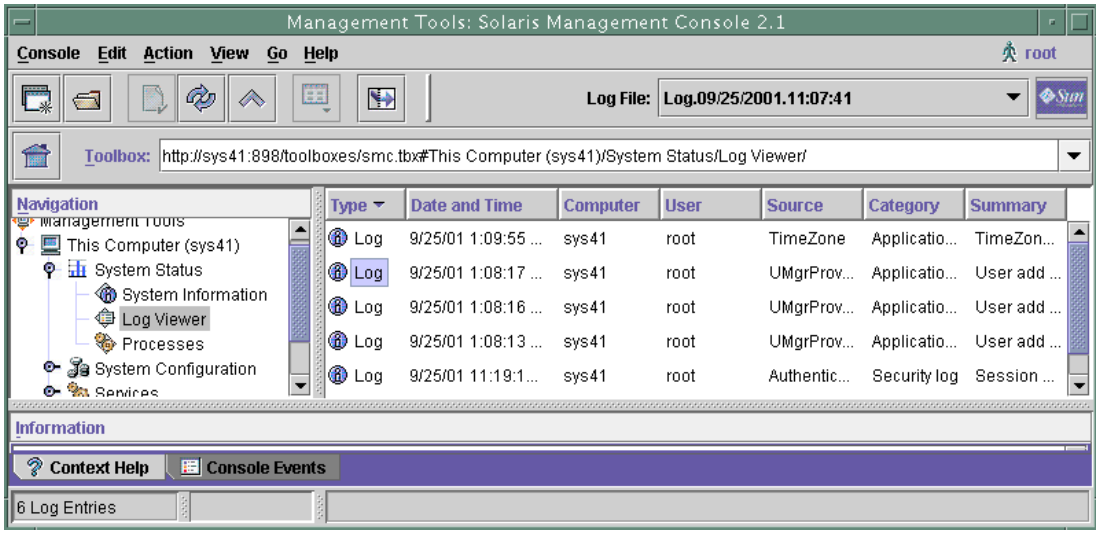


Figure 13-13 Log Viewer Window

The Log Entry Details Window, as shown in Figure 13-14, enables you to select details about the selected logged event, and enables you to navigate to the next and previous event as follows:

- Click the down arrow to select the next logged event.
- Click the up arrow to select the previous logged event.

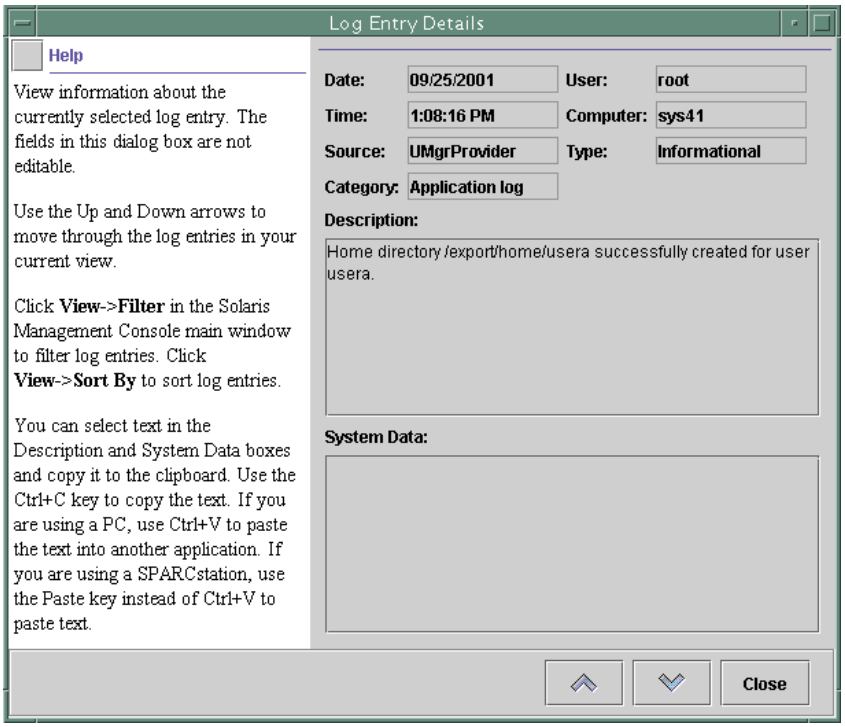


Figure 13-14 Log Entry Details Window

To return to the Log Viewer window, click Close.

Backing Up Management Tools Log File

You can back up the `wbem_log` files at a predefined time interval or when they reach a predefined size limit.

To force a backup of the `wbem_log`:

1. Select Back Up Now from the Action menu, as shown in Figure 13-15.

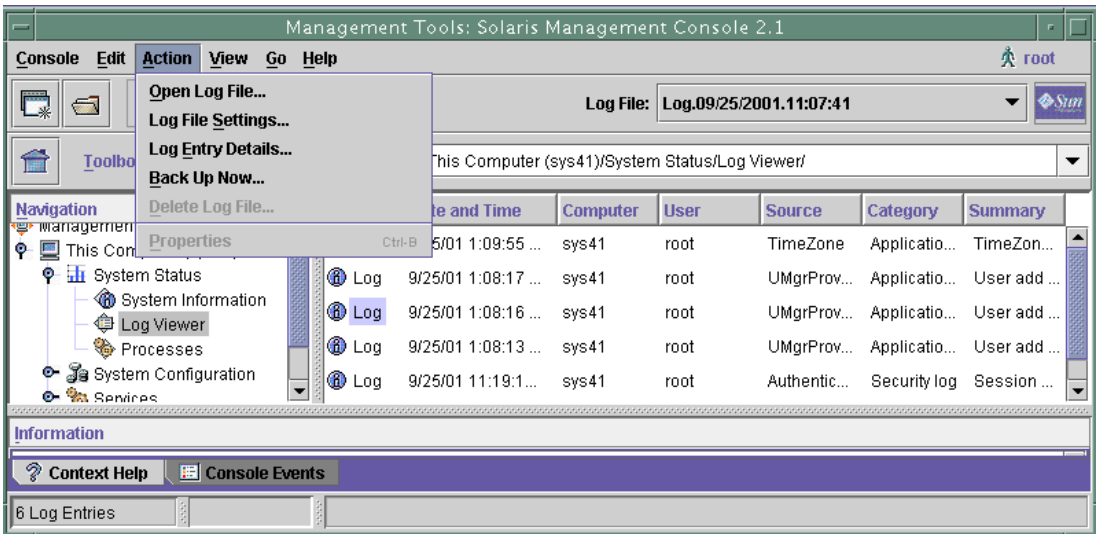


Figure 13-15 Backup of `wbem_log` Generated Message File

2. A new window appears, as shown in Figure 13-16, warning you that the existing log will be renamed.

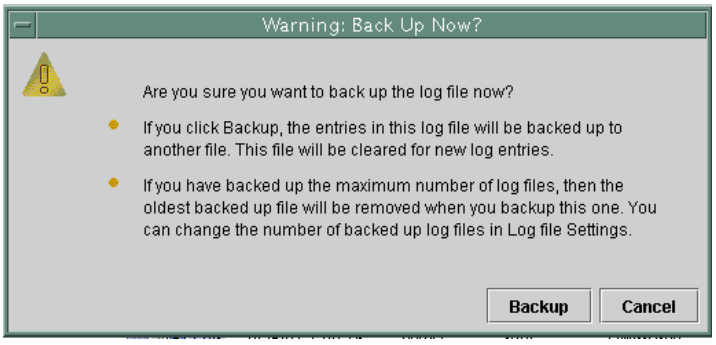


Figure 13-16 Warning: Back Up Now Window

Caution – If you have reached the maximum number of archive copies and you want to keep the oldest archived log, copy the log before you continue with the backup procedure.

3. Click Backup to continue.

   The current log is renamed to reflect the current date and time. Subsequent entries are recorded in the current `wbem_log` file. The New `wbem_log` Generated Message File window, as shown in Figure 13-17, shows that the old log has moved to `wbem_log.1`, and that the Log Viewer display is clear.

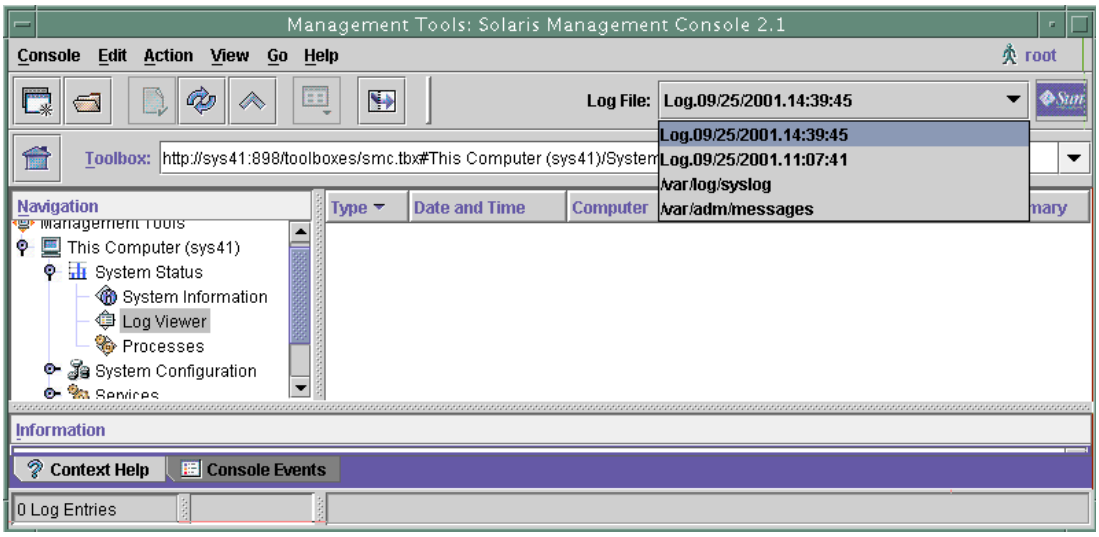


Figure 13-17 New `wbem_log` Generated Message File

4. Select Log File Settings from the Action menu to modify the automatic backup configuration setting on any selected log file, as shown in Figure 13-18.

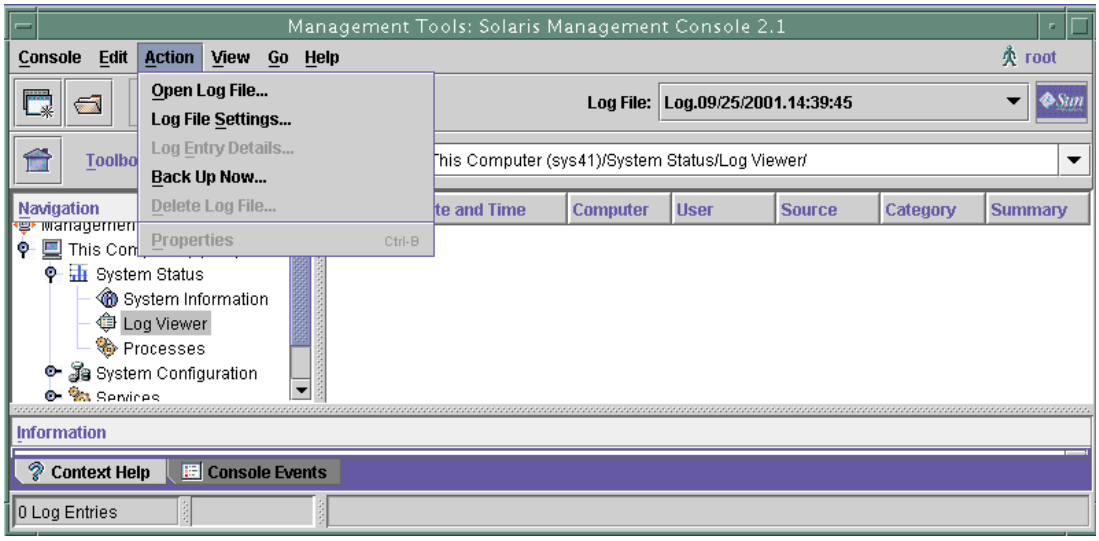


Figure 13-18 Action Menu – Log File Settings

5. In the Log File Settings window (shown in Figure 13-19):
   a. Specify an alternate directory in which to store the `wbem_log` files.
   b. Modify the maximum log file size.
   c. Specify how many backed up `wbem_log` files to maintain.
   d. Enable or disable system logging.

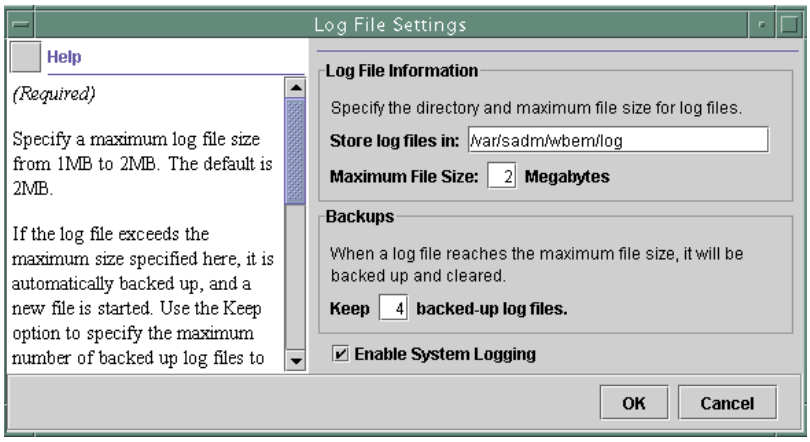


Figure 13-19 Log File Settings Window

6. Do one of the following actions:
   a. Click Cancel to return to the Log Viewer window.
   b. Click OK to accept any changes.

7. To exit the Log Viewer application window, select Exit from the Console menu, as shown in Figure 13-20.

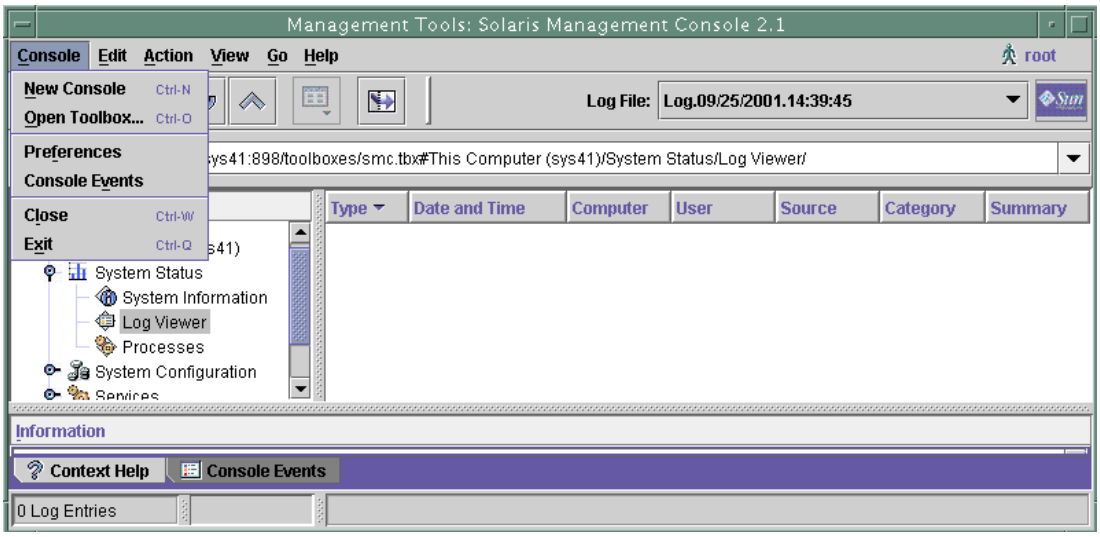


Figure 13-20 Console Menu – Exit

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Using the syslog Function and Auditing Utilities (Level 1)

In this lab, you use the `syslog` function to log messages locally and remotely.

Preparation

This exercise requires installed manual (man) pages and two systems that list each other in the `/etc/hosts` file. Verify that the `CONSOLE` variable is commented out in the `/etc/default/login` file on both systems. Except as noted otherwise, perform all steps on both systems. Refer to the lecture notes as necessary to perform the steps listed.

Tasks

Perform the following tasks:

- Make a backup copy of the `/etc/syslog.conf` file. Use the `tail` command to observe messages as they appear in the `/var/adm/messages` file. Modify the `/etc/init.d/inetsvc` file to enable message tracing. Verify that using the `telnet` command generates messages that appear in the log file.
- Add an entry to the `/etc/syslog.conf` file that would send `local0.notice` messages to the `/var/log/local0.log` file. Create a `/var/log/local0.log` file. Use the `tail` command to monitor `/var/log/local0.log`. Use the `logger` command to send messages from the `local0` *`facility`* at different levels. Verify that messages arrive in the `/var/log/local0.log` file. Send multiple, identical `local0` messages, followed by a different `local0` message, and observe the results in the `/var/log/local0.log` file.
- Designate one system as *`system1`* and the other as *`system2`*. On *`system1`*, modify the `logger0.notice` entry in the `/etc/syslog.conf` file so that it sends messages to *`system2`*, and send a `HUP` signal to the `syslogd` daemon. On *`system2`*, use the `tail` command to monitor the `/var/log/local0.log` file. On *`system1`*, send a `local0.notice` message using the `logger` command. Observe the results on *`system2`*.

- On both systems, uncomment the `auth.notice` entry in the `/etc/syslog.conf` file, and send a `HUP` signal to the `syslogd` daemon. Verify that both systems are listed in the `/etc/hosts` file, and identify which one is associated with the `loghost` alias in each file. On both systems, use the `m4` processor with and without the `-D LOGHOST` option, and record the output for the `auth.notice` entry.
- On both systems, use the `tail` command to monitor the `/var/log/authlog` file. On *system2*, perform a remote login (`rlogin`) to the same system. Check the output from the `tail` command on both systems. Exit the `rlogin` session. On *system2*, make a backup copy of the `/etc/hosts` file. On *system2*, edit the `/etc/hosts` file so that the `loghost` alias is associated with *system1*. Repeat the `rlogin` session, and observe the output from the `tail` command on both systems.
- On *system2*, restore the original `/etc/hosts` file. On both systems, stop all `tail` commands, restore the original `/etc/syslog.conf` files, and send a `HUP` signal to the `syslogd` daemon.

Exercise: Using the syslog Function and Auditing Utilities (Level 2)

In this lab, you use the `syslog` function to log messages locally and remotely.

Preparation

This exercise requires installed manual (man) pages and two systems that list each other in the `/etc/hosts` file. Verify that the `CONSOLE` variable is commented out in the `/etc/default/login` file on both systems. Except as noted otherwise, perform all steps on both systems. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

Complete the following steps:

- Make a backup copy of the `/etc/syslog.conf` file. Use the `tail` command to observe messages as they appear in the `/var/adm/messages` file. Modify the `/etc/init.d/inetsvc` file to enable message tracing. Verify that using the `telnet` command generates messages that appear in the log file.
- Add an entry to the `/etc/syslog.conf` file that would send `local0.notice` messages to the `/var/log/local0.log` file. Create a `/var/log/local0.log` file. Use the `tail` command to monitor `/var/log/local0.log`. Use the `logger` command to send messages from the `local0` *`facility`* at different levels. Verify that messages arrive in the `/var/log/local0.log` file. Send multiple, identical `local0` messages, followed by a different `local0` message, and observe the results in the `/var/log/local0.log` file.
- Designate one system as *`system1`* and the other as *`system2`*. On *`system1`*, modify the `logger0.notice` entry in the `/etc/syslog.conf` file so that it sends messages to *`system2`*, and send a `HUP` signal to the `syslogd` daemon. On *`system2`*, use the `tail` command to monitor the `/var/log/local0.log` file. On *`system1`*, send a `local0.notice` message using the `logger` command. Observe the results on *`system2`*.

- On both systems, uncomment the `auth.notice` entry in the `/etc/syslog.conf` file, and send a `HUP` signal to the `syslogd` daemon. Verify that both systems are listed in the `/etc/hosts` file, and identify which one is associated with the `loghost` alias in each file. On both systems, use the `m4` processor with and without the `-D LOGHOST` option and record the output for the `auth.notice` entry.
- On both systems, use the `tail` command to monitor the `/var/log/authlog` file. On *system2*, perform a remote login (`rlogin`) to the same system. Check the output from the `tail` command on both systems. Exit the `rlogin` session. On *system2*, make a backup copy of the `/etc/hosts` file. On *system2*, edit the `/etc/hosts` file so that the `loghost` alias is associated with *system1*. Repeat the `rlogin` session, and observe the output from the `tail` command on both systems.
- On *system2*, restore the original `/etc/hosts` file. On both systems, stop all `tail` commands, restore the original `/etc/syslog.conf` files, and send a `HUP` signal to the `syslogd` daemon.

Tasks

Perform the following tasks.

Task 1 – Enabling and Logging `inetd` Trace Messages

Complete the following steps:

1. Change the directory to `/etc`, and create a backup copy of the `/etc/syslog.conf` file.
2. Display the man page for the `inetd` process, and verify the *`facility`* and *`level`* used by the `inetd` process when you run the process with the `-t` option.

   Which *`facility`* and *`level`* pair is the `inetd` daemon using?

   _____

3. Examine the `/etc/syslog.conf` file, and determine if the `syslogd` daemon would recognize `inetd` tracing messages.

   If it does recognize `inetd` messages, where would it send the messages?

   _____

   Are `inetd` tracing messages recognized by the `syslogd` daemon (yes or no)?

   _____

   To what destination will the `syslogd` daemon send the messages?

   _____

4. Open a new terminal window, and use the `tail` command to view new entries as they are recorded in the `/var/adm/messages` file.
5. In an available window, use the `telnet` command to connect to your own system. Exit the `telnet` session after you successfully log in.
6. Observe the window in which you are running the `tail` command. Do any new `telnet`-related messages appear in the `/var/adm/messages` file (yes or no)?

   _____

7. Edit the `/etc/init.d/inetsvc` file, and change the line that reads:

```
/usr/sbin/inetd -s &
```

   so that it now reads:

```
/usr/sbin/inetd -s -t &
```

8. To enable connection logging, edit the `/etc/default/inetd` file by setting the following field as:

```
ENABLE_CONNECTION_LOGGING=YES
```

9. Stop the `inetd` process, check that it is stopped, and then restart it. Verify that the `inetd` daemon is running with the `-t` option.
10. Repeat Step 5 and Step 6. Do any new `telnet`-related messages appear in the `/var/adm/messages` file? If yes, list them.

    _____

Task 2 – Using the logger Command to Demonstrate How Levels Operate

Complete the following steps:

1. Edit the /etc/syslog.conf file so that it includes the following line:

```
local0.notice <TAB> /var/log/local0.log
```

2. Create a file called /var/log/local0.log.
3. Cause the syslogd daemon to reread the /etc/syslog.conf file by sending it a HUP signal.
4. In the window in which the tail command is running, stop the tail process. Restart the tail command so that it displays the end of the /var/log/local0.log file.
5. In an available window, use the logger utility to send a message using the local0 *facility* and the notice *level*.

   What, if any, new messages does the tail command display?

   _____

6. In an available window, use the logger command to send a message by using the local0 *facility* and the crit *level*.

   What, if any, new messages does the tail command display?

   _____

7. Run the logger command from Step 5 three times. Examine the output from the tail command in the other window. How many new messages appear in the /var/log/local0.log file?
8. Run the logger command once.

   Which new messages appear in the /var/log/local0.log file?

   _____

9. Stop the tail command in the window where it is running.

Task 3 – Logging Messages to Another System

Complete the following steps:

Note – This step does not require you to change host names. In the following steps, substitute the appropriate host name for *system1* and *system2*.

1. On *system1*, edit the `/etc/syslog.conf` file, and change the line for `local0.notice` so that it reads as follows:

```
local0.notice<TAB>@system2
```

2. On *system1*, cause the `syslogd` daemon to reread the `/etc/syslog.conf` file using a `HUP` signal.
3. On *system2*, open a new terminal window, and use the `tail` command to view new entries as they arrive in the `/var/log/local0.log` file.
4. On *system1*, use the `logger` command to generate a message by using the `local0.notice` *facility* and *level* pair.
5. On *system2*, which message is displayed in the window running the `tail` command?

   _____

6. After verifying that *system1* has successfully passed messages to *system2*, stop the `tail` command on *system2*.

Task 4 – Logging Messages by Using the `loghost` Alias and `ifdef` Statements

Complete the following steps:

1. On both systems, edit the `/etc/syslog.conf` file, and uncomment the line that identifies `auth.notice` messages.

```
auth.noticeifdef('LOGHOST', /var/log/authlog, @loghost)
```

   Which two destinations are possible for these messages?

   _____

2. On both systems, examine the `/etc/inet/hosts` file, and identify the name of the host associated with the `loghost` alias.
3. On both systems, cause the `syslogd` daemon to reread the `/etc/syslog.conf` file by sending it a `HUP` signal.

4. On both systems, run the following `m4` commands, and record the line for `auth.notice` messages.

```
# /usr/ccs/bin/m4 -D LOGHOST /etc/syslog.conf
auth.notice /var/log/authlog
# /usr/ccs/bin/m4 /etc/syslog.conf
auth.notice @loghost
```

5. On both systems, open a terminal window, and use the `tail` command to view new entries as they arrive in the `/var/log/authlog` file.

6. On *system2,* use the `rlogin` command to log in to your own system, and then exit the connection.

   On *system2,* which message is displayed in the window running the `tail` command?

   _____

   On *system1,* does a new message display in the window running the `tail` command (yes or no)?

   _____

7. On *system2,* change to the `/etc/inet` directory, and make a backup copy of the `/etc/inet/hosts` file. Edit the `/etc/inet/hosts` file to remove the `loghost` alias from the entry for *system2,* and add it to the entry for *system1*.

8. On *system2,* force the `syslogd` daemon to reread the `/etc/syslog.conf` file by sending it a `HUP` signal.

9. On *system2,* use the `rlogin` command to log in to your own system, and then exit the connection.

   On *system2,* does a new message display in the window running the `tail` command (yes or no)?

   _____

   On *system1,* which message is displayed in the window running the `tail` command?

   _____

Task 5 – Completing the Exercise

Complete the following steps:

1. On both systems, stop the `tail` command in any window where it is running.
2. On *system2*, replace the `/etc/inet/hosts` file with the backup copy you made earlier.
3. On both systems, replace the `/etc/syslog.conf` file with the backup copy you made earlier.
4. On both systems, ensure that the `syslogd` daemon rereads the `/etc/syslog.conf` file by sending it a `HUP` signal.

Exercise: Using the syslog Function and Auditing Utilities (Level 3)

In this lab, you use the `syslog` function to log messages locally and remotely.

Preparation

This exercise requires installed manual (man) pages and two systems that list each other in the `/etc/hosts` file. Verify that the `CONSOLE` variable is commented out in the `/etc/default/login` file on both systems. Except as noted otherwise, perform all steps on both systems. Refer to the lecture notes as necessary to perform the steps listed.

Task Summary

Perform the following tasks:

- Make a backup copy of the `/etc/syslog.conf` file. Use the `tail` command to observe messages as they appear in the `/var/adm/messages` file. Modify the `/etc/init.d/inetsvc` file to enable message tracing. Verify that using the `telnet` command generates messages that appear in the log file.
- Add an entry to the `/etc/syslog.conf` file that would send `local0.notice` messages to the `/var/log/local0.log` file. Create a `/var/log/local0.log` file. Use the `tail` command to monitor `/var/log/local0.log`. Use the `logger` command to send messages from the `local0` *`facility`* at different levels. Verify that messages arrive in the `/var/log/local0.log` file. Send multiple, identical `local0` messages, followed by a different `local0` message, and observe the results in the `/var/log/local0.log` file.
- Designate one system as *`system1`* and the other as *`system2`*. On *`system1`*, modify the `logger0.notice` entry in the `/etc/syslog.conf` file so that it sends messages to *`system2`*, and send a `HUP` signal to the `syslogd` daemon. On *`system2`*, use the `tail` command to monitor the `/var/log/local0.log` file. On *`system1`*, send a `local0.notice` message using the `logger` command. Observe the results on *`system2`*.

- On both systems, uncomment the `auth.notice` entry in the `/etc/syslog.conf` file, and send a `HUP` signal to the `syslogd` daemon. Verify that both systems are listed in the `/etc/hosts` file, and identify which one is associated with the `loghost` alias in each file. On both systems, use the `m4` processor with and without the `-D LOGHOST` option, and record the output for the `auth.notice` entry.
- On both systems, use the `tail` command to monitor the `/var/log/authlog` file. On *system2*, perform a remote login (`rlogin`) to the same system. Check the output from the `tail` command on both systems. Exit the `rlogin` session. On *system2*, make a backup copy of the `/etc/hosts` file. On *system2*, edit the `/etc/hosts` file so that the `loghost` alias is associated with *system1*. Repeat the `rlogin` session, and observe the output from the `tail` command on both systems.
- On *system2*, restore the original `/etc/hosts` file. On both systems, stop all `tail` commands, restore the original `/etc/syslog.conf` files, and send a `HUP` signal to the `syslogd` daemon.

Tasks and Solutions

The following section lists the tasks you must perform and the solutions to these tasks.

Task 1 – Enabling and Logging `inetd` Trace Messages

Complete the following steps:

1. Change the directory to `/etc`, and create a backup copy of the `/etc/syslog.conf` file.

```
# cd /etc
# cp syslog.conf syslog.conf.bak
```

2. Display the man page for the `inetd` process, and verify the *`facility`* and *`level`* used by the `inetd` process when you run the process with the `-t` option.

```
# man inetd
```

Which *`facility`* and *`level`* pair is the `inetd` daemon using?

```
daemon.notice
```

3. Examine the /etc/syslog.conf file, and determine if the syslogd daemon would recognize inetd tracing messages.

   If it does recognize inetd messages, where would it send the messages?

```
# more /etc/syslog.conf
```

   Are inetd tracing messages recognized by the syslogd daemon (yes or no)?

   Yes

   To what destination will the syslogd daemon send the messages?

   The /var/adm/messages *file.*

4. Open a new terminal window, and use the tail command to view new entries as they are recorded in the /var/adm/messages file.

```
# tail -f /var/adm/messages
```

5. In an available window, use the telnet command to connect to your own system. Exit the telnet session after you successfully log in.

```
# telnet host
Trying nnn.nnn.nnn.nnn...
Connected to host.
Escape character is '^]'.

SunOS 5.9

login: root
Password: xxxxxx
Last login: Fri Mar 30 13:55:55 from 10.1.1.100
Sun Microsystems Inc.   SunOS 5.9 581-54 January 2002
# exit
```

6. Observe the window in which you are running the tail command. Do any new telnet-related messages appear in the /var/adm/messages file (yes or no)?

   Before starting inetd *with* -t*, no.*

7. Edit the /etc/init.d/inetsvc file, and change the line that reads:

```
/usr/sbin/inetd -s &
```

   so that it now reads:

```
/usr/sbin/inetd -s -t &
```

8. To enable connection logging, edit the /etc/default/inetd file by setting the following field as:

```
ENABLE_CONNECTION_LOGGING=YES
```

9. Stop the inetd process, check that it is stopped, and then restart it. Verify that the inetd daemon is running with the -t option.

```
# /etc/init.d/inetsvc stop
# pgrep -l inetd
# /etc/init.d/inetsvc start
# pgrep -lf inetd
```

10. Repeat Step 5 and Step 6. Do any new telnet-related messages appear in the /var/adm/messages file? If yes, list them.

A message similar to the following message appears:

```
Mar 30 14:39:27 host inetd[733]: [ID 317013 daemon.notice] telnet[736]
from 192.11.11.13 32851
```

Task 2 – Using the logger Command to Demonstrate How Levels Operate

Complete the following steps:

1. Edit the /etc/syslog.conf file so that it includes the following line:

```
local0.notice <TAB> /var/log/local0.log
```

2. Create a file called /var/log/local0.log.

```
# touch /var/log/local0.log
```

3. Cause the syslogd daemon to reread the /etc/syslog.conf file by sending it a HUP signal.

4. In the window in which the tail command is running, stop the tail process. Restart the tail command so that it displays the end of the /var/log/local0.log file.

```
# tail /var/log/local0.log
```

5. In an available window, use the logger utility to send a message using the local0 *facility* and the notice *level*.

What, if any, new messages does the tail command display?

A message similar to the following appears:

```
Mar 30 15:21:49 host root: [ID 702911 local0.notice] "Notice-level
message"
```

6. In an available window, use the `logger` command to send a message by using the `local0` *`facility`* and the `crit` *`level`*.

```
# logger -p local0.crit Crit-level message
```

What, if any, new messages does the `tail` command display?

```
Mar 30 15:24:43 host1 root: [ID 702911 local0.crit] "Crit-level message"
```

A message similar to this displays because `crit` *is a higher* *`level`* *than* `notice`*, and the* `syslogd` *daemon is configured to recognize the* `notice` *`level`* *and higher for the* `local0` *`facility`*.

7. Run the `logger` command from Step 5 three times. Examine the output from the `tail` command in the other window. How many new messages appear in the `/var/log/local0.log` file?

One. The `syslogd` *daemon will not report multiple instances of the same message until a different message is logged, or the* `syslogd` *"*`mark`*" interval is reached.*

8. Run the `logger` command once.

Which new messages appear in the `/var/log/local0.log` file?

A message indicating that the previous message was repeated twice, and the new message; for example:

```
Mar 30 16:44:03 host last message repeated 2 times
Mar 30 16:44:38 host root: [ID 702911 local0.notice] "New notice-level
message"
```

9. Stop the `tail` command in the window where it is running.

Task 3 – Logging Messages to Another System

Complete the following steps:

Note – This step does not require you to change host names. In the following steps, substitute the appropriate host name for *`system1`* and *`system2`*.

1. On *`system1`*, edit the `/etc/syslog.conf` file, and change the line for `local0.notice` so that it reads as follows:

```
local0.notice<TAB>@system2
```

2. On *`system1`*, cause the `syslogd` daemon to reread the `/etc/syslog.conf` file using a `HUP` signal.

3. On *system2*, open a new terminal window, and use the `tail` command to view new entries as they arrive in the `/var/log/local0.log` file.

```
# tail /var/log/local0.log
```

4. On *system1*, use the `logger` command to generate a message by using the `local0.notice` *facility* and *level* pair.

```
# logger -p local0.notice Message from system1
```

5. On *system2*, which message is displayed in the window running the `tail` command?

   A message similar to the following:

```
Apr 1 13:07:49 system1 root: [ID 702911 local0.notice] Message from
system1
```

6. After verifying that *system1* has successfully passed messages to *system2*, stop the `tail` command on *system2*.

Task 4 – Logging Messages by Using the `loghost` Alias and `ifdef` Statements

Complete the following steps:

1. On both systems, edit the `/etc/syslog.conf` file, and uncomment the line that identifies `auth.notice` messages.

```
auth.noticeifdef('LOGHOST', /var/log/authlog, @loghost)
```

   Which two destinations are possible for these messages?

   `/var/log/authlog` – *This local host's log file*

   `@loghost` – *The* `syslog` *facility on the "loghost"*

2. On both systems, examine the `/etc/inet/hosts` file, and identify the name of the host associated with the `loghost` alias.

   In the default `/etc/inet/hosts` *file, the* `loghost` *alias is associated with the host name of the local system.*

3. On both systems, cause the `syslogd` daemon to reread the `/etc/syslog.conf` file by sending it a `HUP` signal.

4. On both systems, run the following `m4` commands, and record the line for `auth.notice` messages.

```
# /usr/ccs/bin/m4 -D LOGHOST /etc/syslog.conf
auth.notice /var/log/authlog
# /usr/ccs/bin/m4 /etc/syslog.conf
auth.notice @loghost
```

5. On both systems, open a terminal window, and use the `tail` command to view new entries as they arrive in the `/var/log/authlog` file.

```
# tail /var/log/authlog
```

6. On *system2,* use the `rlogin` command to log in to your own system, and then exit the connection.

```
# rlogin system2
Password: xxxxxx
...
# exit
```

On *system2,* which message is displayed in the window running the `tail` command?

A message similar to the following displays:

```
Mar 31 09:15:23 system2 login: [ID 254462 auth.notice] ROOT LOGIN
/dev/pts/7 FROM system2
```

On *system1,* does a new message display in the window running the `tail` command (yes or no)?

No.

7. On *system2,* change to the `/etc/inet` directory, and make a backup copy of the `/etc/inet/hosts` file. Edit the `/etc/inet/hosts` file to remove the `loghost` alias from the entry for *system2,* and add it to the entry for *system1*.

```
# cd /etc/inet
# cp hosts hosts.bak
# vi hosts
```

8. On *system2,* force the `syslogd` daemon to reread the `/etc/syslog.conf` file by sending it a `HUP` signal.

9. On *system2*, use the `rlogin` command to log in to your own system, and then exit the connection.

```
# rlogin system2
Password: xxxxxx
...
# exit
```

On *system2*, does a new message display in the window running the `tail` command (yes or no)?

No.

On *system1*, which message is displayed in the window running the `tail` command?

A message similar to the following displays:

```
Mar 31 09:34:46 system2 login: [ID 254462 auth.notice] ROOT LOGIN
/dev/pts/7 FROM system2
```

Task 5 – Completing the Exercise

Complete the following steps:

1. On both systems, stop the `tail` command in any window where it is running.
2. On *system2*, replace the `/etc/inet/hosts` file with the backup copy you made earlier.
3. On both systems, replace the `/etc/syslog.conf` file with the backup copy you made earlier.
4. On both systems, ensure that the `syslogd` daemon rereads the `/etc/syslog.conf` file by sending it a `HUP` signal.

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Using Name Services

Objectives

Name services centralize shared information on a network. There are several services that store and provide access to this information.

Upon completion of this module, you should be able to:

- Describe the name service concept
- Describe the name service switch file
- Configure the name service cache daemon (nscd)
- Get name service information

The following course map shows how this module fits into the current instructional goal.

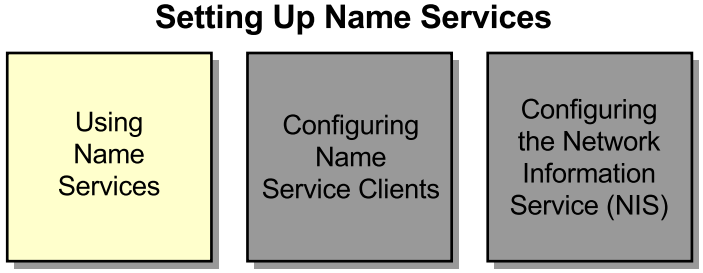


Figure 14-1 Course Map

Introducing the Name Service Concept

The original text-based UNIX® name service was developed for standalone UNIX systems and was then adapted for network use. While UNIX operating systems still support and use this text-based name service, it is not appropriate for large, complex networks. The name service concept uses domains, which are defined as a collection of network points or addresses.

The concept of a name service centralizes the shared information in a network. A single system, the name server, maintains the information previously maintained on each individual host. The name servers provide information such as host names, Internet Protocol (IP) addresses, user names, passwords, and automount maps.

Note – Local text files are never completely replaced by name services, for example, the `/etc/hosts` file.

Other hosts in the name service domain (called *clients*), request the information from the name server. This name server system responds to clients, and translates, or resolves their requests from its memory-based (cached) or disk-based databases.

Figure 14-2 shows one possible name service scenario. Later, this module describes alternatives to this scenario.

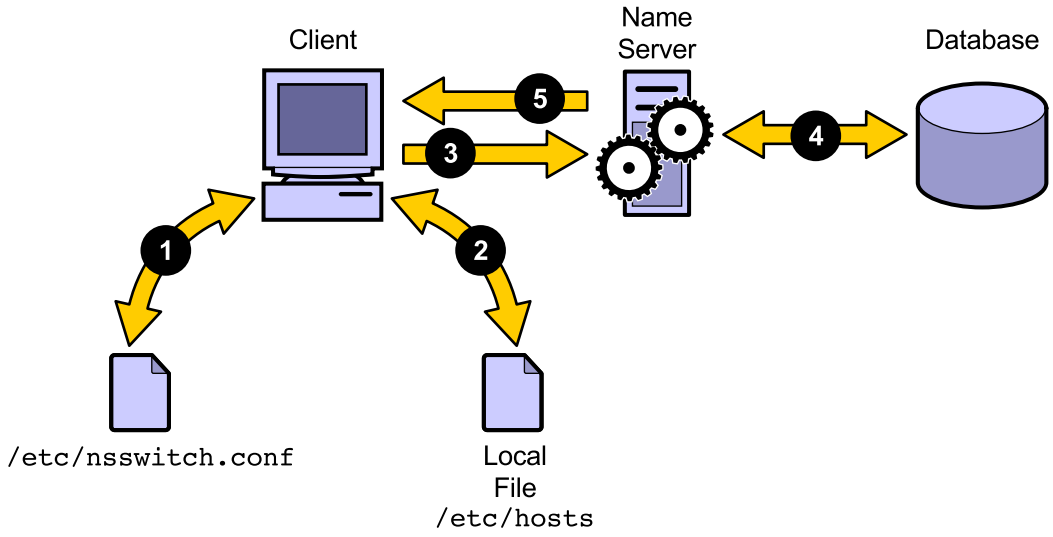


Figure 14-2 Name Service Scenario

The basic process is as follows:

1. The client requires administrative data to be accessed due to some process request. The client references its local name service switch file to determine the possible name service sources to search.
2. The name service switch file instructs the client to first search the local file for the information.
3. When the information is not located in the local files, the client's name service switch file redirects the search to a network name server.
4. The name server searches its database and locates the information.
5. The name server returns the information to its requesting client.

The name service concept provides the following benefits:

- A single point of administration for name service data
- Consistent name service information for systems within the domain
- All clients have access to changed data
- Assurance that clients do not miss updates

  In a file-based scheme, updates distributed by using File Transfer Protocol (FTP) could be missed if a host was down or off the network when the changes were propagated.
- Secondary servers prevent a single-point-of-failure

  While a single master server is all that is required, the name service scheme allows for the creation of secondary servers (sometimes referred to as *slaves* or *replicas*). These secondary servers maintain a copy of the master server's database, receive changes and updates to the database from the master, and participate in client query resolution. Therefore, they not only overcome a single point-of-failure, but they also play a role in improved name service performance by balancing the workload of answering client requests among multiple systems.

Domain Name System (DNS)

Domain Name System (DNS) is an Internet-wide naming system for resolving host names to IP addresses and IP addresses to host names. DNS supports name resolution for both local and remote hosts, and uses the concept of domains to allow hosts with the same name to coexist on the Internet.

The collection of networked systems that use DNS is referred to as the DNS *namespace*. The DNS namespace is divided into a hierarchy of domains. A DNS domain is a group of systems. Each domain is usually supported by two or more name servers, a master name server, and one or more slave name servers. Each server implements DNS by running the `in.named` daemon. On the client's side, DNS is implemented through the `kernel's` *resolver*. The resolver library resolves users' queries. The resolver queries a name server, which then returns either the requested information or a referral to another DNS server.

Figure 14-3 shows that the DNS namespace for the Internet begins with the root (`.`) domain and includes all subdomains, each of which is headed by a top-level domain.

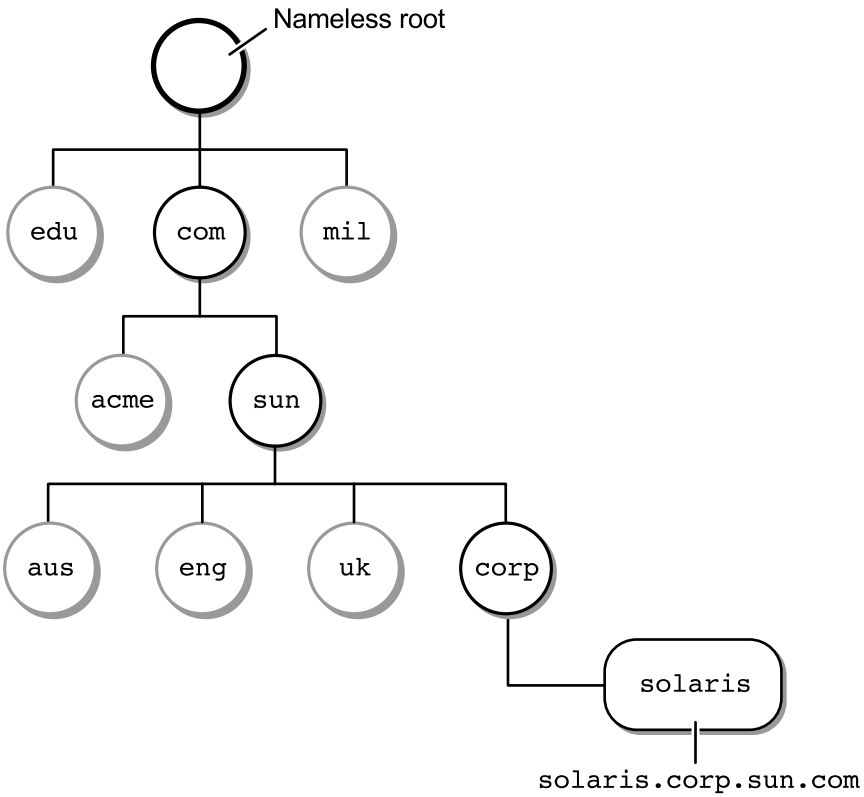


Figure 14-3 DNS Domain Structure

The top-level domains are administered by various organizations, all of which report to the governing authority called the Internet Corporation for Assigned Names and Numbers (ICANN). Administration of the lower-level domains is delegated to the various organizations that are registered domain name members within the top-level domain.

The top-level domain that you choose can depend on which one best suits the needs of your organization. Large organizations tend to use the organizational domains, while small organizations or individuals often choose to use a country code.

Everything below the connection to the domain falls into a zone of authority maintained by the connection to the domain. For example, everything below `sun.com` resides within the zone of authority for Sun Microsystems, Inc. and is, therefore, maintained by Sun Microsystems, Inc.

The DNS name servers store the host and IP address information in files called *zone files*. The `/etc/rc2.d/S72inetsvc` script starts the DNS process during the boot process if the configuration files are available.

Network Information Service (NIS)

Network Information Service (NIS) was developed independently of DNS and has a slightly different focus. DNS focuses on making communication easier by using host names instead of numerical IP addresses. NIS focuses on making network administration more manageable by providing centralized control over a variety of network information. NIS stores information about host names, addresses, users, groups, and network services. This collection of network information is referred to as the NIS namespace.

NIS namespace information is stored in files called NIS *maps*. NIS maps were designed to supplement many of the UNIX `/etc` files. These maps store much more than names and addresses. As a result, the NIS namespace has a large set of maps. NIS maps are database files created from source files in the `/etc` directory (or in a directory that you specify). By default, these maps are stored in the `/var/yp/`*`domainname`* directory on NIS servers. For example, the set of maps that contain hosts information include:

- `hosts.byaddr`
- `hosts.byname`

Note – You can obtain a list of the full set of maps from an NIS-configured system by running the `ypwhich -m` command.

NIS uses domains to define who can access the host names, user information, and other administrative data in its namespace. However, NIS does not use a domain hierarchy to store its data; therefore, the NIS namespace is flat.

You cannot directly connect an NIS domain to the Internet by using just NIS. However, organizations that want to use NIS and also want to be connected to the Internet can combine NIS with DNS. You can use NIS to manage all local information and use DNS for Internet host lookup. NIS provides a forwarding service that forwards host lookups to DNS if the information cannot be found in an NIS map. The Solaris OE also allows you to set up the `nsswitch.conf` file so that lookup requests for hosts:

- Go only to DNS
- Go to DNS and then to NIS, if the requests are not found by DNS
- Go to NIS and then to DNS, if the requests are not found by NIS

NIS uses a client-server arrangement similar to DNS. Replicated NIS servers provide services to NIS clients. The principal server is called a master server, and, for reliability, it has a backup, or a slave server. Both master and slave servers use the NIS information retrieval software and both store NIS maps.

Each server implements NIS by running the `ypserv` daemon. All NIS clients and servers must run the `ypbind` daemon to exchange NIS information. The `/etc/rc2.d/S71rpc` script starts the NIS processes during the boot process. NIS processes are only started if the appropriate configuration conditions are met.

Network Information Service Plus (NIS+)

Network Information Service Plus (NIS+) is similar to NIS but provides many more features. NIS+ is not an extension of NIS. NIS+ is a different software program.

You can configure the NIS+ name service to match the requirements of the organization using it. NIS+ enables you to store information about machine addresses, security information, mail information, Ethernet interfaces, and network services in central locations where all machines on a network can have access to the information. This configuration of network information is referred to as the NIS+ namespace.

The NIS+ namespace is hierarchical and is similar in structure to the UNIX directory tree. The hierarchical structure allows an NIS+ namespace to be configured to conform to the logical hierarchy of an organization. The namespace's layout of information is unrelated to its physical arrangement. Therefore, an NIS+ namespace can be divided into multiple domains that can be administered independently. Clients might have access to information in other domains in addition to their own if they have the appropriate permissions.

NIS+ uses a client-server model to store and gain access to the information contained in an NIS+ namespace. Each domain is supported by a set of servers. The principal server is called the root server, and the backup servers are called replica servers. The network information is stored in standard NIS+ tables in an internal NIS+ database. Both root and replica servers run NIS+ server software as well as maintain copies of NIS+ tables. Unlike NIS, the NIS+ namespace is dynamic because updates can occur and be put into effect at any time by any authorized user. Changes made to the NIS+ data on the root server are automatically and incrementally propagated to the replica servers.

NIS+ includes a sophisticated security system to protect the structure of the namespace and its information. NIS+ uses authentication and authorization to verify whether a client's request for information should be fulfilled. Authentication determines whether the information requester is a valid user on the network. Authorization determines whether a particular user is allowed to have or to modify the information requested.

Each server implements NIS+ by running the `rpc.nisd` daemon. NIS+ clients and servers run the `nis_cachemgr` daemon to enhance data access performance. The `/etc/rc2.d/S71rpc` script starts the NIS+ name service during the boot process. NIS+ processes are only started if the appropriate configuration conditions are met.

Lightweight Directory Access Protocol (LDAP)

The Solaris™ 9 Operating Environment (Solaris 9 OE) supports Lightweight Directory Access Protocol (LDAP) with the iPlanet™ Directory Server 5.1, as well as other LDAP directory servers. Services supported by LDAP include application servers, calendar servers, and messaging servers.

LDAP is the emerging industry standard protocol for accessing directory servers. LDAP is a lightweight protocol that uses a simplified set of system-independent encoding methods and runs directly on top of Transmission Control Protocol/Internet Protocol (TCP/IP).

LDAP directories provide a way to name, manage, and access collections of directory entries. A directory entry is composed of attributes that have a type and one or more values. The syntax for each attribute defines the allowed values, or the allowed data type of the attribute values, such as American Standard Code for Information Interchange (ASCII) characters or a numerical data. LDAP also defines how those values are interpreted during a directory operation; for example, determining if a search or compare is case sensitive.

Directory entries are organized into a tree structure, which can be based on boundaries defined by geography (country), organization (company), or domain components (dc).

Entries are named according to their position in this tree structure by a distinguished name (DN). Each component of the DN is called a relative distinguished name (RDN). An RDN is composed of one or more attributes from the entry.

The hierarchy of the directory tree structure is similar to that of the UNIX file system. An RDN is similar to the relative path name of a file, and the DN is similar to the absolute path name. As in the UNIX file system, sibling directory entries must have unique RDNs. However, in the directory tree, each entry can contain content or attributes.

Like the DNS namespace, LDAP names start with the least significant component and proceed to the most significant; in other words, those just below root. The DN is constructed by concatenating the sequence of RDNs up to the root of the tree.

Figure 14-4 shows an example of a Solaris LDAP Directory Information Tree.

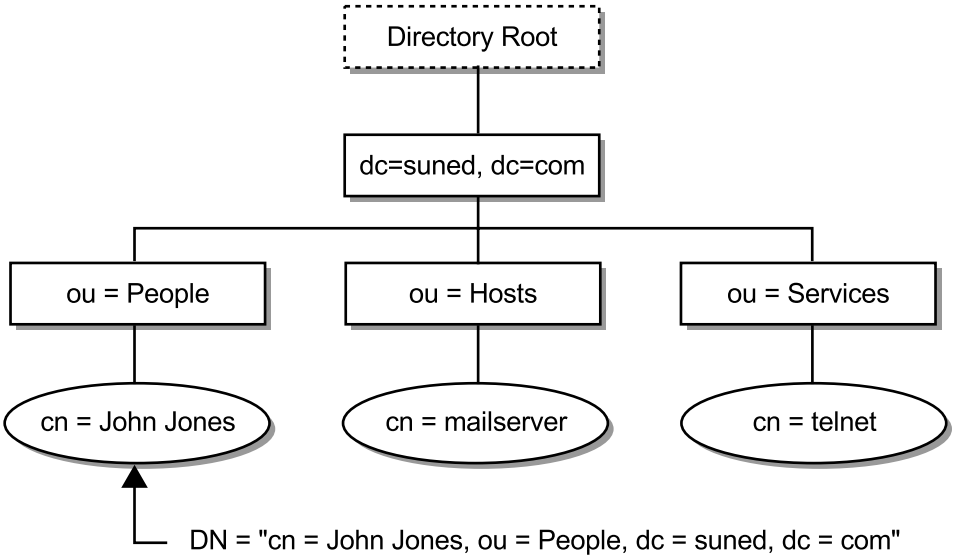


Figure 14-4 Solaris LDAP Directory Information Tree

The iPlanet Directory Server 5.1 must be set up and then configured to support Solaris LDAP clients. On a configured LDAP server, the `/etc/rc2.d/S72directory` script starts the iPlanet Directory Server during the boot process.

Name Service Features Summary

Table 14-1 lists and compares the name services available in the Solaris OE.

Table 14-1 Name Service Features

Feature	DNS	NIS	NIS+	LDAP
Namespace	Hierarchical	Flat	Hierarchical	Hierarchical
Data storage	Files/resource records	Two column maps	Multicolumn tables	Directories (varied)
Server types	Master/ slave/ caching only/ forwarding	Master/ slave	Root master/ non-root master/ replica	Master/ consumer
Transport	TCP/IP	TCP/IP	TCP/IP	TCP/IP
Scale	Wide area network (WAN)	Local area network (LAN)	LAN	WAN

Introducing the Name Service Switch File

The name service switch file determines which name services a system uses to search for information, and in which order the name services are searched. All Solaris OE systems use the `/etc/nsswitch.conf` file as the name service switch file. The `nsswitch.conf` file is loaded with the contents of a template file during the installation of the Solaris OE, depending on the name service that is selected, as shown in Table 14-2.

Table 14-2 Name Service Template Files

Name Service	Name Service Template
Local files	`/etc/nsswitch.files`
DNS	`/etc/nsswitch.dns`
NIS	`/etc/nsswitch.nis`
NIS+	`/etc/nsswitch.nisplus`
LDAP	`/etc/nsswitch.ldap`

Note – If you select the default name service during installation of the Solaris 9 OE, the `/etc/nsswitch.nisplus` template configures the name service for NIS+.

The following example is the `/etc/nsswitch.conf` file configured to support the NIS name service using the `/etc/nsswitch.nis` template.

```
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

# the following two lines obviate the "+" entry in /etc/passwd and
/etc/group.
passwd:     files nis
group:      files nis

# consult /etc "files" only if nis is down.
```

```
hosts:      nis [NOTFOUND=return] files
ipnodes:    files
# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
#ipnodes:    nis [NOTFOUND=return] files

networks:   nis [NOTFOUND=return] files
protocols:  nis [NOTFOUND=return] files
rpc:        nis [NOTFOUND=return] files
ethers:     nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files
bootparams: nis [NOTFOUND=return] files
publickey:  nis [NOTFOUND=return] files

netgroup:   nis

automount:  files nis
aliases:    files nis

# for efficient getservbyname() avoid nis
services:   files nis
sendmailvars:   files
printers:       user files nis

auth_attr:  files nis
prof_attr:  files nis
project:    files nis
```

The `/etc/nsswitch.conf` file includes a list of databases that are sources of information about IP addresses, users, and groups. Data for these can come from a variety of sources. For example, host names and host addresses, are located in the `/etc/hosts` file, NIS, NIS+, LDAP, or DNS. Each database has zero or more sources; the sources and their lookup order are specified in the `/etc/nsswitch.conf` file.

Database Sources

There is an entry in the `/etc/nsswitch.conf` file for each database. Some typical examples of these entries are:

- `ipnodes: files`
- `passwd: files nis`
- `hosts: nis [NOTFOUND=return] files`

The information sources are listed in the order that they are searched, and these sources are defined in Table 14-3.

Table 14-3 Information Sources

Information Sources	**Description**
`files`	Specifies that entries be obtained from a file stored in the client's `/etc` directory. For example, `/etc/hosts`.
`nisplus`	Specifies that entries be obtained from an NIS+ table. For example, the `hosts` table.
`nis`	Specifies that entries be obtained from an NIS map. For example, the `hosts` map.
`dns`	Specifies that host information be obtained from DNS.
`ldap`	Specifies that entries be obtained from the LDAP directory.
`user`	Specifies that printer information be obtained from the `${HOME}/.printers` file

There might be a single information source listed, in which case the search terminates if the information is not found. If two or more sources are listed, the first listed source is searched before moving on to the next listed source. The relationships between these name service keywords, when found in the `nsswitch.conf` file, is further explained in Table 14-4 on page 14-14 and Table 14-5 on page 14-15.

Status Codes

When multiple information sources are specified, it is sometimes necessary to define precisely the circumstances under which each source is searched. When a name service is referenced, the attempt to search this source can return one of the following status codes, as shown in Table 14-4.

Table 14-4 Status Message Codes

Status Message	Meaning of Message
SUCCESS	The requested entry was found in the specified source.
UNAVAIL	The source is not configured on this system and cannot be used. In other words, the NIS or NIS+ processes could not be found or contacted.
NOTFOUND	The source responded with `No such entry`. In other words, the table, map, or file was accessed, but it did not contain the needed information.
TRYAGAIN	The source is busy. It might respond if tried again. In other words, the name service is running and was contacted but could not service the request at that moment.

Actions

For each status code, two actions are possible, as shown in Table 14-5.

Table 14-5 Status Code Actions

Action	Meaning of Action
`return`	Stop looking for the information.
`continue`	Try the next source, if there is one.

When the action is not explicitly specified, the default action is to continue the search using the next specified information source, as follows:

- `SUCCESS = return`
- `UNAVAIL = continue`
- `NOTFOUND = continue`
- `TRYAGAIN = continue`

For example:

```
ipnodes:    files
```

In this example, the `/etc/inet/ipnodes` file is searched for the first entry that matches the requested host name. If no matches are found, an appropriate error is returned, and no further information sources are searched.

Another example:

```
passwd: files nis
```

In this example, the appropriate files in the `/etc` directory are searched for the corresponding password entry. If the entry is not found, the NIS maps are searched for the entry. If no entry is found in the NIS maps, an appropriate error is returned, and no further information sources are searched.

Another example:

```
hosts: nis [NOTFOUND=return] files
```

In this example, the NIS maps are searched for the entry. If the source (NIS) is not running, the system returns the status `UNAVAIL`, and continues to search the `/etc/inet/hosts` file. If the entry returns the status `NOTFOUND`, an appropriate error is returned, and the search is terminated without searching the `/etc/inet/hosts` file.

Configuring the Name Service Cache Daemon (nscd)

To properly use the name service cache daemon (nscd), you must be able to perform the following:

- Describe the purpose of the name service cache daemon
- Configure the name service cache daemon
- Stop and start the name service cache daemon

The nscd Daemon

The nscd daemon is a process that provides a cache for the most common name service requests. The nscd daemon starts during multiuser boot. The /etc/nscd.conf configuration file controls the behavior of the nscd daemon. The nscd daemon provides caching for the passwd, group, hosts, ipnodes, exec_attr, prof_attr, and user_attr databases. Solaris OE system calls automatically reference the nscd cache if the nscd cache holds the type of data needed. Standardized calls retrieve the cached data. The calls take the form of getXbyY, such as gethostbyname, gethostbyaddr, and so on.

The data in each cache has a separately defined, time-to-live. Modifying the local database (/etc/hosts, for example) causes the corresponding cache to become invalidated upon the next call to the nscd daemon.

Configuring the nscd Daemon

The `/etc/nscd.conf` file contains the configuration information for the `nscd` daemon. Each line specifies either an *attribute* and a *value*, or an *attribute*, a *cache name*, and a *value*. An example of an attribute and a value is:

```
logfile                 /var/adm/nscd.log
```

An example of an attribute, a cache name, and a value is:

```
enable-cache            hosts           no
# cat /etc/nscd.conf
#
# Copyright (c) 1994-2001 by Sun Microsystems, Inc.
# All rights reserved.
#
#ident  "@(#)nscd.conf  1.6     01/01/26 SMI"
#

#
#       Currently supported cache names: passwd, group, hosts, ipnodes
#           exec_attr, prof_attr, user_attr
#

#       logfile                 /var/adm/nscd.log
#       enable-cache            hosts           no

        debug-level             0

        positive-time-to-live   passwd          600
        negative-time-to-live   passwd          5
        suggested-size          passwd          211
        keep-hot-count          passwd          20
        old-data-ok             passwd          no
        check-files             passwd          yes

        positive-time-to-live   group           3600
        negative-time-to-live   group           5
        suggested-size          group           211
        keep-hot-count          group           20
        old-data-ok             group           no
        check-files             group           yes

        positive-time-to-live   hosts           600
        negative-time-to-live   hosts           5
```

```
suggested-size          hosts           211
keep-hot-count          hosts           20
old-data-ok             hosts           no
check-files             hosts           yes

positive-time-to-live   ipnodes         3600
negative-time-to-live   ipnodes         5
suggested-size          ipnodes         211
keep-hot-count          ipnodes         20
old-data-ok             ipnodes         no
check-files             ipnodes         yes

positive-time-to-live   exec_attr       3600
negative-time-to-live   exec_attr       300
suggested-size          exec_attr       211
keep-hot-count          exec_attr       20
old-data-ok             exec_attr       no
check-files             exec_attr       yes

positive-time-to-live   prof_attr       3600
negative-time-to-live   prof_attr       5
suggested-size          prof_attr       211
keep-hot-count          prof_attr       20
old-data-ok             prof_attr       no
check-files             prof_attr       yes

positive-time-to-live   user_attr       3600
negative-time-to-live   user_attr       5
suggested-size          user_attr       211
keep-hot-count          user_attr       20
old-data-ok             user_attr       no
check-files             user_attr       yes
```

Stopping and Starting the nscd Daemon

Proper updates to the name service databases notify the nscd daemon to update its cache, as needed. However, the nscd daemon's cache might become out of date due to various abnormal circumstances or due to hand-editing files. A common way to force the nscd daemon to update its cache is to stop and start the daemon.

The preferred method for stopping and starting the nscd daemon is by using the /etc/init.d/nscd script.

Stopping the nscd Daemon

The nscd daemon stops automatically when the system changes to:

- Run level 1 using the /etc/rc1.d/K40nscd script
- Run level S using the /etc/rcS.d/K40nscd script
- Run level 0 using the /etc/rc0.d/K40nscd script

You can also manually stop the nscd daemon as follows:

```
# /etc/init.d/nscd stop
```

Starting the nscd Daemon

The nscd daemon starts automatically when the system changes to run level 2 using the /etc/rc2.d/S76nscd script. You can also manually start the nscd daemon as follows:

```
# /etc/init.d/nscd start
```

Retrieving Name Service Information

There are many tools available for acquiring information stored within the various name service information sources. Selecting the correct tool can reduce troubleshooting time when isolating name service malfunctions. The `getent` command provides a generic retrieval interface to search many name service databases.

The `getent` Command

As a system administrator, you can query name service information sources with tools, such as the `ypcat`, `nslookup`, `niscat`, and `ldaplist` commands.

You can use the `ypcat` command to query the NIS namespace. You can use the `nslookup` command to query the DNS namespace. However, when trying to isolate a problem, using one of these tools can return different results than standard system search operations, because the `nsswitch.conf` file is not referenced by these commands.

The `getent` command has these advantages:

- The primary advantage is that the command searches the information sources in the order in which they are configured in the name service switch file.
- A secondary advantage is that by using the name service switch file, the defined status message codes and actions are tested as they are currently configured. Therefore, if a `return` action is improperly placed in the name service switch file, the `getent` command will find the problem, whereas the specific commands used to test the name service information sources (such as `ypcat` or `nslookup`) will not find the problem because they directly use the name service database without referencing the `nsswitch.conf` file.

Using the getent Command

The `getent` command retrieves a list of entries from the administrative database specified by *database*. The sources for the database are specified in the `/etc/nsswitch.conf` file. The syntax is:

```
getent database [key]...
```

where:

database	The name of the database to be examined. This name can be `passwd`, `group`, `hosts`, `ipnodes`, `services`, `protocols`, `ethers`, `networks`, or `netmasks`.
key	A value that corresponds to an entry in a database. The *key* must be in a format appropriate for searching on the respective database. For example, it can be a username or numeric user ID (UID) for `passwd`, or a host name or IP address for `hosts`.

For the following examples, the `/etc/nsswitch.conf` file is configured to search files and then to search NIS.

```
# getent passwd lp
lp:x:71:8:LinePrinter Admin:/usr/spool/lp:

# getent group 10
staff::10:

# getent hosts sys44
192.168.30.44 sys44 loghost
```

The previous example assumes that the `/etc/nsswitch.conf` file is configured to search files and then to search NIS. If the `/etc/nsswitch.conf` file is configured to search NIS and then to search files, the output of the final search would be:

```
# getent hosts sys44
192.168.30.44 sys44
```

Notice the absence of `loghost` in this output. The `loghost` alias is a feature of the `sys44` entry in the `/etc/inet/hosts` file but not the NIS map. Therefore, when the `/etc/nsswitch.conf` file search order is altered, the `getent` command looks up the entry in the NIS map before consulting the `/etc/inet/hosts` file.

Exercise: Reviewing Name Services

In this lab, you evaluate your understanding of the name services concepts presented in this module.

Preparation

If necessary, refer to your lecture notes to answer these exercise questions.

Tasks

Answer the following questions:

1. List the name services that can be configured in the `/etc/nsswitch.conf` file.

   _____

   _____

2. Which name service is selected by default during the installation of the Solaris 9 OE?

   _____

3. What are the two main services provided by DNS?

   _____

   _____

4. What types of information are stored within the NIS+ namespace?

   _____

   _____

5. Which file is referred to as the name service switch file, and why?

   _____

   _____

6. If you decide to use the LDAP for name service resolution, which template file would you use to create the name service switch file?

   _____

   _____

7. How is the following entry in the name service switch file interpreted?

   ```
   hosts: nis [NOTFOUND=return] files
   ```

   _____

   _____

8. Is the following an appropriate entry to the `/etc/nsswitch.conf` file? Why or why not?

   ```
   groups: dns files nis
   ```

   _____

   _____

Task Solutions

1. List the name services that can be configured in the `/etc/nsswitch.conf` file.

   Local files, DNS, NIS, NIS+, and LDAP.

2. Which name service is the default selection during the installation of the Solaris 9 OE?

   NIS+ is selected by default during a Solaris 9 OE installation.

3. What are the two main services provided by DNS?

   DNS provides host name-to-IP address translation and IP address-to-host name translation.

4. What types of information are stored within the NIS+ namespace?

   The NIS+ namespace stores information about workstation addresses, security information, mail information, Ethernet interfaces, printers, and network services.

5. Which file is referred to as the name service switch file, and why?

   The `/etc/nsswitch.conf` *file is referred to as the name service switch file because the operating system uses it to determine where to go for any information lookups. This file indicates whether DNS, NIS, NIS+, LDAP, or local files are to be used for name service resolution. If more than one name service is to be used, this file indicates the order in which these services should be accessed.*

6. If you decide to use the LDAP for name service resolution, which template file would you use to create the name service switch file?

`/etc/nsswitch.ldap`

7. How is the following entry in the name service switch file interpreted?

```
hosts: nis [NOTFOUND=return] files
```

   Assuming that the NIS name service is running and available, the syntax for this entry means that the NIS hosts table is searched. If an NIS server is busy or unavailable, the local files are searched. If an NIS server has no map entry for a host lookup, the system would not reference the local files.

8. Is the following an appropriate entry to the `/etc/nsswitch.conf` file? Why or why not?

```
groups: dns files nis
```

   This is not an appropriate entry in the `/etc/nsswitch.conf` *file, because* `dns` *only applies to the* `hosts` *entry in the name service switch file.*

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 15

Configuring Name Service Clients

Objectives

This module explains how to configure a client to use DNS or LDAP as the name service. Setting up the DNS server is described in the SA-399: *Network Administration for the Solaris™ 9 Operating Environment* course. Setting up the LDAP server is described in the IN-350: *LDAP Design and Deployment* course.

Upon completion of this module, you should be able to:

- Configure a DNS client
- Set up an LDAP client

The following course map shows how this module fits into the current instructional goal.

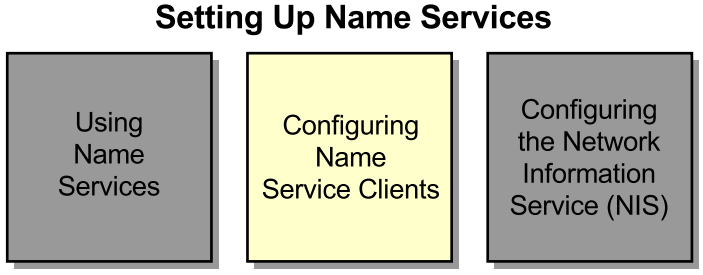


Figure 15-1 Course Map

Configuring a DNS Client

Name resolution using the Internet domain name system begins with the client-side resolver. The resolver is a set of routines that are built into the operating system. The client resolver code is controlled by the following files:

`/etc/resolv.conf`	Contains directives to specify the scope of a query
`/etc/nsswitch.conf`	Contains the reference to DNS for the hosts entry

Configuring the DNS Client During Installation

During the system identification phase of a Solaris 9 OE installation, you you will use several windows to configure the name service.

To configure the system to use DNS, complete the following steps:

1. In the Name Service window, select DNS as the name service, as shown in Figure 15-2. Press F2 to continue.

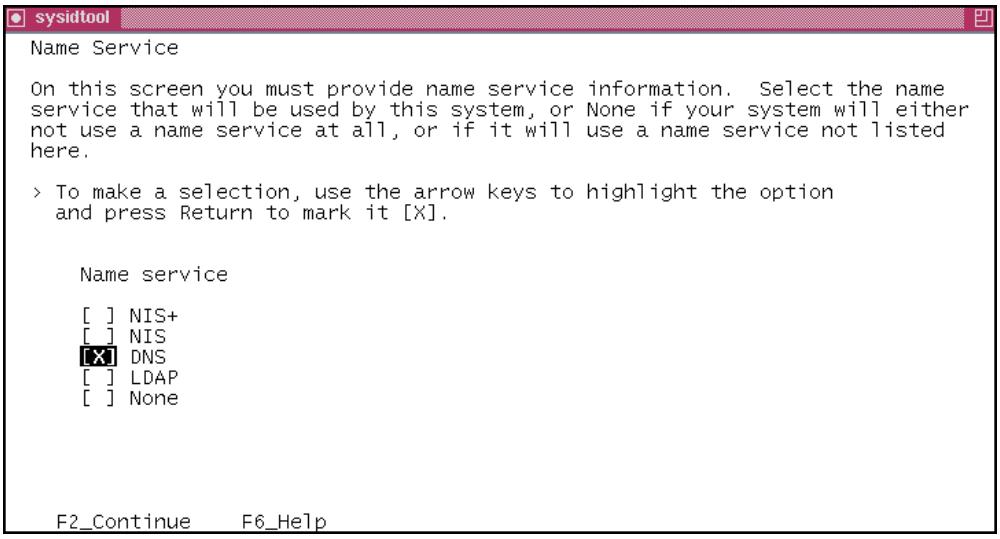


Figure 15-2 Name Service Window

2. In the Domain Name window, enter the DNS domain name to which the client will belong, as shown in Figure 15-3, and press F2 to continue.

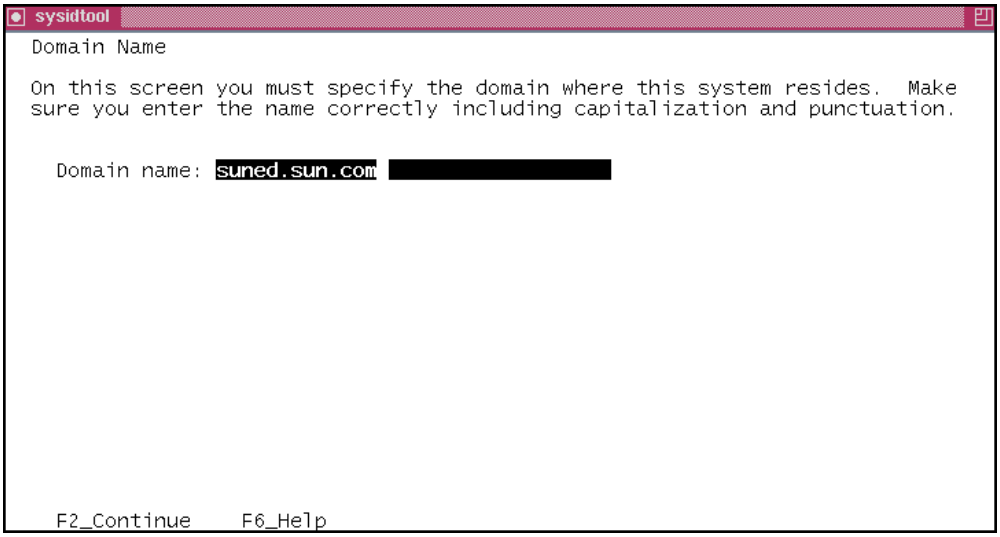


Figure 15-3 Domain Name Window

3. In the DNS Server Address window, enter the IP addresses of up to three DNS servers that the client will use for lookups, as shown in Figure 15-4. Press F2 to continue.

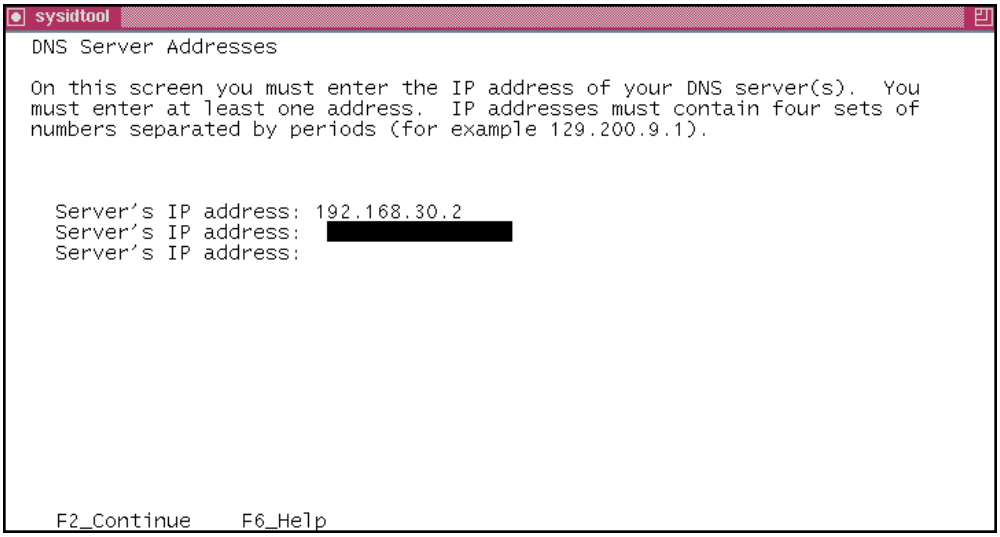


Figure 15-4 DNS Server Address Window

4. In the DNS Search List window, enter search suffixes that will supplement searches for names that are not fully qualified (names that do not include a complete domain name), as shown in Figure 15-5. Press F2 to continue.

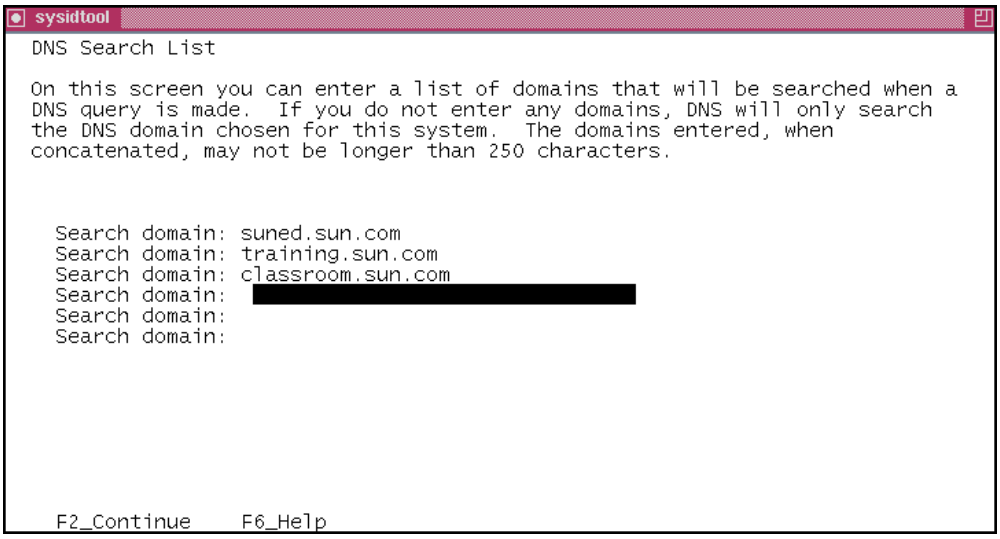


Figure 15-5 DNS Search List Window

5. In the Confirm Information window, verify that you have provided accurate information, as shown in Figure 15-6. Press F2 to continue.

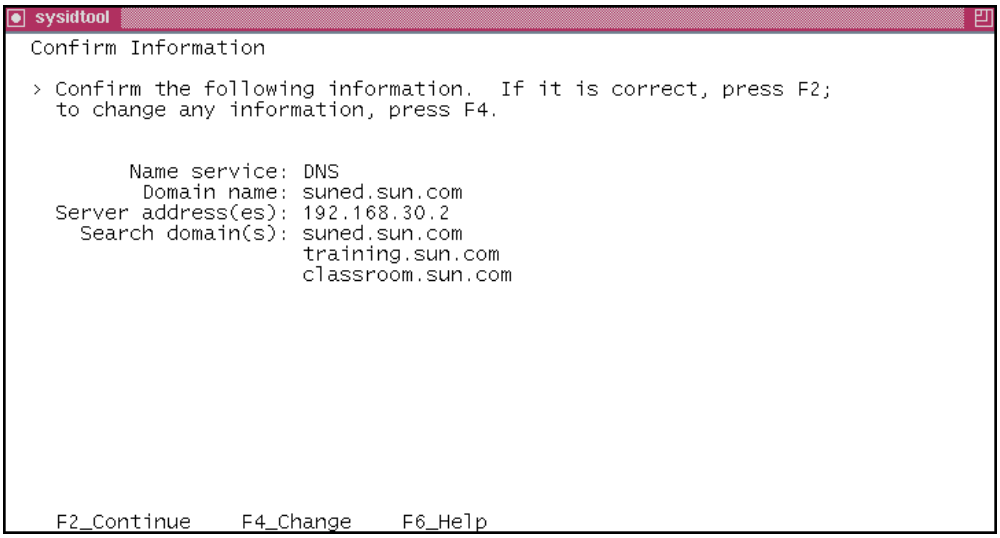


Figure 15-6 Confirm Information Window

Editing DNS Client Configuration Files

The installation window only allows the selection of DNS with the default of local files for the name service. Therefore, to use DNS with another name service, such as NIS or LDAP, you must manually modify the configuration files after the system is configured.

Editing the /etc/resolv.conf File

The /etc/resolv.conf file contains configuration directives for the DNS resolver. The directives include:

nameserver	Specifies the IP address of a name server for the DNS domain in which the host is located. You can list up to three name servers, one on each line.
domain	Specifies the local domain name. Specifying the local domain name allows queries using just the host name.
search	Provides a list of domain names, separated by spaces or tabs, that is appended to unqualified name queries until a match is found. When used without the presence of the domain directive, the first domain listed in the search list is the local domain.

The following resolv.conf example shows two name servers for the suned.sun.com domain with a search that includes the training.sun.com domain, as well as the sun.com domain.

```
# cat /etc/resolv.conf
nameserver 192.168.10.11
nameserver 192.168.20.88
search suned.sun.com training.sun.com sun.com
```

Note – Using the domain directive is a legacy method of listing the local domain. Although the domain directive is still a supported directive, beginning with the Solaris 9 OE release, training examples list the local domain as the first argument to the search directive.

Copying the /etc/nsswitch.dns File to the /etc/nsswitch.conf File

To configure a client to use DNS in combination with the system's local files, copy the /etc/nsswitch.dns file to the /etc/nsswitch.conf file. This action only changes the hosts entry as follows:

```
# cat /etc/nsswitch.conf
...
hosts: files dns
...
```

If you want to add DNS name resolution to a system currently running a name service, such as NIS or NIS+, you cannot copy a nsswitch template into the nsswitch.conf file. You must manually edit the current nsswitch file, and place the dns keyword on the hosts line in the specific location, along with other keywords. The following example shows that DNS is queried after NIS and the /etc/hosts file.

```
# cat /etc/nsswitch.conf
...
hosts: nis files dns
...
```

Setting Up an LDAP Client

Native LDAP is the client implementation of the LDAP name service. An LDAP server, such as the iPlanet Directory Server 5.*x* that is bundled with the Solaris 9 OE, must exist on the network.

Note – The LDAP server cannot be a client of itself. Getting this configuration to work properly requires changes to the LDAP server and the LDAP client.

Client Authentication

An LDAP client must establish a session with an LDAP server. This authentication process is known as binding. After a client is authenticated, it can then perform operations, such as “search and modify,” on the data. Authorization is the granting of access to controlled system resources. Solaris OE LDAP clients have read-only access to name service data, such as host names, email aliases, and net groups. Users have read-write access to certain data, such as their own passwords. Privileged administrator accounts have read-write access to other data. When finished, the client unbinds, or closes, the session.

Details on how the client is authenticated and what data the client is authorized to access is maintained on the LDAP server. To simplify Solaris OE client setup and to avoid having to reenter the same information for each and every client, a single client profile is created on the directory server.

Client Profile and Proxy Account

A single client profile defines the configuration parameters for a group of Solaris OE clients allowed to access the LDAP database.

A client profile:

- Contains the client's credential information
- Describes how authentication is to take place
- Provides the client with various configuration parameters

A proxy account is created to allow multiple clients to bind to the server with the same access privileges. Only one name and password is needed for all the clients in a group to bind to the LDAP server, rather than configuring each client with its own account name and password.

Client Initialization

The client profile and proxy account are created as part of the iPlanet Directory Server 5.*x* setup procedures on the Solaris 9 OE. By default, the client profile named `default` and the proxy account `proxyagent` are created under a special profile directory entry.

When the Solaris LDAP client is initialized, a copy of the client profile is retrieved from the server and stored on disk. On the LDAP client, the `ldap_cachemgr` daemon is responsible for maintaining and updating the changes to the client profile information. The `ldap_cachemgr` daemon keeps a copy of the profile in memory and uses it when binding to the server.

Configuring the LDAP Client During Installation

To configure the LDAP client, perform the following steps:

1. In the Name Service window, select LDAP as the name service, as shown in Figure 15-7, and press F2 to continue.

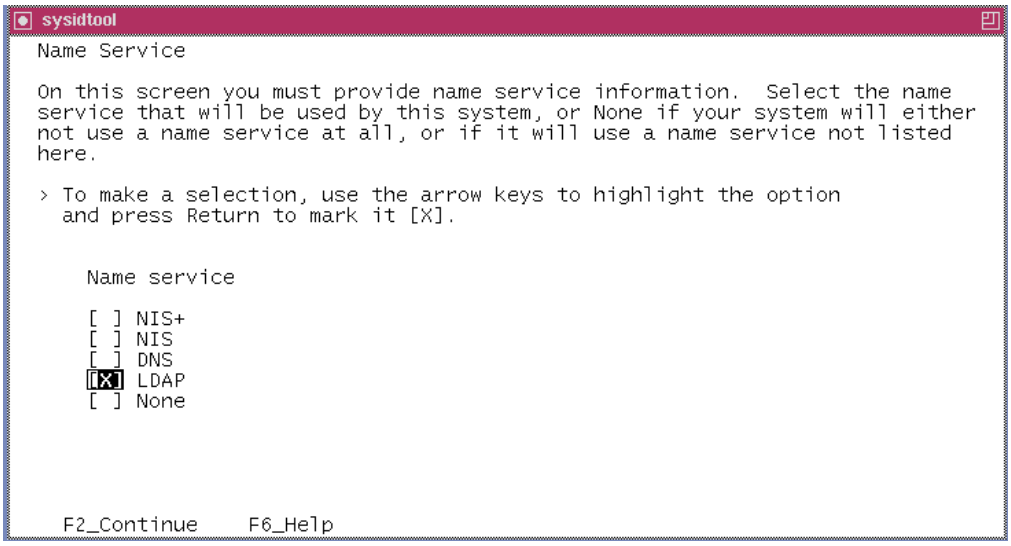


Figure 15-7 Name Service Window

Note – When you specify LDAP as the name service, the client host name must exist in the `ou=hosts` container on the LDAP server.

2. In the Domain Name window, enter the domain name where the system is located, as shown in Figure 15-8, and press F2 to continue.

Figure 15-8 Domain Name Window

3. In the LDAP Profile window, enter the profile name and server IP address, as shown in Figure 15-9, and press F2 to continue.

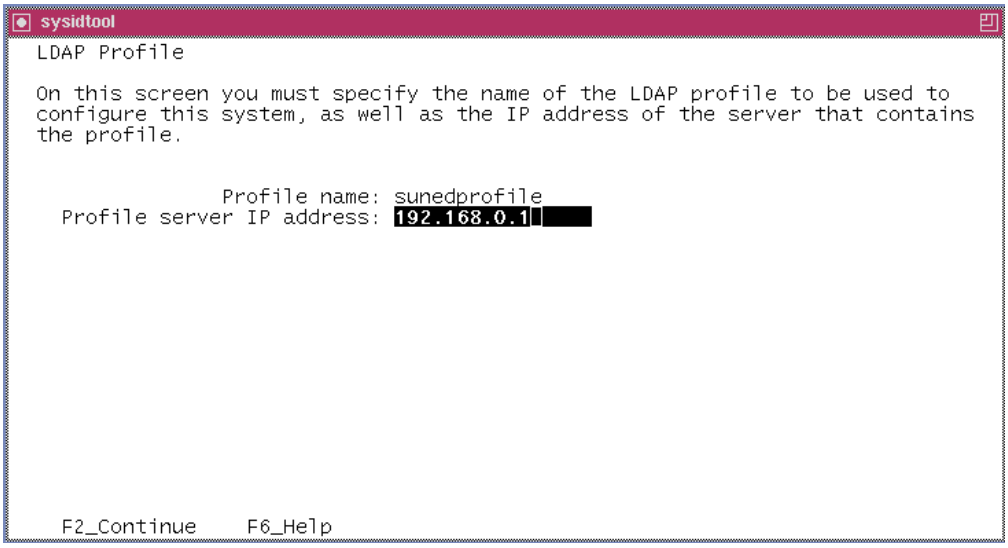


Figure 15-9 LDAP Profile Window

4. In the Confirm Information window, verify that you have provided accurate information, as shown in Figure 15-10, and press F2 to continue.

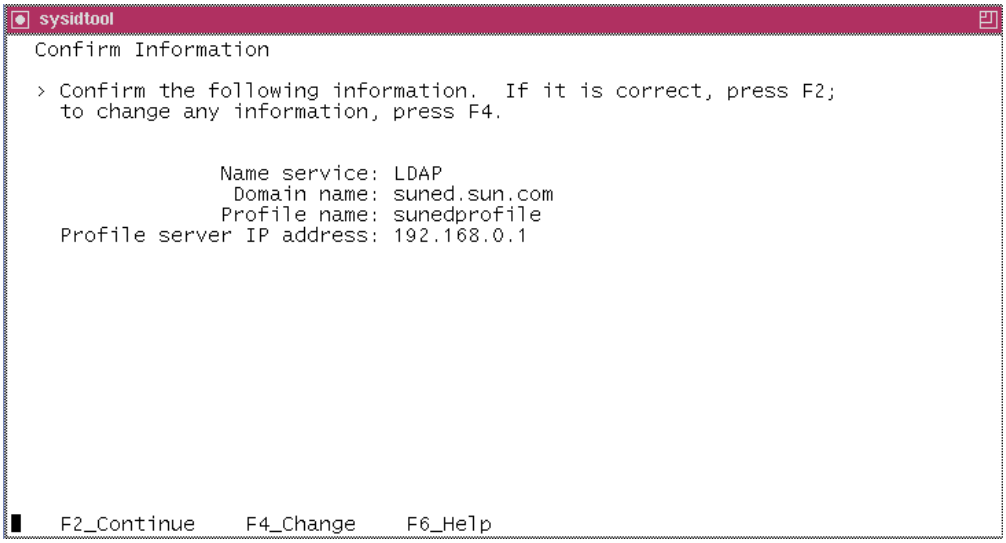


Figure 15-10 Confirm Information Window

Note – The information that must be supplied during the installation is some of the same information that you would enter using the `ldapclient` command.

Initializing the Native LDAP Client

You execute the `ldapclient` command on the client system once to initiate the client as a native LDAP client. The required command-line arguments include the proxy name, password, and the LDAP server's IP address.

The following example describes a typical client initialization:

```
# ldapclient init -a proxyPassword=proxy \
-a proxyDN=cn=proxyagent,ou=profile,dc=suned,dc=sun,dc=com\
-a domainname=suned.sun.com 192.168.0.100
System successfully configured
```

where:

`init`	Initializes the host as an LDAP client
`proxyPassword`	The password for the `proxyagent`
`proxyDN`	The DN for the `proxyagent`
`domainname`	The domain for which the server is configured
`192.168.0.100`	LDAP server IP address

The `ldapclient` command creates two files in the `/var/ldap` directory on the LDAP client. These files contain the information that the LDAP clients use when binding to and accessing the LDAP database.

Note – The two files in the `/var/ldap` directory are currently ASCII files, but might not be in the future. The `ldapclient list` command is the best way to see this information.

The `ldap_client_cred` file contains the proxy agent information that the client uses for LDAP authentication; for example:

```
# cat ldap_client_cred
#
# Do not edit this file manually; your changes will be lost.Please use
ldapclient (1M) instead.
#
NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=suned,dc=sun,dc=com
NS_LDAP_BINDPASSWD= {NS1}ecc423aad0
```

The `ldap_client_file` file contains the configuration information from the client profile in the LDAP server database; for example:

```
# cat ldap_client_file
#
# Do not edit this file manually; your changes will be lost.Please use
ldapclient (1M) instead.
#
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_SERVERS= 192.168.0.100
NS_LDAP_SEARCH_BASEDN= dc=suned,dc=sun,dc=com
NS_LDAP_AUTH= simple
NS_LDAP_SEARCH_REF= FALSE
NS_LDAP_SEARCH_SCOPE= one
NS_LDAP_SEARCH_TIME= 30
NS_LDAP_CACHETTL= 43200
NS_LDAP_PROFILE= default
NS_LDAP_CREDENTIAL_LEVEL= proxy
NS_LDAP_BIND_TIME= 10
```

Note – Do not modify the `/var/ldap/ldap_client_file` file directly.

You can also use the `ldapclient` command to view the current client's local configuration. Refer to the `ldapclient` man page for a description of these attributes.

```
# ldapclient list
NS_LDAP_FILE_VERSION= 2.0
NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=suned,dc=sun,dc=com
NS_LDAP_BINDPASSWD= {NS1}ecc423aad0
NS_LDAP_SERVERS= 192.168.0.100
NS_LDAP_SEARCH_BASEDN= dc=suned,dc=sun,dc=com
NS_LDAP_AUTH= simple
NS_LDAP_SEARCH_REF= FALSE
NS_LDAP_SEARCH_SCOPE= one
NS_LDAP_SEARCH_TIME= 30
NS_LDAP_PROFILE= default
NS_LDAP_CREDENTIAL_LEVEL= proxy
NS_LDAP_BIND_TIME= 10
```

Copying the /etc/nsswitch.ldap File to the /etc/nsswitch.conf File

During LDAP client initialization, the /etc/nsswitch.ldap file is copied over the /etc/nsswitch.conf file.

The default nsswitch.conf file for an LDAP client follows.

```
# more nsswitch.conf
#
# /etc/nsswitch.ldap:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses LDAP in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

# the following two lines obviate the "+" entry in /etc/passwd and
/etc/group.
passwd:     files ldap
group:      files ldap
# consult /etc "files" only if ldap is down.
hosts:      ldap [NOTFOUND=return] files
ipnodes:    files
# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
#ipnodes:    ldap [NOTFOUND=return] files

networks:   ldap [NOTFOUND=return] files
protocols:  ldap [NOTFOUND=return] files
rpc:        ldap [NOTFOUND=return] files
ethers:     ldap [NOTFOUND=return] files
netmasks:   ldap [NOTFOUND=return] files
bootparams: ldap [NOTFOUND=return] files
publickey:  ldap [NOTFOUND=return] files

netgroup:   ldap

automount:  files ldap
aliases:    files ldap
# for efficient getservbyname() avoid ldap
```

```
services:   files ldap
sendmailvars:   files

# role-based access control
auth_attr: files ldap
exec_attr: files ldap
prof_attr: files ldap
user_attr: files ldap

# audit
audit_user: files ldap
project:    files ldap
```

Listing LDAP Entries

You use the `ldaplist` command to list the naming information from the LDAP servers. This command uses the application programming interface (API) to access the information. Refer to the `ldaplist` man page for additional information.

Without any arguments, the `ldaplist` command returns all of the containers in the current search `baseDN`. For example:

```
# ldaplist
dn: ou=Hosts,dc=suned,dc=sun,dc=com

dn: ou=Group,dc=suned,dc=sun,dc=com

dn: ou=rpc,dc=suned,dc=sun,dc=com

dn: ou=protocols,dc=suned,dc=sun,dc=com

dn: ou=networks,dc=suned,dc=sun,dc=com

dn: ou=netgroup,dc=suned,dc=sun,dc=com

dn: ou=aliases,dc=suned,dc=sun,dc=com

dn: ou=people,dc=suned,dc=sun,dc=com

dn: ou=services,dc=suned,dc=sun,dc=com

dn: ou=Ethers,dc=suned,dc=sun,dc=com

dn: ou=profile,dc=suned,dc=sun,dc=com

dn: nismapname=auto_home,dc=suned,dc=sun,dc=com

dn: nismapname=auto_direct,dc=suned,dc=sun,dc=com

dn: nismapname=auto_master,dc=suned,dc=sun,dc=com
```

Unconfiguring an LDAP Client

To unconfigure an LDAP client, use the `ldapclient` command with the `uninit` option. This command removes the client files from the `/var/ldap` directory and restores the previous `/etc/nsswitch.conf` file. The `ldap_cachemgr` process is also stopped. The changes to the client name service configuration are dynamic; therefore, no reboot is needed.

```
# ldapclient uninit
System successfully unconfigured
```

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Configuring a System to Use DNS and LDAP (Level 1)

In this exercise, you configure the Solaris 9 OE client system to use DNS and LDAP as name services.

Preparation

Refer to the lecture notes to perform the tasks listed. The instructor's system is configured as a DNS server and as an LDAP server for the classroom network, using a domain name of `suned.sun.com`.

Tasks

Perform the following tasks:

- Configure your system to use DNS, and verify that you can resolve other systems in your domain.
- Configure the system to be an LDAP client, and verify that you can resolve other systems in the classroom network.

Exercise: Configuring a System to Use DNS and LDAP (Level 2)

In this exercise, you configure the Solaris 9 OE client system to use DNS and LDAP as name services.

Preparation

Refer to the lecture notes to perform the tasks listed. The instructor's system is configured as a DNS server and as an LDAP server for the classroom network, using a domain name of `suned.sun.com`.

Task Summary

Perform the following tasks:

- Configure your system to use DNS and verify that you can resolve other systems in your domain.
- Configure the system to be an LDAP client and verify that you can resolve other systems in the classroom network.

Tasks

Complete the following steps:

1. Add DNS to the name service by copying the `/etc/nsswitch.dns` file to the `/etc/nsswitch.conf` file.
2. Create the `/etc/resolv.conf` file, and:
   a. Add a name server directive by using the address `192.168.30.30`.
   b. Add a domain directive by using `suned.sun.com`.
3. Verify that you can access another system in the classroom by using the `ping` command. First, use only the host name, and then use the fully qualified domain name – *`hostname`*`.suned.sun.com`.

4. Use the `ldapclient` command to initialize the system. The name of the profile is `default`.
5. Verify the name service switch file has been updated with the LDAP configuration.
6. Verify that you can access another system in the classroom by using the `ping` command.
7. Display the directory information tree (DIT) containers.
8. Display the Hosts container.
9. Unconfigure the LDAP client.
10. Verify the LDAP configuration has been removed from the name service switch file.

Exercise: Configuring a System to Use DNS and LDAP (Level 3)

In this exercise, you configure the Solaris 9 OE client system to use DNS and LDAP as name services.

Preparation

Refer to the lecture notes to perform the tasks listed. The instructor's system is configured as a DNS server and as an LDAP server for the classroom network, using a domain name of `suned.sun.com`.

Task Summary

Perform the following tasks:

- Configure your system to use DNS and verify that you can resolve other systems in your domain.
- Configure the system to be an LDAP client and verify that you can resolve other systems in the classroom network.

Tasks and Solutions

Complete the following steps:

1. Add DNS to the name service by copying the `/etc/nsswitch.dns` file to the `/etc/nsswitch.conf` file.

```
# cp /etc/nsswitch.dns /etc/nsswitch.conf
```

2. Create the `/etc/resolv.conf` file, and:

   a. Add a name server directive by using the address `192.168.30.30`.

   b. Add a domain directive by using `suned.sun.com`.

```
# vi /etc/resolv.conf
```

Use `vi` *to create the* `/etc/resolv.conf` *file, and insert the following lines:*

```
nameserver 192.168.30.30
domain suned.sun.com
```

3. Verify that you can access another system in the classroom by using the `ping` command. First, use only the host name, and then use the fully qualified domain name – *`hostname`*`.suned.sun.com`.

```
# ping sys11
sys11 is alive
# ping sys11.suned.sun.com
sys11.suned.sun.com is alive
```

4. Use the `ldapclient` command to initialize the system. The name of the profile is `default`.

```
# ldapclient -v init -a proxyPassword=proxy \
-a proxyDN=cn=proxyagent,ou=profile,dc=suned,dc=sun,dc=com \
-a domainname=suned.sun.com 192.168.30.30
```

5. Verify the name service switch file has been updated with the LDAP configuration.

```
# more /etc/nsswitch.conf
```

6. Verify that you can access another system in the classroom by using the `ping` command.

```
# ping sys11
sys11 is alive
```

7. Display the DIT containers.

```
# ldaplist
```

8. Display the Hosts container.

```
# ldaplist hosts
```

9. Unconfigure the LDAP client.

```
# ldapclient -v uninit
```

10. Verify the LDAP configuration has been removed from the name service switch file.

```
# more /etc/nsswitch.conf
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 16

Configuring the Network Information Service (NIS)

Objectives

Network Information Service (NIS) enables you to create central repositories for administrative files on server systems within a single UNIX domain. The NIS client-server relationship requires that each system must be configured as an NIS client and that at least one system must be configured as an NIS master server.

Upon completion of this module, you should be able to:

- Describe NIS fundamentals
- Configure the name service switch file
- Describe NIS security
- Configure an NIS domain
- Build custom NIS maps
- Troubleshoot NIS

The following course map shows how this module fits into the current instructional goal.

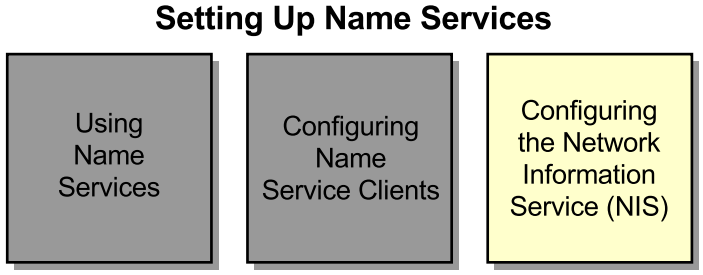


Figure 16-1 Course Map

Introducing NIS Fundamentals

NIS facilitates the creation of server systems that act as central repositories for several of the administrative files found on UNIX systems. The benefits of NIS include:

- Centralized administration of files
- Better scaling of file administration as networks grow

Figure 16-2 shows that NIS is organized into named administrative domains. Conceptually, within each domain there is one NIS master server, zero or more slave servers, and one or more clients.

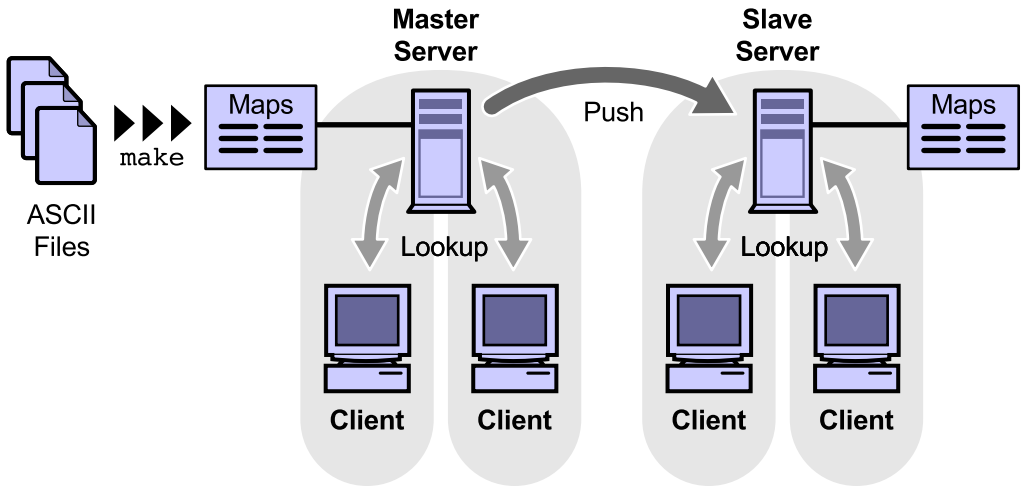


Figure 16-2 NIS Domains

NIS Namespace Information

NIS makes network administration more manageable by providing centralized control over a variety of network information. NIS stores information about host names and their IP addresses, users, the network itself, and network services. This collection of network information is called the NIS namespace.

NIS maps can replace or be used with the configuration files that exist on each UNIX system.

NIS maps are located in the `/var/yp/`*`domainname`* directory (where *`domainname`* is the name of the NIS domain). There are two files (`.pag` and `.dir` files) for each map in this directory.

Map Contents and Sort Keys

Each map contains a *key* and *value* pair. The *key* represents data used to perform the lookup in the map, while the *value* represents data returned after a successful lookup. The maps are the results of sorting the data based on different keys.

For example, the /var/yp/*domainname*/hosts.byaddr.pag map contains the data for the hosts map indexed by host IP addresses. Similarly, the /var/yp/*domainname*/hosts.byname.pag map contains the same host data using the host name as the lookup key. For the domain name training, the NIS map files list for the hosts map are:

- The /var/yp/training/hosts.byname.pag file
- The /var/yp/training/hosts.byname.dir file
- The /var/yp/training/hosts.byaddr.pag file
- The /var/yp/training/hosts.byaddr.dir file

The syntax for the NIS maps is:

map.*key*.pag and *map*.*key*.dir

where:

map	The base name of the map (hosts, passwd, and so on).
key	The map's sort key (byname, byaddr, and so on).
pag	The map's data.
dir	An index to the *.pag file. If the *.pag file is large if the *.pag file is small, the *.dir file might be empty.

Commands to Read Maps

You can use two commands to read maps:

- `ypcat [ -k ]` *`mname`* – The `ypcat` command prints out values in the NIS name service map specified by the *`mname`* argument, which can be either a map name or a map nickname.

```
# ypcat hosts
localhost 127.0.0.1     localhost
sysprint 192.168.30.70  sysprint
sys44 192.168.30.44     sys44 loghost
sys43 192.168.30.43     sys43
sys42 192.168.30.42     sys42
sys41 192.168.30.41     sys41
```

- `ypmatch [ -k ]` *`value mname`* – The `ypmatch` command prints the values associated with one or more keys from the NIS name services map specified by the *`mname`* argument, which can be either a map name or a map nickname.

```
# ypmatch sys44 hosts
sys44: 192.168.30.44    sys44 loghost
# ypmatch usera passwd
usera: usera:LojyTdiQev5i2:3001:10::/export/home/usera:/bin/ksh
```

NIS Domains

An NIS domain is a collection of hosts and interconnecting networks that are organized into a single administrative authority. NIS uses domains to arrange the hosts, users, and networks in its namespace. An NIS namespace does not use a domain hierarchy. Each NIS domain contains:

- One NIS master server
- NIS slave servers (optional)
- NIS clients

The NIS Master Server

Within each domain, the NIS master server:

- Contains the original `/etc` ASCII files used to build the NIS maps
- Contains the NIS maps generated from the ASCII files
- Provides a single point-of-control for the entire NIS domain

NIS Slave Servers

Within each domain, the NIS slave servers:

- Do not contain the original `/etc` ASCII files used to build the NIS maps
- Contain copies of the NIS maps copied from the NIS master server
- Provide a backup repository for NIS map information
- Provide redundancy in case of server failures
- Provide load sharing on large networks

NIS Clients

Within each domain, the NIS clients:

- Do not contain the original `/etc` ASCII files used to build the NIS maps
- Do not contain any NIS maps
- Bind to the master server or to a slave server to obtain access to the administrative file information contained in that server's NIS maps
- Dynamically rebind to another server in case of server failure
- Make all appropriate system calls aware of NIS

Note – All hosts in the NIS environment are clients. All NIS clients that are configured as NIS master server and NIS slave servers contain copies of the NIS maps to support the server function.

NIS Processes

The main daemons involved in the running of an NIS domain are:

- The `ypserv` daemon
- The `ypbind` daemon
- The `rpc.yppasswdd` daemon
- The `ypxfrd` daemon
- The `rpc.ypupdated` daemon

Figure 16-3 shows a domain and its NIS daemons.

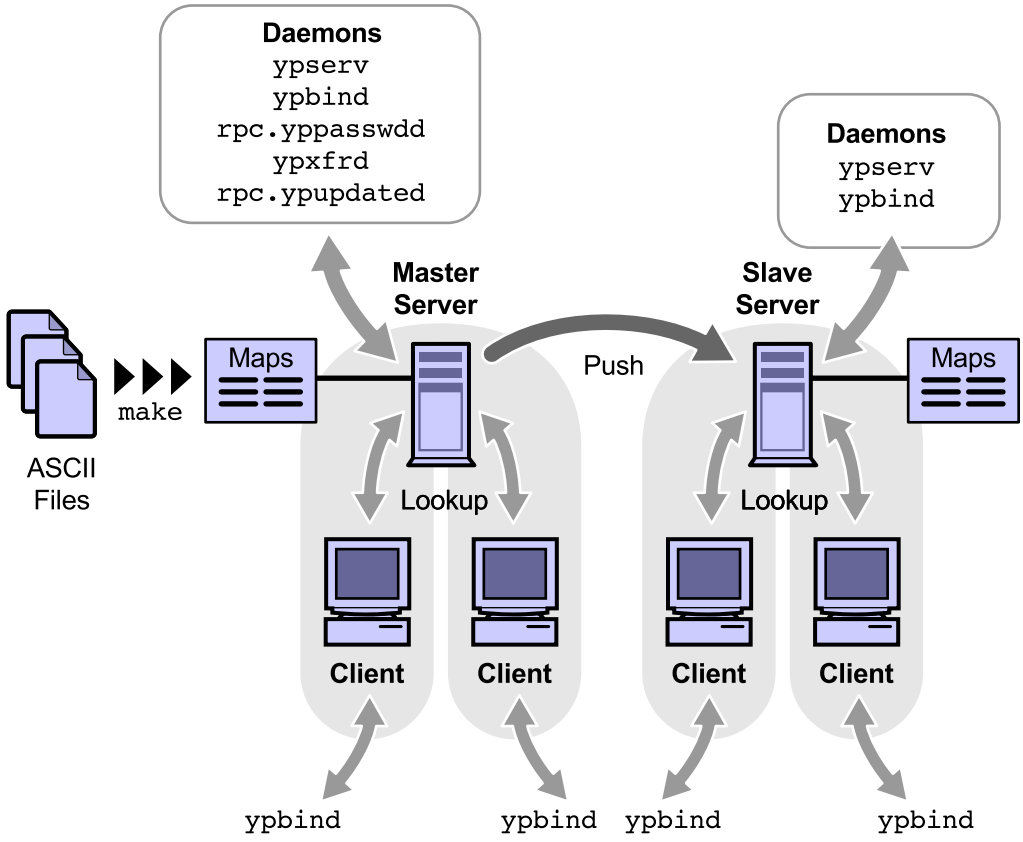


Figure 16-3 NIS Processes and Daemons

The ypserv Daemon

The ypserv daemon:

- Runs on master and slave servers
- Answers ypbind requests from clients
- Responds to client information requests

The ypbind Daemon

The ypbind daemon:

- Runs on all NIS client systems
- Makes initial client-to-server binding requests
- Stores binding information in the /var/yp/binding/*domainname* directory
- Rebinds to another server if the connection is lost with the initial server
- Requests NIS map information at the library-call level

The rpc.yppasswdd Daemon

The rpc.yppasswdd daemon:

- Allows users to change their passwords
- Updates the passwd and shadow files on the master server
- Updates the NIS password map
- Provides or "pushes" the NIS password map to all slave servers

The ypxfrd Daemon

The ypxfrd daemon:

- Runs on the NIS master server only
- Responds to requests, generated in the slave servers by using the ypxfr command to pull the maps from the master
- Transfers NIS maps at a high speed

The rpc.ypupdated Daemon

The rpc.ypupdated daemon:

- Runs on the NIS master server only
- Updates NIS maps using the configuration stored in the /var/yp/updaters file

Note – The rpc.ypupdated daemon and the /var/yp/updaters file relate to systems running secure Remote Procedure Call (RPC) services. By default, the updating master's Makefile is not used to authenticate changing any conventional NIS maps.

Configuring the Name Service Switch

The name service switch is a file named `/etc/nsswitch.conf`. This file controls how a client host or application obtains network information. A template file is provided for each of the Solaris OE name services to assist you in configuring the respective name services. When you select NIS as the name service, the `etc/nsswitch.nis` configuration file loads into the default `etc/nsswitch.conf` file.

```
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

# the following two lines obviate the "+" entry in /etc/passwd and
/etc/group.
passwd:     files nis
group:      files nis

# consult /etc "files" only if nis is down.
hosts:      nis [NOTFOUND=return] files
ipnodes:    files
# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
#ipnodes:    nis [NOTFOUND=return] files

networks:   nis [NOTFOUND=return] files
protocols:  nis [NOTFOUND=return] files
rpc:        nis [NOTFOUND=return] files
ethers:     nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files
bootparams: nis [NOTFOUND=return] files
publickey:  nis [NOTFOUND=return] files

netgroup:   nis

automount:  files nis
aliases:    files nis
```

```
# for efficient getservbyname() avoid nis
services:   files nis
sendmailvars:   files
printers:       user files nis

auth_attr:  files nis
prof_attr:  files nis
project:    files nis
```

The name service switch file is a database list. Each entry is followed by ordered lists of information that help locate specific information from the respective databases. Although you can customize the `nsswitch.conf` file to specify any search order, the three most common search orders are:

- Search files and then NIS
- Search NIS and then files
- Forward host lookup requests from NIS to DNS

Changing Lookup Requests to Go From Files to NIS

A default `/etc/nsswitch.nis` file is provided with the Solaris 9 OE. This file helps specific databases send lookup requests to local files and then to NIS maps:

```
passwd:     files nis
group:      files nis
automount:  files nis
aliases:    files nis
services:   files nis
auth_attr:  files nis
prof_attr:  files nis
project:    files nis
```

Using the `passwd` database as an example, the entry states that user information lookup is performed first by using the `/etc/passwd` and `/etc/shadow` files. If the information does not exist in these local files, then the password lookup requests search the NIS maps on the NIS server.

Changing Lookup Requests to Go From NIS to Files

The default `/etc/nsswitch.nis` file provided with the Solaris 9 OE is also configured so that specific databases can send lookup requests first to the NIS maps and then to the local files. The databases that follow this procedure are:

```
hosts:      nis [NOTFOUND=return] files
networks:   nis [NOTFOUND=return] files
protocols:  nis [NOTFOUND=return] files
rpc:        nis [NOTFOUND=return] files
ethers:     nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files
bootparams: nis [NOTFOUND=return] files
publickey:  nis [NOTFOUND=return] files
```

Using the `hosts` database as an example, the entry states that hosts lookup requests first search the NIS maps on the NIS server. If these maps do not contain the information, then the hosts lookup requests search the `/etc/inet/hosts` file on the *client* system.

To further define this search, use a status message and a name service switch action option. The `[NOTFOUND=return]` condition works as follows:

- If the NIS maps source does not respond or is unavailable, it indicates that the map cannot be accessed. You must continue to search the local file for the map.
- If you get a "no such entry" response from the NIS maps, it indicates that the `NOTFOUND` condition is configured with the `return` action, which causes the system to stop looking for the information. Therefore, when the entry is not found in the NIS map file, stop the search.

The NIS client requests information from the NIS server as usual. If the information is not found, the NIS client requests the information from the DNS server directly. The NIS client is configured as a DNS client so that it can request the information directly from the DNS server. Therefore, you do not need to configure the `Makefile` file. Using this method, you can configure the hosts database information source in the `/etc/nsswitch.conf` file to recognize both NIS and DNS. The following line requests information first from the NIS namespace and then, if the information is not found, it searches the DNS namespace.

```
hosts:      nis dns
```

Figure 16-4 shows the process of searching NIS and DNS namespaces. If the information is not located in the NIS namespace, the NIS server returns a status of `NOTFOUND`. In the name service switch, the default action for the `NOTFOUND` status is to continue the search with the next listed information source. In this case, the next information source is DNS; therefore, the client requests the information from the DNS namespace.

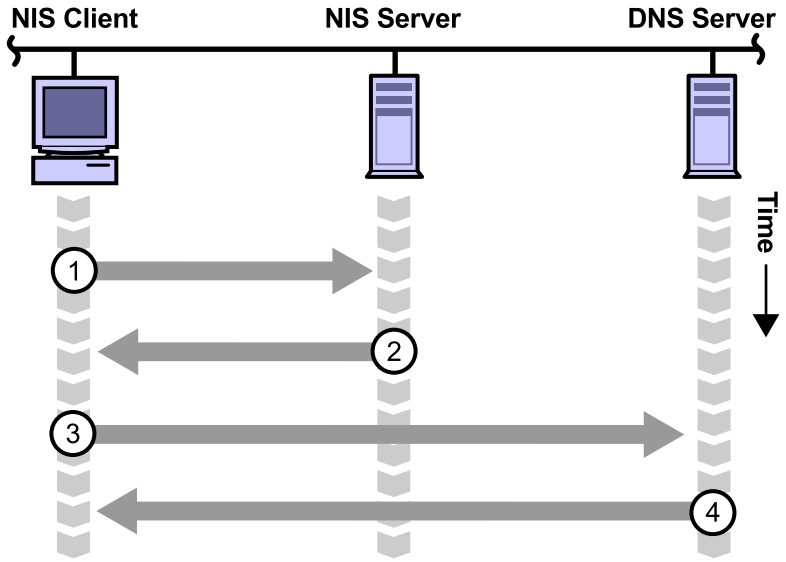


Figure 16-4 Searching NIS and DNS Namespaces

Introducing NIS Security

Just as NIS makes the network information more manageable, it can also create inadvertent security holes. Two methods of closing these security holes are using the `securenets` file to restrict access to a single host or to a subnetwork, and using the `passwd.adjunct` file to limit access to the password information across the network.

The `securenets` File

The `/var/yp/securenets` file limits access to NIS services. If the `/var/yp/securenets` file exists on an NIS server, the server only answers queries or supplies maps to hosts and networks whose IP addresses exist in the file.

The server must be able to access itself. To access itself, the server can be a part of the subnet that is allowed to access the server, or you can add the following entry:

```
host   127.0.0.1
```

The following example describes a `securenets` file

where:

- The server is configured to access itself.
- A class C network is configured for access.
- Two specific hosts, `13.13.14.1` and `13.13.14.2`, are configured to access the NIS information.

```
# Each line contains two fields separated by white space. The first field
# is a netmask, the second a network. The netmask  field  may  be either
# 255.255.255.255 (IPv4), ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff (IPv6),
# or the string ‘‘host’’ indicating  that  the  second field is a
# specific host to be allowed access.
#
# Two methods of giving access to a system are using the netmask followed
# by the IP address or using the host keyword followed by the IP address.
# 255.255.255.255   192.9.1.20
# host   192.9.1.20
#
# To allow access for an individual IPv6 address, use one of the
# following:
# ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff  fec0::111:abba:ace0:fba5e:1
```

```
# host  fec0::111:abba:ace0:fba5e:1
#
# If access is to be given to an entire class C network, the entry could
# be:
#255.255.255.0  192.9.1.0
#
#The entry for access to a class B network could be:
# 255.255.0.0    9.9.0.0
#
# To allow access for all IPv6 addresses starting with fe80, use:
# ffff::  fe80::#
#
host   127.0.0.1
255.255.255.0   150.10.1.0
host   13.13.14.1
host   13.13.14.2
```

If you modify entries in the `/var/yp/securenets` file, you must kill and restart the `ypserv` and `ypxfrd` daemons. To restart the daemons, stop and restart the NIS services with:

```
# /usr/lib/netsvc/yp/ypstop
# /usr/lib/netsvc/yp/ypstart
```

The passwd.adjunct File

The `passwd.adjunct` file prevents unauthorized users from seeing the encrypted passwords that normally form part of the output when viewing the NIS `passwd` maps.

Encrypted passwords are normally hidden from the user in the `/etc/shadow` file. With the default NIS configuration, however, the encrypted password string is shown as part of the `passwd` maps.

The following example shows that the user `passwd` is hidden from view when viewing the `/etc/passwd` file:

```
# cat /etc/passwd |grep usera
usera:x:3001:10::/export/home/usera:/bin/ksh
```

When the `ypmatch` command runs against the `usera` account value in the `passwd` map, the following output appears:

```
# ypmatch -k usera passwd
usera: usera:LojyTdiQev5i2:3001:10::/export/home/usera:/bin/ksh
```

The encrypted user password is included as part of the NIS `passwd` maps. To maintain the same security, the system configures the `passwd.adjunct` file. The `passwd.adjunct` file contains the account name preceded by ## in the password field. Subsequent attempts to gain account information, using the `ypcat` or `ypmatch` commands, returns the password entry from the `passwd.adjunct` file, as follows:

```
# ypmatch -k usera passwd
usera: usera:##usera:3001:10::/export/home/usera:/bin/ksh
```

One method to enable the `passwd.adjunct` file is to follow the procedures to configure C2 security features. These procedures are located on the SunSolve℠ Web site at `http://sunsolve.sun.com`.

Configuring NIS Domain

To generate NIS maps, you need the source files. You can find source files in the `/etc` directory on the master server. Sometimes copies of the source files are found in an alternative directory. Do not keep the source files in `/etc` directory, because the contents of the maps are then the same as the contents of the local files that control access to the master server. This is a special problem for the `/etc/passwd` and `/etc/shadow` files, because all users would have access to the master server's `root` password that would be available to all NIS clients through the `passwd` map.

To locate the source files in another directory, modify the `/var/yp/Makefile` file:

- Change the `DIR=/etc` line to `DIR=/`*`your-choice`*
- Change the `PWDIR=/etc` line to `PWDIR=/`*`your-choice`*

where *`your-choice`* is the name of the directory that you are using to store the source files. This process enables you to keep the local files on the server separate from those files used for NIS.

Caution – Before you make any modifications to the `/var/yp/Makefile` file, save a copy of the original `Makefile` file.

Generating NIS Maps

The NIS configuration script, `/usr/sbin/ypinit`, and the `make` utility generate NIS maps. The `ypinit` command reads the `/var/yp/Makefile` file for source file locations, and converts ASCII source files into NIS maps.

Note – For security reasons and to prevent unauthorized root access, the files that build the NIS password maps should not contain an entry for the `root` user. To make sure of this, copy the files to an alternative directory, and modify the `PWDIR` entry in the `Makefile` file.

Locating Source Files

The source files are located in the `/etc` directory on the master server, but the files can be copied into some other directory (such as `/etc/yp_dir` in the Figure 16-5).

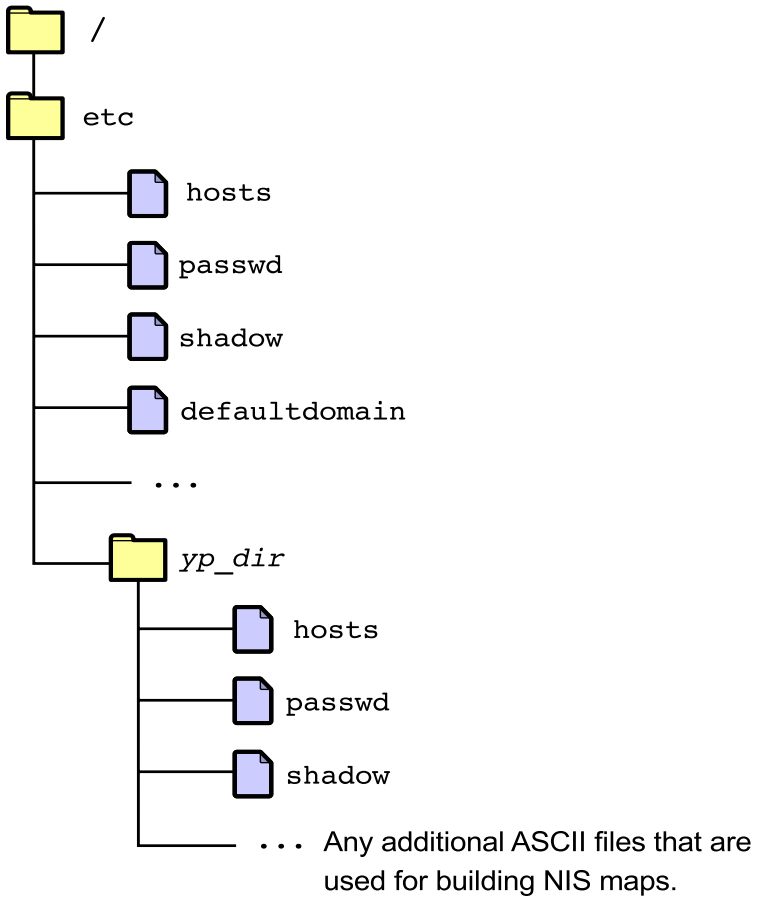


Figure 16-5 Important Files on the NIS Master (Part 1)

Figure 16-5 also shows the location of the `defaultdomain` file that resides in the `/etc` directory. The `/etc/defaultdomain` file sets the NIS domain name during system boot.

The `ypinit` script calls the program `make`, which uses the `Makefile` file located in the `/var/yp` directory. Figure 16-6 shows a default `Makefile` in the `/var/yp` directory, which contains the commands needed to transform the source files into `ndbm` format maps.

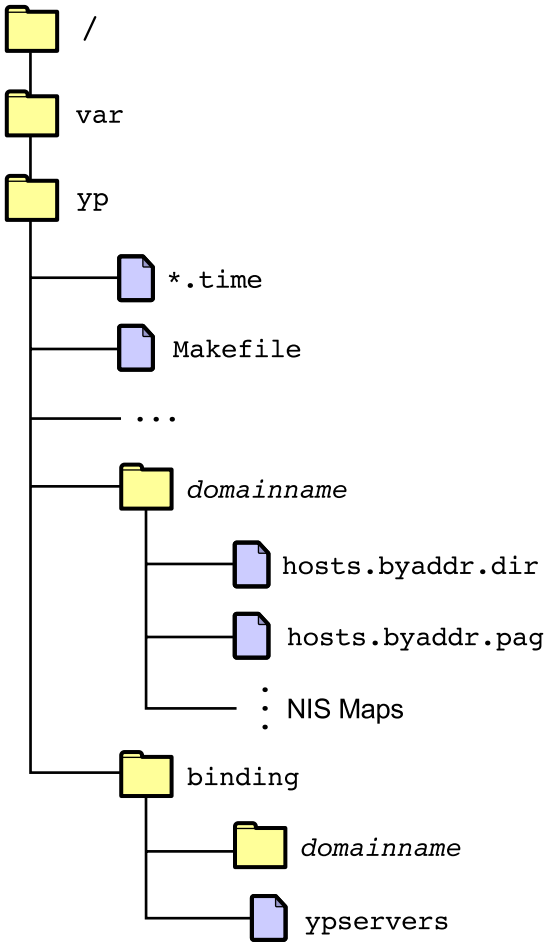


Figure 16-6 Important Files on the NIS Master (Part 2)

The `/var/yp` directory contains a subdirectory named after the NIS domain name. This *`domainname`* directory is the repository for the NIS maps created by the `ypinit` script. The `/var/yp/binding/`*`domainname`* directory contains the `ypservers` file where the names of the NIS master server and NIS slave servers are stored.

Figure 16-7 shows that the /usr/lib/netsvc/yp directory contains the ypstop and ypstart commands that stop and start NIS services, respectively.

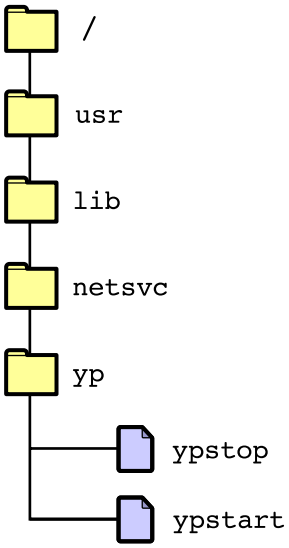


Figure 16-7 Important Files on the NIS Master (Part 3)

Dependencies of the NIS Makefile File

The NIS Makefile works by using a set of dependencies. When the make command is executed, it is effectively a make all command. The second section of the Makefile contains the target line all, which determines which maps are built. The all target entries are matched with dependency entries in the fourth section of the Makefile to match them with the final dependencies that define which code segments in the section three of the Makefile are executed to construct the specified NIS maps. Section one of the Makefile contains macros that are called out in section three. These macros redirect the make utility to alternate locations of source file when you choose to use a source file directory other than the /etc directory.

Note – These sections of the Makefile are described in detail later in this module.

Converting ASCII Source Files Into NIS Maps

To build new maps on the master server, perform the command:

```
# /usr/sbin/ypinit -m
```

The `ypinit` command prompts for a list of other machines to become NIS slave servers. Type the name of the server on which you are working, along with the names of your NIS slave servers. The `ypinit` command asks whether you want the procedure to terminate at the first nonfatal error or to continue despite nonfatal errors. The `ypinit` command asks whether the existing files in the `/var/yp/domainname` directory can be destroyed. This message is displayed only if NIS has been previously installed. You must answer yes to install a new version of NIS maps. After the `ypinit` command has constructed the list of servers, it invokes the `make` command.

This program uses the instructions contained in the `Makefile` file (either the default one or the one you modified) located in the `/var/yp` directory. The `make` command strips any remaining comment lines from the source files and runs the `makedbm` function on them, creating the appropriate maps and establishing the name of the master server in each map.

Configuring the NIS Master Server

Note – Installations that select Core, End User, or Developer software configuration clusters do not have all of the necessary files in the `/usr/lib/netsvc/yp` directory to allow a host to function as an NIS server.

To set up the NIS name service master server, perform the following steps:

1. Determine which machines on your network domain will be NIS servers. There can be one NIS master server and as many NIS slave servers as needed. All systems within the domain are NIS clients.
2. Copy the `/etc/nsswitch.nis` file to the `/etc/nsswitch.conf` file. If necessary, modify the file.
3. Choose an NIS domain name. This is usually less than 32 characters in length. (The maximum length is 256 characters, and it is case sensitive.)

4. Enter the `domainname` command to set the local NIS domain.

```
# domainname domainname
```

For example:

```
# domainname classroom.Central.Sun.COM
```

5. Create an `/etc/defaultdomain` file with the domain name. You must maintain the format established by the original files and update the text files in the `/etc` directory (all of the files that are used for NIS maps) on the master server with information about the domain.

Note – You can also copy the network information files to some other location on the system and modify them there rather than modifying them in the `/etc` directory.

6. If the files do not already exist, use the `touch` command to create zero-length files with the following names: `/etc/ethers`, `/etc/bootparams`, `/etc/locale`, `/etc/timezone`, `/etc/netgroup`, and `/etc/netmasks`. These files are necessary for the creation of the complete set of NIS maps as directed in the `Makefile` file. When you initialize NIS, you receive error messages for each of these files if they do not exist.
7. Install an updated `Makefile` file in the `/var/yp` directory if you intend to use NIS on the system that functions as your JumpStart™ server. Doing this installation provides entries that create a map for the `/etc/locale` file.

Note – The lab at the end of this module shows you how to create the updated `Makefile` file.

8. Create or populate the `/etc/locale` file, and make an entry for each domain on your network using the following format:

```
domainname      locale
```

For example:

```
classroom.Central.Sun.COM      en_US
```

9. Initialize the master server by using the local `/etc` files. Enter the `ypinit -m` command.

```
# ypinit -m
```

   a. When the program prompts you for a list of slave servers and after you complete your list, press Control-D. You can make entries for all slaves now, or you can rerun the `ypinit -m` command after you determine whether you need more or less slave servers.

   b. The program asks if you want to terminate it on the first fatal error. If you answer `n`, the procedure reports any error and attempts to complete the creation of the NIS database files. If you answer `y`, the process aborts with the first error. You can correct the error and restart the `ypinit` program.

   The following example shows the text feedback displayed as the program begins:

```
# ypinit -m
In order for NIS to operate successfully, we have to construct a list of
the NIS servers. Please continue to add the names for YP servers in order
of preference, one per line. When you are done with the list, type a
<control D> or a return on a line by itself.
                next host to add: server1
                next host to add: <Control-D>
The current list of yp servers looks like this:
server1
Is this correct?  [y/n: y]  y
Installing the YP database will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
OK, please remember to go back and redo manually whatever fails.  If you
don't, some part of the system (perhaps the yp itself) won't work.
```

Note – If you have to restart the `ypinit` program, you are prompted to destroy the `/var/yp/`*`domainname`* directory. Answer `y`.

10. Start the NIS daemons on the master server with the following command:

```
# /usr/lib/netsvc/yp/ypstart
```

11. If you want to stop the NIS service running on the NIS master, perform the command:

```
# /usr/lib/netsvc/yp/ypstop
```

Testing the NIS Service

There are a number of commands that you can use to obtain information from and about the NIS database. You can also use these commands to test the functionality of the NIS service. You do not have to be the superuser to use these commands.

The most commonly used NIS commands are:

`ypcat`	Prints values from an NIS map
`ypmatch`	Prints the value of one or more keys from an NIS map
`ypwhich`	Returns the name of the NIS server that supplies the NIS map services to an NIS client

Using the `ypcat` Command

The following example prints the information from the `hosts` database.

```
$ ypcat hosts
192.168.30.30   instructor instructor1
127.0.0.1       localhost loghost
192.168.30.45   sys45
192.168.30.44   sys44
192.168.30.43   sys43
192.168.30.42   sys42
192.168.30.41   sys41
...
<output truncated>
...
```

Using the `ypmatch` Command

The following example matches individual host entries.

```
# ypmatch sys41 localhost hosts
192.168.30.41   sys41
127.0.0.1       localhost loghost
```

The following example matches a specific user in the password database.

```
# ypmatch user5 passwd
user5:.dJJ.oofIqCLs:4005:10::/export/home/user5:/bin/ksh
```

Using the ypwhich Command

Perform the ypwhich command to identify the master server:

```
$ ypwhich
sys44
```

When used with the -m option, the ypwhich command provides a list of all databases and the name of the master server for each map.

```
$ ypwhich -m
...
<output truncated>
...
timezone.byname sys44
netmasks.byaddr sys44
netid.byname sys44
bootparams sys44
netgroup.byhost sys44
netgroup.byuser sys44
netgroup sys44
...
<output truncated>
...
```

Configuring the NIS Client

All systems within an NIS domain that are not configured as servers are configured as clients. To configure the NIS client, complete the following steps:

1. Copy the /etc/nsswitch.nis file to the /etc/nsswitch.conf file. If necessary, modify the file.
2. Edit the /etc/hosts file to ensure that the NIS master server and all slave servers have been defined.
3. Execute the domainname *domainname* command to set the local NIS domain.

```
# domainname domainname
```

For example:

```
# domainname classroom.Central.Sun.COM
```

4. Create or populate the `/etc/defaultdomain` file with the domain name.
5. To initialize the system as an NIS client, perform the command:

```
# ypinit -c
```

6. When the system prompts you for a list of NIS servers, enter the names of the NIS master and all slave servers.

Note – To exit the `ypinit` command without building a specific list of NIS servers, press Control-D. The client then broadcasts to bind the first available server during subsequent `ypbind` operations. When not operating in broadcast mode, clients can only bind to servers that are listed in their `/var/yp/binding/`*`domainname`*`/ypservers` file.

---

7. Start NIS with the following command:

```
# /usr/lib/netsvc/yp/ypstart
```

8. On the newly configured NIS client, test the NIS functionality by performing the command:

```
# ypwhich -m
```

The output shows a list of maps together with the NIS master server for each map.

Configuring the NIS Slave Server

You should have at least one NIS slave server to provide backup if the NIS master server becomes unavailable. To configure an NIS slave server, complete the following steps on the system that you want to designate as the slave server:

1. Copy the `/etc/nsswitch.nis` file to the `/etc/nsswitch.conf` file. If necessary, modify the file.
2. Edit the `/etc/hosts` file to ensure that the NIS master and all NIS slave servers have been defined.
3. Execute the `domainname` command to set the local NIS domain.

```
# domainname domainname
```

For example:

```
# domainname classroom.Central.Sun.COM
```

4. Create or populate the `/etc/defaultdomain` file with the domain name.

5. Initialize the system as an NIS client by performing the command:

```
# ypinit -c
```

6. When the system prompts for a list of NIS servers, enter the NIS master host followed by the name of the local host and all other NIS slave servers on the local network.

7. On the NIS master, ensure that the `ypserv` process is running by performing the command:

```
# ps -ef | grep ypserv
```

If it is not running, refer to the previous section on how to start NIS daemons on the master.

8. Return to the proposed NIS slave system, and enter the `ypstart` command to start the `ypbind` daemon.

```
# /usr/lib/netsvc/yp/ypstart
```

9. Initialize the system as an NIS slave by performing the command:

```
# ypinit -s master
```

where ***master*** is the name of the NIS master.

Note – If you did not add the name of the NIS slave server when you initially configured the NIS master server using the `ypinit` command, enter the `ypinit -m` command once more on the NIS master server. In the process of updating the NIS master, the script prompts you for confirmation when it is about to destroy the existing domain database. Confirm by entering `y`.

10. To start the `ypserv` daemon on the slave server, perform the command:

```
# /usr/lib/netsvc/yp/ypstart
```

11. To test NIS functionality on the newly configured NIS slave server, perform the command:

```
# ypwhich -m
```

The output shows a list of maps together with the NIS master server for each map.

Updating the NIS Map

Because database files change with time, you must update your NIS maps. To update the NIS maps (on the master server), complete the following steps:

1. Update the text files in your source directory (typically, `/etc`, unless it was changed in the `Makefile` file).
2. Change to the `/var/yp` directory.

```
# cd /var/yp
```

3. Refresh the NIS database maps using the `make` utility.

```
# /usr/ccs/bin/make
```

Updating the NIS Password Map

If the NIS master is running the `rpc.yppasswdd` daemon, any client system can update the NIS password map by using the `yppasswd` or `passwd` commands, as shown in Figure 16-8.

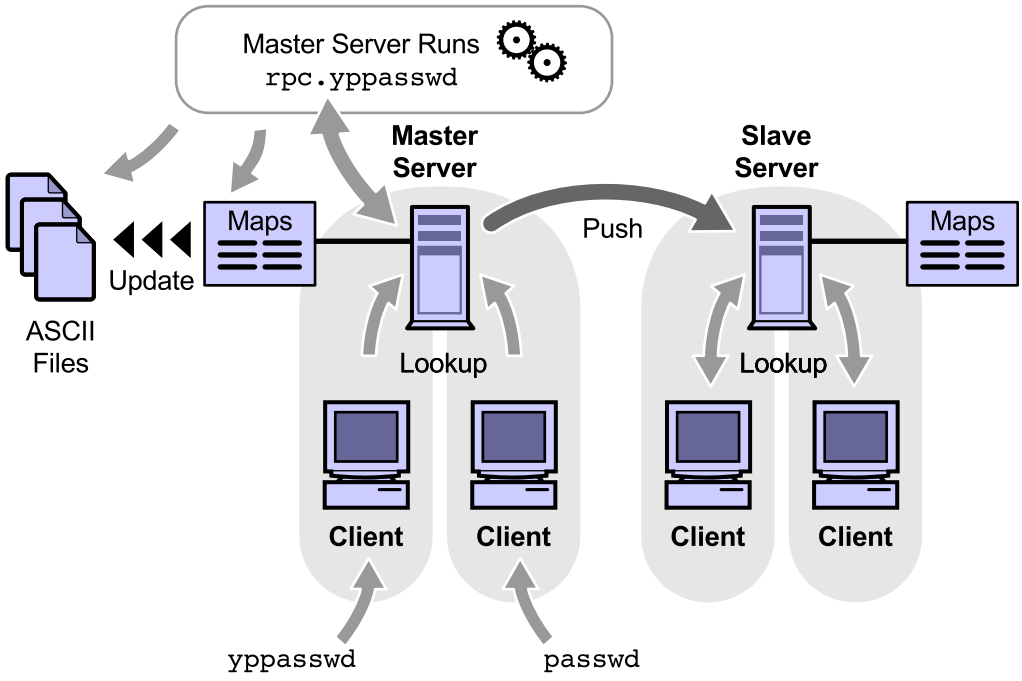


Figure 16-8 Updating the NIS Password Map

To update the password map complete the following steps:

1. Run the `rpc.yppasswdd` daemon on the NIS master server

```
# /usr/lib/netsvc/yp/rpc.yppasswdd /$PWDIR/passwd -m passwd
```

When users change their NIS passwords, the `rpc.yppasswdd` daemon updates the NIS master's */$PWDIR/*`passwd` file and `passwd` map. The `passwd` map is then pushed to all slave servers.

2. Enter the `passwd` command on any NIS client.

```
$ passwd
Changing NIS password for user1 on server1.
Old password:
New password:
Retype new password:
NIS entry changed on server1
```

Updating the NIS Slave Server Map

The following steps manually update the NIS `timezone` map on the master server and propagate all maps to the slave servers:

1. Edit the source file on the NIS master.

```
# vi /etc/timezone
```

2. Remake and push the NIS maps to the slave servers.

```
# cd /var/yp; /usr/ccs/bin/make
```

a. If the push from the master fails, the following commands run on the slave server and manually "pull" only the `timezone` map from the master server.

```
# /usr/lib/netsvc/yp/ypxfr timezone.byname
```

b. To pull all of the maps from the master server at once, perform the command:

```
# ypinit -s nis_master
```

Sometimes maps fail to propagate, and you must manually use the `ypxfr` command to retrieve new map information. To automate the updating and propagating of NIS maps on slave servers, you can install shell scripts to run as `cron` jobs. Because maps have different rates of change, scheduling a map transfer by using the `crontab` command enables you to set specific propagation intervals for individual maps.

The Solaris OE provides several template scripts in the `/usr/lib/netsvc/yp` directory that you can use and modify to meet your local site requirements. These scripts are useful when slave servers are down during NIS map propagations.

Figure 16-9 shows you how to update `passwd` maps using slave servers with scripts. When slave servers are down, they might not receive the update unless you run a "safety valve" script.

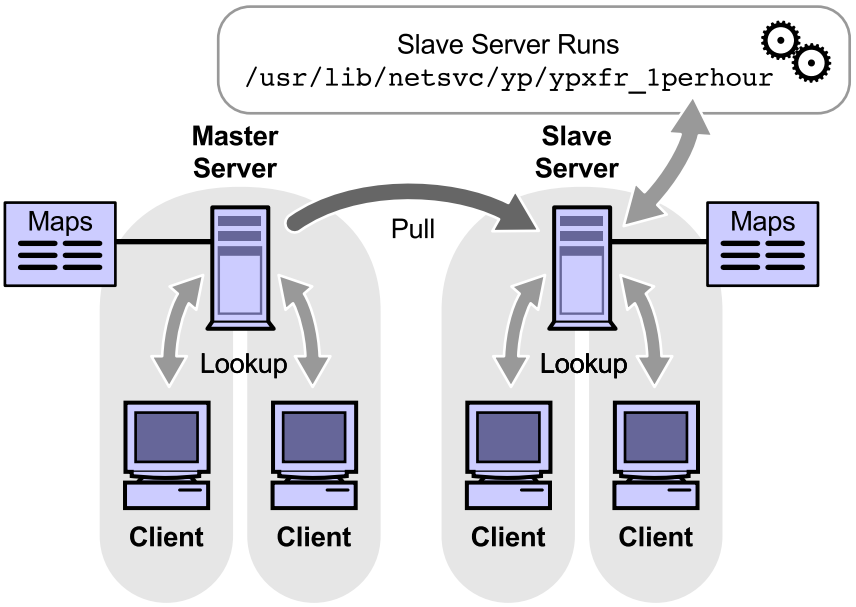


Figure 16-9 Updating `passwd` Maps on Slave Servers With Scripts

Using the ypxfr_1perhour Script

The following text lists the contents of the ypxfr_1perhour script that, if run hourly using the cron daemon, ensures that the NIS slave server's passwd map is never more than one hour out of date.

```
#! /bin/sh
#
# Copyr 1990 Sun Microsystems, Inc.
#ident  "@(#)ypxfr_1perhour.sh  1.2     00/05/01 SMI"
#
# ypxfr_1perhour.sh - Do hourly NIS map check/updates
#

PATH=/bin:/usr/bin:/usr/lib/netsvc/yp:$PATH
export PATH

# set -xv
ypxfr passwd.byname
ypxfr passwd.byuid
```

Using the ypxfr_1perday Script

The following output details the contents of the ypxfr_1perday script. If run daily using the cron daemon, the script ensures that the NIS slave server's NIS maps for the group, protocols, networks, services, and ypservers keys are never more than one day out of date.

```
#! /bin/sh
#
# Copyr 1990 Sun Microsystems, Inc.
#ident  "@(#)ypxfr_1perday.sh   1.2     00/05/01 SMI"
#
# ypxfr_1perday.sh - Do daily NIS map check/updates
#

PATH=/bin:/usr/bin:/usr/lib/netsvc/yp:$PATH
export PATH

# set -xv
ypxfr group.byname
ypxfr group.bygid
ypxfr protocols.byname
ypxfr protocols.bynumber
ypxfr networks.byname
ypxfr networks.byaddr
```

```
ypxfr services.byname
ypxfr ypservers
```

Using the ypxfr_2perday Script

The following output details the contents of the `ypxfr_2perday` script. If run twice daily using the `cron` daemon, the script ensures that the NIS slave server's NIS maps for the `hosts`, `ethers`, `netgroups` keys, and `mail` aliases are never more than 12 hours out of date.

```
#! /bin/sh
#
# Copyr 1990 Sun Microsystems, Inc.
#ident  "@(#)ypxfr_2perday.sh   1.2     00/05/01 SMI"
#
# ypxfr_2perday.sh - Do twice-daily NIS map check/updates
#

PATH=/bin:/usr/bin:/usr/lib/netsvc/yp:$PATH
export PATH

# set -xv
ypxfr hosts.byname
ypxfr hosts.byaddr
ypxfr ethers.byaddr
ypxfr ethers.byname
ypxfr netgroup
ypxfr netgroup.byuser
ypxfr netgroup.byhost
ypxfr mail.aliases
```

Building Custom NIS Maps

As system requirements or configurations change, you must keep the name service configuration the same as the system configuration.

Using the make Utility

You can learn how to build customized NIS maps with the `make` utility. The `make` utility:

- Is used by programmers to build programs
- Is used by administrators to build NIS maps
- Can be generalized to build customized NIS maps

Building Targets

The `make` utility receives its instructions from the `Makefile` file. The `Makefile` file uses variable definitions (called macros), targets, and dependencies.

You can use macros as variables, similar to those used in a shell script. You must define a macro at the beginning of the `Makefile` file. Prefix the name of the macro with a dollar sign ($) when using it throughout the `Makefile` file.

The `make` utility builds targets. Targets need dependencies. Dependencies can represent other targets that must be completely built before the original target is considered “made.” This structure enables you to nest the target and dependency pairs at an arbitrary depth, letting you build complex hierarchical code structures.

When making NIS maps, you should keep the target and dependency relationship very basic.

Editing the NIS Makefile File

The NIS Makefile file is located in the /var/yp directory and is composed of four main sections:

- The first section contains macro definitions.
- The second section contains the first target, all.
- The third section defines the final target and dependencies.
- The fourth section contains entries for each of the dependencies.

Configuring the Sections of Makefile

The first section of the Makefile file contains the following macro definitions:

```
#B=-b
B=
DIR =/etc
INETDIR=/etc/inet
RBACDIR=/etc/security
PWDIR =/etc
DOM = ‘domainname‘
NOPUSH = ""
ALIASES = /etc/mail/aliases
YPDIR=/usr/lib/netsvc/yp
SBINDIR=/usr/sbin
YPDBDIR=/var/yp
YPPUSH=$(YPDIR)/yppush
MAKEDBM=$(SBINDIR)/makedbm
MULTI=$(YPDIR)/multi
REVNETGROUP=$(SBINDIR)/revnetgroup
STDETHERS=$(YPDIR)/stdethers
STDHOSTS=$(YPDIR)/stdhosts
MKNETID=$(SBINDIR)/mknetid
MKALIAS=$(YPDIR)/mkalias
```

The second section of the `Makefile` file contains the first target, `all`.

```
all: passwd group hosts ipnodes ethers networks rpc services protocols \
        netgroup bootparams aliases publickey netid netmasks c2secure \
        timezone auto.master auto.home \
        auth.attr exec.attr prof.attr user.attr audit.user
```

The `all` target has several dependencies, each of which represents one of the NIS maps to be built. This feature enables the entire set of NIS maps to be built by typing:

```
# cd /var/yp; /usr/ccs/bin/make
```

The `all` target is not considered to be built until each of its targets is first built. Each of the targets for `all` depends on another target.

When adding custom maps to NIS, the name of the new map to be built should be added to the `all` target list (`auto.direct` in the following example).

```
all: passwd group hosts ipnodes ethers networks rpc services protocols \
        netgroup bootparams aliases publickey netid netmasks c2secure \
        timezone auto.master auto.home auto.direct\
        auth.attr exec.attr prof.attr user.attr audit.user
```

Note – The fourth section is covered before the third section, because the fourth section continues the dependency thread introduced by the `all` target.

The entry in the fourth section of the `Makefile` file for each of the dependencies in the `all` target is:

```
passwd: passwd.time
group: group.time
project: project.time
hosts: hosts.time
ipnodes: ipnodes.time
ethers: ethers.time
networks: networks.time
rpc: rpc.time
services: services.time
protocols: protocols.time
netgroup: netgroup.time
bootparams: bootparams.time
aliases: aliases.time
publickey: publickey.time
netid: netid.time
passwd.adjunct: passwd.adjunct.time
group.adjunct: group.adjunct.time
netmasks: netmasks.time
timezone: timezone.time
auto.master: auto.master.time
auto.home: auto.home.time
auth.attr:auth.attr.time
exec.attr:exec.attr.time
prof.attr:prof.attr.time
user.attr:user.attr.time
audit.user:audit.user.time
$(DIR)/netid:
$(DIR)/timezone:
$(DIR)/auto_master:
$(DIR)/auto_home:
$(PWDIR)/shadow:
$(DIR)/auth_attr:
$(DIR)/exec_attr:
$(DIR)/prof_attr:
$(DIR)/user_attr:
$(DIR)/audit_user:
```

Using the previous example of an `auto.direct` map, add a new map to the NIS domain by appending the appropriate entries to the end of this "second level" target and dependency pair.

```
...
auto.direct: auto.direct.time
...
$(DIR)/auto_direct:
```

After you modify the `auto.direct` map, the final lines from the fourth section of the `Makefile` file would look like:

```
...
auto.master: auto.master.time
auto.home: auto.home.time
auto.direct: auto.direct.time
auth.attr:auth.attr.time
exec.attr:exec.attr.time
prof.attr:prof.attr.time
user.attr:user.attr.time
audit.user:audit.user.time
$(DIR)/netid:
$(DIR)/timezone:
$(DIR)/auto_master:
$(DIR)/auto_home:
$(DIR)/auto_direct:
$(PWDIR)/shadow:
...
```

The target is the `auto.direct` map, which depends on the `auto.direct.time` target.

The third section of the `Makefile` file defines the final target and dependencies, as well as instructions on how to build each map in the domain.

Edit the `Makefile` file by adding the following lines to build a new `auto_direct` map:

```
auto.direct.time:  $(DIR)/auto_direct
        -@if [ -f $(DIR)/auto_direct ]; then \
                sed -e "/^#/d" -e s/#.*$$// $(DIR)/auto_direct \
                | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/auto.direct; \
                touch auto.direct.time; \
                echo "updated auto.direct"; \

                if [ ! $(NOPUSH) ]; then \
```

```
                $(YPPUSH) auto.direct; \
                echo "pushed auto.direct"; \
        else \
        : ; \
        fi \
else \
        echo "couldn't find $(DIR)/auto_direct"; \
fi
```

Caution – You can copy and paste lines from a section to another map; however, the proper use of tabs and spaces in the `Makefile` file is critical. Look up the `make` command in the online manual pages for the correct usage of tabs and spaces.

Some points to consider are:

- You must indent subsequent lines of `make` instructions by using tabs.
- You can use `make` macros in the instructions.
- Instructions that begin with the at (@) sign are not echoed to the terminal screen. Removing the `@` sign is useful for debugging new instructions.
- Instructions that begin with a leading dash (–) before the `@` sign do not echo error messages to the terminal screen.

Troubleshooting NIS

If only one or two clients are experiencing symptoms that indicate NIS binding difficulty, the problems are probably on those clients. If many NIS clients are failing to bind properly, the problem probably exists on one or more of the NIS servers.

Troubleshooting NIS Server Failure Messages

This section addresses some common errors associated with NIS server configuration.

No Server Available

If your domain name is set correctly, the `ypbind` daemon is running, and you get messages indicating that the client cannot communicate with a server, it can indicate a number of different problems:

- Does the client have a `/var/yp/binding/`*`domainname`*`/ypservers` file containing a list of servers to which it can bind? If not, enter the `ypinit -c` command, and specify the servers that this client should bind to, in the order of preference.
- If the client has a `/var/yp/binding/`*`domainname`*`/ypservers` file, does it have enough servers listed in it if a couple of servers should become unavailable? If not, add additional servers to the list by using the `ypinit -c` command.
- If none of the servers listed in the client's `ypservers` file are available, the client searches for an operating server by using broadcast mode. If there is a functioning server on the client's subnet, the client will find it. If there are no functioning servers on the client's subnet, you can solve the problem in several ways:
  - If the client does not have a server on the subnet or have a route to one, install a new slave server on that subnet.
  - Make sure that your routers are configured to pass broadcast packets so that the client can use broadcast to find a server on another subnet. Use the `netstat -r` command to verify the route.
  - If there should be a working route to a server on another network, check to see if either the `in.rdisc` or `in.routed` daemons are running. If neither daemon is running, run the command `/etc/init.d/inetinit start` to start them.

Note – For reasons of security and administrative control, specify the servers that a client should bind to in the client's `ypservers` file rather than have the client search for servers through broadcasting. Broadcasting slows down the network, as well as the client, and prevents you from balancing server load by listing different servers for different clients.

- Do the servers listed in a clients `ypservers` file have entries in the `/etc/inet/hosts` file? If not, add the servers to the NIS maps hosts input file, and rebuild your maps by using the `ypinit -c` or `ypinit -s` commands.
- Is the `/etc/nsswitch.conf` file set up to consult the client's local `hosts` file in addition to NIS?

The `ypwhich` Command Displays Are Inconsistent

When you use the `ypwhich` command several times on the same client, the resulting output varies because the NIS server changes, which is normal. The binding of the NIS client to the NIS server changes over time when the network or the NIS servers are busy. Whenever possible, the network becomes stable at a point where all clients get an acceptable response time from the NIS servers. As long as your client machine gets NIS service, it does not matter where the service comes from. For example, an NIS server machine can get its own NIS services from another NIS server on the network.

Network or Servers Are Overloaded

NIS can hang if the network or NIS servers are so overloaded that the `ypserv` daemon cannot get a response back to the client `ypbind` process within the time-out period.

Under these circumstances, every client on the network experiences the same or similar problems. In most cases, the condition is temporary. The messages usually go away when the NIS server reboots and restarts the `ypserv` daemon, or when the load on the NIS servers or network itself decreases.

Server Malfunction

Make sure the servers are up and running. If you are not physically near the servers, use the `ping` *`NIS_server`* command.

NIS Daemons Not Running

If the servers are up and running and you can find a client machine behaving normally, perform the `ypwhich` command on the client, as follows:

```
# ypwhich
```

If the `ypwhich` command does not respond, kill the `ypwhich` command.

```
# pkill ypwhich
```

Log in as the `root` user on the NIS server, and check if the NIS `daemons` are running by performing the command:

```
# ps -e | grep yp
```

Note – Do not use the `-f` option with the `ps` command, because this option attempts to translate user IDs into names, which causes more name service lookup requests that might not succeed.

If either the `ypbind` or `ypserv` daemons are not running, stop and then restart the NIS services by performing the command:

```
# /usr/lib/netsvc/yp/ypstop
# /usr/lib/netsvc/yp/ypstart
```

If both the `ypserv` and `ypbind` processes are running on the NIS server, and the `ypwhich` command does not respond, the `ypserv` process has probably hung. You must restart the process. Log in as `root` on the server, and kill the `ypserv` process.

```
# pkill ypserv
```

Start the `ypserv` process by restarting the NIS services. Perform the commands:

```
# /usr/lib/netsvc/yp/ypstop
# /usr/lib/netsvc/yp/ypstart
```

Troubleshooting NIS Client Failure Messages

This section addresses some common errors associated with NIS client configuration.

Missing or Incorrect Domain Name

One client has problems, the other clients are operating normally, but `ypbind` is running on the problem client. The client might not be set to the correct domain.

On the client, perform the `domainname` command to see which domain name is set.

```
# domainname
suned.Sun.COM
```

Compare the output with the actual domain name in the `/var/yp` directory on the NIS master server. The actual NIS domain is shown as a subdirectory in the `/var/yp` directory and reported with the `domainname` command on the master server.

```
# domainname
suned.sun.com
```

If the domain name returned by running the `domainname` command on a client is not the same as the server domain name listed as a directory in the `/var/yp` directory, the domain name specified in the client's `/etc/defaultdomain` file is incorrect. Log in as superuser, and correct the client's domain name in the client's `/etc/defaultdomain` file to ensure that the domain name is correct every time the machine boots. Then reboot the machine.

Note – The domain name is case sensitive.

Client Not Bound to Server

If your domain name is set correctly, the `ypbind` daemon is running, and commands still hang, then make sure that the client is bound to a server by running the `ypwhich` command.

```
# ypwhich
NIS_server
```

The server to which this client is currently bound can be the NIS master server or any NIS slave server that answers the `ypbind` broadcast.

If you have just started the `ypbind` daemon, then enter the `ypwhich` command several times (typically, the first `ypwhich` command entry reports that the domain is not bound and the second command entry succeeds).

Performing the Exercises

You have the option to complete any one of three versions of a lab. To decide which to choose, consult the following descriptions of the levels:

- Level 1 – This version of the lab provides the least amount of guidance. Each bulleted paragraph provides a task description, but you must determine your own way of accomplishing each task.
- Level 2 – This version of the lab provides more guidance. Although each step describes what you should do, you must determine which commands (and options) to input.
- Level 3 – This version of the lab is the easiest to accomplish because each step provides exactly what you should input to the system. This level also includes the task solutions for all three levels.

Exercise: Configuring NIS (Level 1)

Perform the following tasks:

- Configure the following:
  - An NIS master server
  - An NIS slave server
  - An NIS client
- Test the dynamic rebind feature
- Add a custom map to NIS

Preparation

Choose two partners for this lab, and determine which systems to configure as the NIS master server, the NIS slave server, and the NIS client.

NIS_master: _____

NIS_slave: _____

NIS_client: _____

domainname: _____

On all systems, verify that the entries for all three hosts exist in the `/etc/hosts` file. Refer to your lecture notes as necessary to perform the steps listed.

Tasks

Perform the following tasks:

- Create and configure an NIS master server. Select an NIS domain name to use for your group of three systems. Set the domain name, and record its name in the `/etc/defaultdomain` file. Enter the `touch` command to create any files in the `/etc` directory that are required by the target `all` in the `Makefile` file. Edit the automount master map and indirect map to comment out "+" entries.
- On the system to be the NIS master server, share the `/export/home` directory by using NFS. Create three user accounts and set passwords for these users. Configure the `/etc/passwd` file and the automount indirect map to allow the users to mount their home directories from the NIS master. Use the `ypinit -m` command to initialize the NIS master. Configure the `/etc/nsswitch.conf` file for NIS, and start the NIS server daemons.
- Create and configure an NIS slave server. Set the NIS domain name to be the same as in the NIS master. Use the `ypinit -c` command to configure the system as an NIS client. Configure the `/etc/nsswitch.conf` file for NIS, and start the NIS client daemons. Use the `ypinit -s` command to configure the system as an NIS slave server. Stop and restart the NIS daemons. Verify the list of servers found in the `ypservers` map.
- Create and configure an NIS client system. Set the NIS domain name to be the same as in the NIS master. Use the `ypinit -c` command to configure the system as an NIS client. Configure the `/etc/nsswitch.conf` file for NIS, and start the NIS client daemons. Test the configuration with the `ypwhich` command.
- Test the dynamic rebind feature by stopping the NIS services on the NIS master server. Monitor the NIS client with the `ypwhich` command, and observe when the client binds to the slave server. Start the NIS services on the NIS master.
- Make the appropriate changes in the `/var/yp/Makefile` file to support a new automount direct map called `auto_direct`. Create the direct map in the `/etc` file. Configure the direct map and NFS shares to allow all three systems to automatically mount the man pages from the NIS master server.
- Test if the new users can log in on all three systems. Verify that their home directories automatically mount. Verify that the man pages are available through the `automount` service on all three systems.

Exercise: Configuring NIS (Level 2)

Perform the following tasks:

- Configure the following
  - An NIS master server
  - An NIS slave server
  - An NIS client
- Test the dynamic rebind feature
- Add a custom map to NIS

Preparation

Choose two partners for this lab, and determine which systems to configure as the NIS master server, the NIS slave server, and the NIS client.

NIS_master: _____

NIS_slave: _____

NIS_client: _____

domainname: _____

On all systems, verify that entries for all three hosts exist in the `/etc/hosts` file. Refer to your lecture notes as necessary to perform the steps listed.

Task Summary

Perform the following tasks:

- Create and configure an NIS master server. Select an NIS domain name to use for your group of three systems. Set the domain name, and record its name in the `/etc/defaultdomain` file. Enter the `touch` command to create any files in the `/etc` directory that are required by the target `all` in the `Makefile` file. Edit the automount master map and indirect map to comment out "+" entries.
- On the system to be the NIS master server, share the `/export/home` directory by using NFS. Create three user accounts and set passwords for these users. Configure the `/etc/passwd` file and the automount indirect map to allow the users to mount their home directories from the NIS master. Use the `ypinit -m` command to initialize the NIS master. Configure the `/etc/nsswitch.conf` file for NIS, and start the NIS server daemons.
- Create and configure an NIS slave server. Set the NIS domain name to be the same as in the NIS master. Use the `ypinit -c` command to configure the system as an NIS client. Configure the `/etc/nsswitch.conf` file for NIS and start the NIS client daemons. Use the `ypinit -s` command to configure the system as an NIS slave server. Stop and restart the NIS daemons. Verify the list of servers found in the `ypservers` map.
- Create and configure an NIS client system. Set the NIS domain name to be the same as in the NIS master. Use the `ypinit -c` command to configure the system as an NIS client. Configure the `/etc/nsswitch.conf` file for NIS, and start the NIS client daemons. Test the configuration with the `ypwhich` command.
- Test the dynamic rebind feature by stopping the NIS services on the NIS master server. Monitor the NIS client with the `ypwhich` command, and observe when the client binds to the slave server. Start the NIS services on the NIS master.
- Make the appropriate changes in the `/var/yp/Makefile` file to support a new automount direct map called `auto_direct`. Create the direct map in the `/etc` file. Configure the direct map and NFS shares to allow all three systems to automatically mount the man pages from the NIS master server.
- Test if the new users can log in on all three systems. Verify that their home directories automatically mount. Verify that the man pages are available through the `automount` service on all three systems.

Tasks

This section describes how to create and test the NIS master server, slave server, and client. Perform the following tasks.

Task 1 – Setting Up the NIS Master

Complete the following steps:

1. Change the directory to `/var/yp`, and make a backup copy of the `Makefile` file.
2. In the `/var/yp/Makefile`, remove the `aliases` entry from the target `all`.
3. Verify that the `/etc/hosts` file contains entries for the systems that will become the NIS slave server and the NIS client.
4. Select a name to use as your NIS domain name. Set it by using the `domainname` command.
5. Populate the `defaultdomain` file with your domain name.
6. Use the `touch` command to create the `ethers`, `bootparams`, and `netgroup` files.
7. Create the `/etc/timezone` file, and include an appropriate entry for your time zone and NIS domain.
8. Edit the `/etc/auto_master` file, and comment out the `+auto_master` entry.
9. Edit the `/etc/auto_home` file, and comment out the `+auto_home` entry. Add a new entry that supports automatically mounting all user home directories located in the `/export/home` directory on the NIS master server.

10. Configure the NIS master to share the `/export/home` directory:
    a. Create an entry in the `/etc/dfs/dfstab` file to share the users' home directories.
    b. Check if the `mountd` and `nfsd` NFS server daemons are running.
    c. If the NFS server daemons are not running, start them. The directory listed in `/etc/dfs/dfstab` will be automatically shared.
    d. If the NFS server daemons are already running, perform the command to share the new directory listed in the `/etc/dfs/dfstab` file.
11. Create one user account for each member of your lab team.

Note – Create their respective home directories in `/export/home`; for example: `/export/home/user1` for `user1`, `/export/home/user2` for `user2`, and so on.

12. Create a password for each new user account.
13. To enable using the `automount` service to mount these users' home directories, you must modify the users' entries in the `/etc/passwd` file on the NIS master server.

    Edit the `/etc/passwd` file, and change the home directory for each user from `/export/home/`*`username`* to `/home/`*`username`*.
14. Copy the `/etc/nsswitch.nis` template to the `/etc/nsswitch.conf` file.
15. Set up this system as an NIS master server:
    a. Use the `ypinit -m` command to start the setup process.

       The `ypinit` command lists the current system as an NIS server, and then prompts you for the next host to add as an NIS slave server.
    b. Enter the name of the system that you want to use as an NIS slave server. Press Control-D when the list is complete.

c. Specify that you do not want the `ypinit` command to quit on nonfatal errors.

   The `ypinit` command then proceeds to build the required maps.

Note – If the initialization process is successful, the `ypinit` command displays a message indicating that the current system was set up as a master server without any errors. This message is displayed even if nonfatal errors occur in the procedure.

d. If the initialization process fails, correct the problems indicated by the error messages and repeat Steps a, b, and c.

16. Start the NIS daemons.
17. Verify that this system is the NIS master by using the `ypwhich` command.

Task 2 – Setting Up the NIS Slave Server

Complete the following steps:

1. Verify that the `/etc/hosts` file contains entries for the NIS master server and that the system that will become the NIS client.
2. Set the NIS domain for this system by using the `domainname` command.
3. Populate the `defaultdomain` file with your domain name.
4. Use the `ypinit` command as follows to set up this system as an NIS client:
   a. Use the `ypinit -c` command to start the setup process.
   b. When prompted for a list of NIS servers, enter the name of the NIS master server followed by the name of the local host (which subsequently becomes a slave server). Press Control–D to terminate the list.
5. Copy the `/etc/nsswitch.nis` template to the `/etc/nsswitch.conf` file.
6. Start the NIS daemons.
7. Verify that this system is using NIS and is bound to the NIS master by using the `ypwhich` command.
8. Initialize the system as an NIS slave. Indicate that you do not want the `ypinit` command to quit on nonfatal errors.

The `ypinit` command then proceeds to retrieve the required maps from the master server.

If the initialization process is successful, the `ypinit` command displays a message that indicates that the NIS database was set up without any errors.

Note – If you did not add the name of the NIS slave server when you initially configured the NIS master, this process might fail. To correct the problem, enter the `ypinit -m` command once more on the NIS master, and add the slave server's host name. In the process of updating the NIS master, the script prompts you for confirmation when it is about to destroy the existing domain database. Confirm by typing `y`. Then, initialize the slave server again.

9. Stop and restart the NIS daemons on the slave server.
10. On the newly configured NIS slave server, test the NIS functionality by entering the following commands:

```
# ypwhich -m
# ypcat hosts
```

Note – The output of the `ypwhich` command should include the name of each map it provides to the NIS domain and include the name of the master server that controls the maps.

11. List the `ypservers` map known to the local domain. The output should include the names of the master and slave servers.

Task 3 – Setting Up the NIS Client

Complete the following steps:

1. Verify that the `/etc/hosts` file contains entries for the NIS master and slave servers.
2. Set the NIS domain for this system using the `domainname` command.
3. Populate the `defaultdomain` file with your domain name.
4. Set up this system as an NIS client:
   a. Use the `ypinit -c` command to start the setup process.
   b. Enter the name of the NIS master server and the NIS slave server (in order of preference), and press Control-D to terminate the list.

5. Copy the `/etc/nsswitch.nis` template to the `/etc/nsswitch.conf` file.
6. Start the NIS daemons.
7. Verify that this system is using NIS by using the `ypwhich` command.

Task 4 – Testing Dynamic Rebind

Complete the following steps:

1. Confirm that the NIS client is bound to the NIS master server bu using the `ypwhich` command.

Note – The output should list the name of the NIS master server.

2. Test the client's ability to bind to the NIS slave server when the master becomes unavailable:

Note – This process *only* works if you entered the names of *both* the NIS master and the NIS slave servers when you set up the client system by using the `ypinit -c` command. The NIS client searches only for servers listed in the `/var/yp/binding/`*`domainname`*`/ypservers` file, which the `ypinit -c` command creates.

   a. On the NIS master server, stop the NIS services.
   b. On the NIS client, determine to which NIS server to which it is bound. It can take a minute or two for the client to bind to the NIS slave.

      Allow a few moments to pass, and then repeat the `ypwhich` command. Do this until you see that the NIS client has bound to the slave server.

3. On the NIS master, start the NIS services.

Task 5 – Adding a Custom Map to the NIS Master Database

If entries for an `auto_direct` map do not exist in the `Makefile` file that you are using, complete the following steps to add them:

1. On the NIS master server, edit the `/var/yp/Makefile` file, and make the following changes:

   a. Add `auto.direct` to the list of maps associated with the target `all`. These entries exist in the *second* section of the `/var/yp/Makefile` file:

```
all: passwd group hosts ipnodes ethers networks rpc services protocols \
        netgroup bootparams aliases publickey netid netmasks c2secure \
        timezone auto.master auto.home \
        auth.attr exec.attr prof.attr user.attr audit.user auto.direct
```

   b. Add entries for a the new map in the fourth section of the `/var/yp/Makefile` file. Place a corresponding entry for `auto.direct` and `auto_direct` below the entries for `auto.home` and `auto_home`; for example:

```
auto.master: auto.master.time
auto.home: auto.home.time
auto.direct: auto.direct.time
$(DIR)/auto_master:
$(DIR)/auto_home:
$(DIR)/auto_direct:
```

   c. In the *third* section of the `Makefile` file, add the code required to build the `auto_direct` map. Duplicate the lines associated with `auto.home`, and substitute `auto.direct` or `auto_direct` for each instance of `auto.home` or `auto_home` in that code. The result should look like this:

```
auto.direct.time:  $(DIR)/auto_direct
        -@if [ -f $(DIR)/auto_direct ]; then \
                sed -e "/^#/d" -e s/#.*$$// $(DIR)/auto_direct \
                | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/auto.direct; \
                touch auto.direct.time; \
                echo "updated auto.direct"; \
                if [ ! $(NOPUSH) ]; then \
                        $(YPPUSH) auto.direct; \
                        echo "pushed auto.direct"; \
                else \
                : ; \
                fi \
        else \
                echo "couldn't find $(DIR)/auto_direct"; \
        fi
```

d. Save the modified `Makefile` file, and exit the editor.

2. On the master server, edit the `/etc/auto_master` file to include an entry for the new direct map. Add the following line:

```
/-              auto_direct     -nosuid
```

3. On the master server, create a file called `/etc/auto_direct`, and insert the following line in it. Substitute the name of the master server for *`master_server`*.

```
/usr/share/man     -ro     master_server:/usr/share/man2
```

4. On all three hosts, rename the existing `/usr/share/man` directory to `/usr/share/man2`.
5. Create a new directory called `/usr/share/man`.
6. On the master server, add an entry to the `/etc/dfs/dfstab` file to share the `/usr/share/man2` directory.
7. Share the directory.
8. Start the NIS daemons on the servers.

Note – If the daemons are already running, perform the `/usr/lib/netsvc/yp/ypstop` command to stop them.

9. On the master server, change the directory to `/var/yp`.
10. Update the NIS maps by running the `make` utility.

    The `make` command hangs when it tries to push the new `auto.direct` map to the slave server. Press Control-C to stop the `make` command when this happens.

11. On the NIS slave server, use the `ypxfr` command to transfer the `auto.direct` map for the first time.
12. On the NIS master server, update the NIS maps again by running the `make` command. This time the `make` command should complete successfully.
13. On all three hosts, use the `init 6` command to reboot.
14. Verify that you can use the user accounts you created earlier to log in to the NIS slave server and in to the NIS client.

15. On the NIS slave and NIS client, verify that your home directory automatically mounts from the NIS master server.
16. On all systems, attempt to access the `/usr/share/man` directory by using the `man` command.

    If the content of the man page for the `ls` command is displayed, your configuration of the direct map in NIS is correct.

Exercise: Configuring NIS (Level 3)

Perform the following tasks:

- Configure the following
  - An NIS master server
  - An NIS slave server
  - An NIS client
- Test the dynamic rebind feature
- Add a custom map to NIS

Preparation

Choose two partners for this lab, and determine which systems to configure as the NIS master server, the NIS slave server, and the NIS client.

NIS_master: _____

NIS_slave: _____

NIS_client: _____

domainname: _____

On all systems, verify that entries for all three hosts exist in the `/etc/hosts` file. Refer to your lecture notes as necessary to perform the steps listed.

Task Summary

Perform the following tasks:

- Create and configure an NIS master server. Select an NIS domain name to use for your group of three systems. Set the domain name, and record its name in the `/etc/defaultdomain` file. Enter the `touch` command to create any files in the `/etc` directory that are required by the target `all` in the `Makefile` file. Edit the automount master map and indirect map to comment out "+" entries.
- On the system to be the NIS master server, share the `/export/home` directory by using NFS. Create three user accounts and set passwords for these users. Configure the `/etc/passwd` file and the automount indirect map to allow the users to mount their home directories from the NIS master. Use the `ypinit -m` command to initialize the NIS master. Configure the `/etc/nsswitch.conf` file for NIS, and start the NIS server daemons.
- Create and configure an NIS slave server. Set the NIS domain name to be the same as in the NIS master. Use the `ypinit -c` command to configure the system as an NIS client. Configure the `/etc/nsswitch.conf` file for NIS and start the NIS client daemons. Use the `ypinit -s` command to configure the system as an NIS slave server. Stop and restart the NIS daemons. Verify the list of servers found in the `ypservers` map.
- Create and configure an NIS client system. Set the NIS domain name to be the same as in the NIS master. Use the `ypinit -c` command to configure the system as an NIS client. Configure the `/etc/nsswitch.conf` file for NIS, and start the NIS client daemons. Test the configuration with the `ypwhich` command.
- Test the dynamic rebind feature by stopping the NIS services on the NIS master server. Monitor the NIS client with the `ypwhich` command, and observe when the client binds to the slave server. Start the NIS services on the NIS master.
- Make the appropriate changes in the `/var/yp/Makefile` file to support a new automount direct map called `auto_direct`. Create the direct map in the `/etc` file. Configure the direct map and NFS shares to allow all three systems to automatically mount the man pages from the NIS master server.
- Test if the new users can log in on all three systems. Verify that their home directories automatically mount. Verify that the man pages are available through the `automount` service on all three systems.

Tasks and Solutions

This section describes how to create and test the NIS master server, slave server, and client.

Task 1 – Setting Up the NIS Master

Complete the following steps:

1. Change the directory to `/var/yp`, and make a backup copy of the `Makefile` file.

```
# cd /var/yp
# cp Makefile Makefile.orig
```

2. In the `/var/yp/Makefile`, remove the `aliases` entry from the target `all`.
3. Verify that the `/etc/hosts` file contains entries for the systems that will become the NIS slave server and the NIS client.
4. Select a name to use as your NIS domain name. Set it by using the `domainname` command.

```
# domainname yourdomain
```

Note – Replace *yourdomain* with your chosen domain name.

5. Populate the `defaultdomain` file with your domain name.

```
# cd /etc
# domainname > defaultdomain
```

6. Use the `touch` command to create the `ethers`, `bootparams`, and `netgroup` files.

```
# touch ethers bootparams netgroup
```

7. Create the `/etc/timezone` file, and include an appropriate entry for your time zone and NIS domain.

   For example, the following entry would set the time zone for systems located within an NIS domain called *yourdomain*.

```
your_timezone      yourdomain
```

Note – Replace *your_timezone* time zone with your local time zone and *yourdomain* with your own domain name.

8. Edit the `/etc/auto_master` file, and comment out the `+auto_master` entry.

```
# Master map for automounter#
# +auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
/xfn            -xfn
```

9. Edit the `/etc/auto_home` file, and comment out the `+auto_home` entry. Add a new entry that supports automatically mounting all user home directories located in the `/export/home` directory on the NIS master server.

```
# Home directory map for automounter
#
# +auto_home
*    master_server:/export/home/&
```

10. Configure the NIS master to share the `/export/home` directory:

    a. Create an entry in the `/etc/dfs/dfstab` file to share the users' home directories.

```
share -d "home dirs" /export/home
```

    b. Check if the `mountd` and `nfsd` NFS server daemons are running.

```
# pgrep -xl mountd
# pgrep -xl nfsd
```

    c. If the NFS server daemons are not running, start them. The directory listed in `/etc/dfs/dfstab` will be automatically shared.

```
# /etc/init.d/nfs.server start
```

    d. If the NFS server daemons are already running, perform the command to share the new directory listed in the `/etc/dfs/dfstab` file.

```
# shareall
```

11. Create one user account for each member of your lab team.

Note – Create their respective home directories in `/export/home`; for example: `/export/home/user1` for `user1`, `/export/home/user2` for `user2`, and so on. If you use the Solaris Management Console application to create the user accounts, the account is configured to use the `automount` command, and the `/export/home/user1` directory is translated to the `/home/user1` directory.

12. Create a password for each new user account.
13. To enable using the `automount` service to mount these users' home directories, you must modify the users' entries in the `/etc/passwd` file on the NIS master server.

    Edit the `/etc/passwd` file, and change the home directory for each user from `/export/home/`*`username`* to `/home/`*`username`*.
14. Copy the `/net/nsswitch.nis` template to the `/etc/nsswitch.conf` file.

```
# cp nsswitch.nis nsswitch.conf
```

15. Set up this system as an NIS master server:
    a. Use the `ypinit -m` command to start the setup process.

```
# ypinit -m
```

    The `ypinit` command lists the current system as an NIS server, and then prompts you for the next host to add as an NIS slave server.

    b. Enter the name of the system that you want to use as an NIS slave server. Press Control-D when the list is complete.

```
next host to add: master_server
next host to add: slave_server
next host to add: <Control-D>
(list of servers)
is this list correct? [y/n: y] y
```

c. Specify that you do not want the `ypinit` command to quit on nonfatal errors.

```
...quit on nonfatal errors? [y/n: n] n
```

The `ypinit` command then proceeds to build the required maps.

Note – If the initialization process is successful, the `ypinit` command displays a message indicating that the current system was set up as a master server without any errors. This message is displayed even if nonfatal errors occur in the procedure.

d. If the initialization process fails, correct the problems indicated by the error messages and repeat Step a, Step , and Step c.

16. Start the NIS daemons.

```
# /usr/lib/netsvc/yp/ypstart
```

17. Verify that this system is the NIS master by using the `ypwhich` command.

```
# ypwhich -m
```

Task 2 – Setting Up the NIS Slave Server

Complete the following steps:

1. Verify that the `/etc/hosts` file contains entries for the NIS master server and that the system that will become the NIS client.
2. Set the NIS domain for this system by using the `domainname` command.

```
# domainname yourdomain
```

Note – Replace *`yourdomain`* with the NIS domain name you used to set up the NIS master server.

3. Populate the `defaultdomain` file with your domain name.

```
# cd /etc
# domainname > defaultdomain
```

4. Use the `ypinit` command as follows to set up this system as an NIS client:

   a. Use the `ypinit -c` command to start the setup process.

```
# ypinit -c
```

   b. When prompted for a list of NIS servers, enter the name of the NIS master server followed by the name of the local host (which subsequently becomes a slave server). Press Control–D to terminate the list.

```
next host to add: master_server
next host to add: slave_server
next host to add: <Control-D>
(list of servers)
is this list correct? [y/n: y] y
```

5. Copy the `/etc/nsswitch.nis` template to the `/etc/nsswitch.conf` file.

```
# cp nsswitch.nis nsswitch.conf
```

6. Start the NIS daemons.

```
# /usr/lib/netsvc/yp/ypstart
```

7. Verify that this system is using NIS and is bound to the NIS master by using the `ypwhich` command.

```
# ypwhich
```

8. Initialize the system as an NIS slave.

```
# ypinit -s master_server
```

   Indicate that you do not want the `ypinit` command to quit on nonfatal errors.

```
...quit on nonfatal errors? [y/n: n] n
```

   The `ypinit` command then proceeds to retrieve the required maps from the master server.

   If the initialization process is successful, the `ypinit` command displays a message that indicates that the NIS database was set up without any errors.

Note – If you did not add the name of the NIS slave server when you initially configured the NIS master, this process might fail. To correct the problem, enter the `ypinit -m` command once more on the NIS master, and add the slave server's host name. In the process of updating the NIS master, the script prompts you for confirmation when it is about to destroy the existing domain database. Confirm by typing `y`. Then, initialize the slave server again.

9. Stop and restart the NIS daemons on the slave server.

```
# /usr/lib/netsvc/yp/ypstop
# /usr/lib/netsvc/yp/ypstart
```

10. On the newly configured NIS slave server, test the NIS functionality by entering the following commands:

```
# ypwhich -m
# ypcat hosts
```

Note – The output of the `ypwhich` command should include the name of each map it provides to the NIS domain and include the name of the master server that controls the maps.

11. List the `ypservers` map known to the local domain. The output should include the names of the master and slave servers.

```
# ypcat -k ypservers
slave_server
master_server
```

Task 3 – Setting Up the NIS Client

Complete the following steps:

1. Verify that the `/etc/hosts` file contains entries for the NIS master and slave servers.
2. Set the NIS domain for this system using the `domainname` command.

```
# domainname yourdomain
```

Note – Replace *yourdomain* with the NIS domain name you used to set up the NIS master server.

3. Populate the `defaultdomain` file with your domain name.

```
# cd /etc
# domainname > defaultdomain
```

4. Set up this system as an NIS client:
   a. Use the `ypinit -c` command to start the setup process.

```
# ypinit -c
```

   b. Enter the name of the NIS master server and the NIS slave server (in order of preference), and press Control-D to terminate the list.

```
next host to add: master_server
next host to add: slave_server
next host to add: <Control-D>
(list of servers)
is this list correct? [y/n: y] y
```

5. Copy the `/etc/nsswitch.nis` template to the `/etc/nsswitch.conf` file.

```
# cd /etc
# cp nsswitch.nis nsswitch.conf
```

6. Start the NIS daemons.

```
# /usr/lib/netsvc/yp/ypstart
```

7. Verify that this system is using NIS by using the `ypwhich` command.

```
# ypwhich -m
```

Task 4 – Testing Dynamic Rebind

Complete the following steps:

1. Confirm that the NIS client is bound to the NIS master server by using the `ypwhich` command.

```
# ypwhich
master_server
```

Note – The output should list the name of the NIS master server.

2. Test the client's ability to bind to the NIS slave server when the master becomes unavailable:

Note – This process *only* works if you entered the names of *both* the NIS master and the NIS slave servers when you set up the client system by using the `ypinit -c` command. The NIS client searches only for servers listed in the `/var/yp/binding/`*`domainname`*`/ypservers` file, which the `ypinit -c` command creates.

   a. On the NIS master server, stop the NIS services.

```
# /usr/lib/netsvc/yp/ypstop
```

   b. On the NIS client, determine to which NIS server to which it is bound. It can take a minute or two for the client to bind to the NIS slave.

      Allow a few moments to pass, and then repeat the `ypwhich` command. Do this until you see that the NIS client has bound to the slave server.

```
# ypwhich
```

3. On the NIS master, start the NIS services.

```
# /usr/lib/netsvc/yp/ypstart
```

Task 5 – Adding a Custom Map to the NIS Master Database

If entries for an `auto_direct` map do not exist in the `Makefile` file that you are using, complete the following steps to add them:

1. On the NIS master server, edit the `/var/yp/Makefile` file, and make the following changes:
   a. Add `auto.direct` to the list of maps associated with the target `all`. These entries exist in the *second* section of the `/var/yp/Makefile` file:

```
all: passwd group hosts ipnodes ethers networks rpc services protocols \
        netgroup bootparams aliases publickey netid netmasks c2secure \
        timezone auto.master auto.home \
        auth.attr exec.attr prof.attr user.attr audit.user auto.direct
```

   b. Add entries for a the new map in the fourth section of the `/var/yp/Makefile` file. Place a corresponding entry for `auto.direct` and `auto_direct` below the entries for `auto.home` and `auto_home`; for example:

```
auto.master: auto.master.time
auto.home: auto.home.time
auto.direct: auto.direct.time

$(DIR)/auto_master:
$(DIR)/auto_home:
$(DIR)/auto_direct:
```

c. In the *third* section of the `Makefile` file, add the code required to build the `auto_direct` map. Duplicate the lines associated with `auto.home`, and substitute `auto.direct` or `auto_direct` for each instance of `auto.home` or `auto_home` in that code. The result should look like this:

```
auto.direct.time:  $(DIR)/auto_direct
        -@if [ -f $(DIR)/auto_direct ]; then \
                sed -e "/^#/d" -e s/#.*$$// $(DIR)/auto_direct \
                | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/auto.direct; \
                touch auto.direct.time; \
                echo "updated auto.direct"; \
                if [ ! $(NOPUSH) ]; then \
                        $(YPPUSH) auto.direct; \
                        echo "pushed auto.direct"; \
                else \
                : ; \
                fi \
        else \
                echo "couldn't find $(DIR)/auto_direct"; \
        fi
```

d. Save the modified `Makefile` file, and exit the editor.

2. On the master server, edit the `/etc/auto_master` file to include an entry for the new direct map. Add the following line:

```
/-              auto_direct     -nosuid
```

3. On the master server, create a file called `/etc/auto_direct`, and insert the following line in it. Substitute the name of the master server for *`master_server`*.

```
/usr/share/man      -ro     master_server:/usr/share/man2
```

4. On all three hosts, rename the existing `/usr/share/man` directory to `/usr/share/man2`.

```
# mv /usr/share/man /usr/share/man2
```

5. On all three hosts, create a new directory called `/usr/share/man`.

```
# mkdir /usr/share/man
```

6. On the master server, add an entry to the `/etc/dfs/dfstab` file to share the `/usr/share/man2` directory.

```
# vi /etc/dfs/dfstab
share -o ro /usr/share/man2
```

7. Share the directory.

```
# shareall
```

8. Start the NIS daemons on the servers.

Note – If the daemons are already running, perform the `/usr/lib/netsvc/yp/ypstop` command to stop them.

```
# /usr/lib/netsvc/yp/ypstart
```

9. On the master server, change the directory to `/var/yp`.

```
# cd /var/yp
```

10. Update the NIS maps by running the `make` utility.

```
# /usr/ccs/bin/make
...
<Control-C>
#
```

The `make` command hangs when it tries to push the new `auto.direct` map to the slave server. Press Control-C to stop the `make` command when this happens.

11. On the NIS slave server, use the `ypxfr` command to transfer the `auto.direct` map for the first time.

```
# /usr/lib/netsvc/yp/ypxfr auto.direct
```

12. On the NIS master server, update the NIS maps again by running the `make` command. This time the `make` command should complete successfully.

```
# cd /var/yp
# /usr/ccs/bin/make
```

13. On all three hosts, use the `init 6` command to reboot.

```
# init 6
```

14. Verify that you can use the user accounts you created earlier to log in to the NIS slave server and in to the NIS client.

15. On the NIS slave and NIS client, verify that your home directory automatically mounts from the NIS master server.

```
$ pwd
```

16. On all systems, attempt to access the `/usr/share/man` directory by using the `man` command.

```
$ man ls
```

If the content of the man page for the `ls` command is displayed, your configuration of the direct map in NIS is correct.

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Module 17

Configuring the Custom JumpStart™ Procedure

Objectives

The JumpStart™ procedure provides a mechanism for automatically installing the Solaris™ 9 Operating Environment (Solaris 9 OE) on multiple systems simultaneously. The JumpStart procedure can be custom configured to fit the profile of the client systems being installed.

Upon completion of this module, you should be able to:

- Describe the JumpStart procedure
- Implement a basic JumpStart server
- Set up JumpStart software configuration alternatives
- Troubleshoot the JumpStart procedure

The following course map shows how this module fits into the current instructional goal.

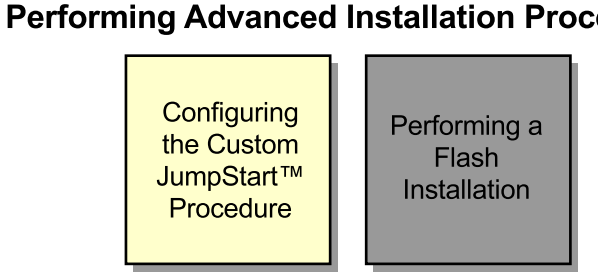


Figure 17-1 Course Map

Introducing the JumpStart Procedure

The JumpStart procedure is an automatic installation process available in the Solaris 9 OE. The JumpStart procedure enables you to install the Solaris OE automatically and configure it differently, depending on the characteristics of client systems. The JumpStart procedure uses these identifying characteristics to select the correct configuration for each client system.

Purpose of the JumpStart Procedure

System administrators who need to install multiple systems with similar configurations can use the JumpStart procedure to automate the installation process. The JumpStart procedure eliminates the need for operator intervention during the installation process.

The advantages of using the JumpStart procedure include the following:

- It lets system administrators avoid the lengthy question-and-answer session that is part of the interactive installation process.
- It lets system administrators install different types of systems simultaneously.
- It allows automatic installation of the Solaris 9 OE and unbundled software.
- It simplifies administration tasks when widely used applications must be updated frequently.

The JumpStart procedure provides considerable time savings when multiple or ongoing installations are required for networked computing environments.

Four main services support the software installation process using the JumpStart procedure:

- Boot services
- Identification services
- Configuration services
- Installation services

Configuring the JumpStart procedure program requires setting up these services on one or more networked servers. You can configure a single server to provide all four services for the JumpStart procedure, or you can configure the services separately on different servers.

Figure 17-2 shows a typical JumpStart block diagram.

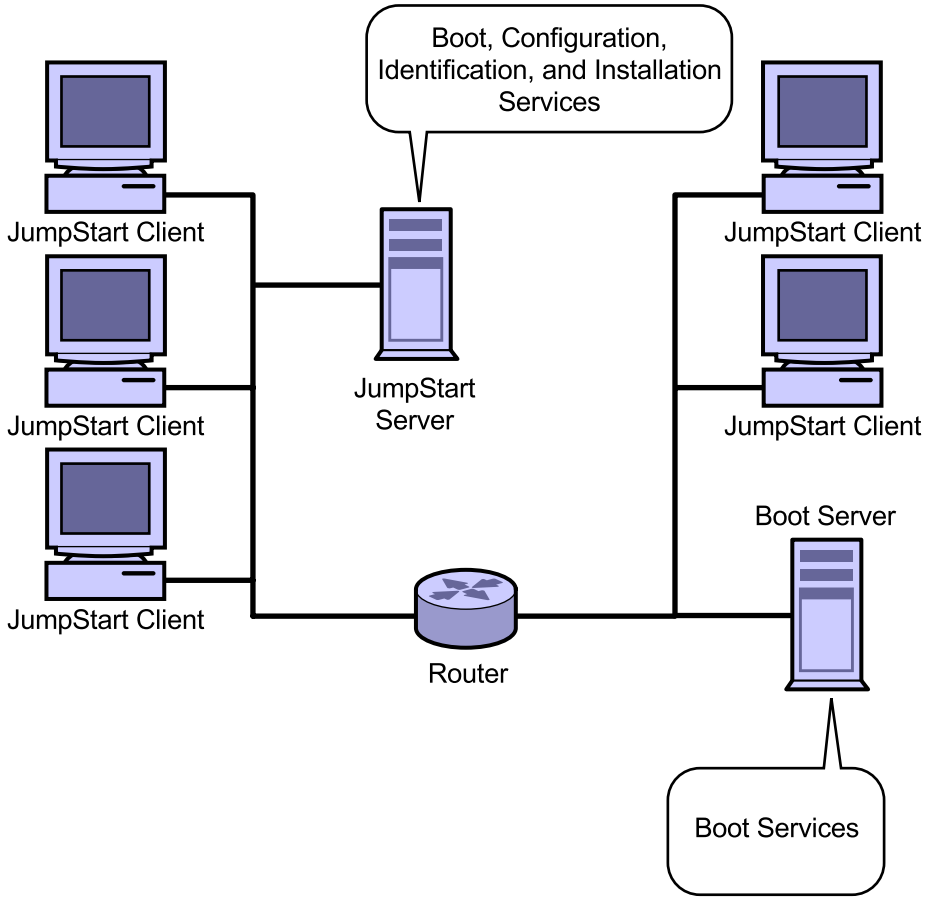


Figure 17-2 JumpStart Server Component Services

Boot Services

To boot the JumpStart client using the network, clients require support from a server that can respond to their Reverse Address Resolution Protocol (RARP), Trivial File Transfer Protocol (TFTP), and `bootparams` file requests. A system that provides these services is called a boot server. You can configure a boot server to provide any of the other required JumpStart services, or to only provide boot services.

If other servers provide identification, configuration, and installation services, the boot server identifies those servers for the JumpStart client. To support client RARP requests, the boot server must reside on the same subnet as the client, but the servers that provide these other services can reside on other network segments.

For boot operations to proceed, the following files must be properly configured on the boot server:

- The `/etc/ethers` file
- The `/etc/inet/hosts` file
- The `/tftpboot` file
- The `/etc/inet/inetd.conf` file
- The `/etc/bootparams` file
- The `/etc/dfs/dfstab` file

The `/etc/ethers` and `/etc/inet/hosts` files configure the boot server to support RARP requests from JumpStart clients.

For each JumpStart client that the boot server supports, the `/tftpboot` directory must contain a symbolic link that points to a network bootstrap program. The `/etc/inet/inetd.conf` file must contain an entry that allows the `in.tftpd` daemon to run on demand.

The boot server provides access to a boot image (a root (/) file system) that all JumpStart clients on the subnet use during the network boot process. The `/etc/bootparams` file lists the location of this root (/) file system and the locations of other directories that the JumpStart client requires. The `/etc/dfs/dfstab` file configures JumpStart servers to share the directories that they provide.

You can configure boot services using the `add_install_client` script. The `add_install_client` script allows you to specify all of the information required in the files that support boot services. This script also creates the required files in the `/tftpboot` directory and appropriately modifies the `/etc/inet/inetd.conf` file.

Identification Services

JumpStart clients require support from a server to automatically get the answers to system identification questions that the client systems issue. The identification service is often provided by a boot server, but the service can be provided by any network server configured to provide identification.

JumpStart clients can obtain identification information from different sources, including the `/etc/inet/hosts` file on the boot server, the `sysidcfg` file, and a name service such as Network Information Service (NIS) or Network Information Service Plus (NIS+). You can use a combination of these sources to answer the client's identification requests. Identification information provided in a `sysidcfg` file overrides information provided by other sources.

Configuring a server to provide identification services is, for the most part, a manual process. You must manually edit the `sysidcfg` file, and share the directory where it resides. During the installation process, JumpStart clients use the Network File System (NFS) service to mount the directory that contains the `sysidcfg` file.

If you use a name service, configuring identification services involves editing the source files that the name service uses and running commands to update the name service.

If the JumpStart client cannot obtain a response from a server for any identification item, the client interrupts the automatic identification process and asks for the information.

Listing Identification Items and Their Sources

Table 17-1 lists the identification items that JumpStart clients using SPARC® technology require, and also lists the sources in the Solaris 9 OE that can provide the information. In earlier releases of the Solaris OE, the list of items and usable sources sometimes differed. In systems with Intel Architecture, you might need to specify additional items in the `sysidcfg` file.

Table 17-1 JumpStart Client Identification Items

Identification Item	**Configurable With the `sysidcfg` File?**	**Configurable With a Name Service?**
Name service	Yes	Yes
Domain name	Yes	No
Name server	Yes	No
Network interface	Yes	No
Host name	Yes	Yes
IP address	Yes	Yes
Netmask	Yes	Yes, for the primary interface
Dynamic Host Configuration Protocol (DHCP)	Yes	No
Internet Protocol Version 6 (IPv6)	Yes	No
Default router	Yes	No
Root password	Yes	No
Security policy	Yes	No
Locale	Yes	Yes if NIS or NIS+, No if DNS or Lightweight Directory Access Protocol (LDAP)
Terminal Type	Yes	No
Time zone	Yes	Yes
Date and time	Yes	Yes
Power management (auto shutdown)	No	No

For more information, refer to the *Advanced Installation Guide* online at `http://docs.sun.com`.

Configuration Services

JumpStart clients require support from a server to automatically obtain answers for system configuration questions that they issue. A system that provides this service is called a configuration server.

A configuration server provides information that specifies how the Solaris OE installation will proceed on the JumpStart client. Configuration information can include:

- Installation type
- System type
- Disk partitioning and file system specifications
- Configuration cluster selection
- Software package additions or deletions

On the configuration server, files known as profile files store the configuration information. A file called `rules` on the configuration server allows JumpStart clients to select an appropriate profile file.

Associating a Configuration With a Client

A configuration server shares a directory (typically the `/export/config` directory) that minimally contains the files shown in Table 17-2.

Table 17-2 Files in the `/export/config` Directory

File	Description
The `rules` file	The `rules` file associates classes of clients with specific installation profiles. Classes in the `rules` file are identified using predefined keywords that include: • `hostname` • `arch` • `domainname` • `memsize` • `model` Clients select a profile by matching their own characteristics with an entry in the `rules` file.
The profile (class) files	The profile files specify how the installation is to proceed and what software is to be installed. A separate profile file can exist for each class of JumpStart client on your network.
The `check` script	Run the `check` script after creating the `rules` and profile files. The `check` script verifies the syntax in the `rules` and profile files. If there are no syntax errors, the `check` script creates the `rules.ok` file.
The `rules.ok` file	The `check` script creates the `rules.ok` file from the `rules` file. The JumpStart installation procedure reads the `rules.ok` file during the automatic installation process (the `rules` file is not read).
Optional begin and finish scripts	The JumpStart client uses begin and finish scripts to perform preinstallation and postinstallation tasks. You can use these scripts to further customize the installation process, such as configuring power management on the JumpStart client. The begin and finish scripts are located in the configuration directory hierarchy shared by the configuration server.

Installation Services

JumpStart clients require support from a server to find an image of the Solaris OE to install. A system that provides this service is called an install server. An install server shares a Solaris OE image from a CD-ROM, from a DVD, or from a local disk. JumpStart clients use the NFS service to mount the installation image during the installation process.

Sources of the OE Image

An install server provides the Solaris OE image by sharing one of the following:

- The Solaris 9 Software 1 of 2 CD-ROM
- The Solaris 9 Software DVD
- A spooled image of the Solaris 9 OE obtained from either the CD-ROM or DVD media
- A Flash installation image

CD-ROM and DVD

An install server can provide installation services by sharing either the Solaris 9 Software 1 of 2 CD-ROM or the Solaris 9 Software DVD.

The Solaris 9 Software 1 of 2 CD-ROM and the Solaris 9 Software DVD both contain a boot image and an installation image. Sharing either of these supports both boot services and installation services.

The installation image found on the Solaris 9 Software 1 of 2 CD-ROM only supports installing the Core and End User configuration clusters. The Solaris 9 Software 2 of 2 CD-ROM contains the remainder of the installation image, but there is no support for changing CD-ROMs in the middle of a JumpStart installation procedure.

Beginning with the Solaris 8 2/02 release, the Solaris™ Media Kit has been available on either CD-ROM or DVD media.

The Spooled Image

An install server can provide installation services by sharing a spooled image on local disk. When you spool the Solaris OE image from CD-ROM or DVD, the result is a directory that contains the boot image and the installation image:

The boot image	JumpStart clients can boot from the root (/) file system contained in the boot image. For example, if you spool the Solaris 9 OE into a directory called `/export/install`, the boot image would be located in the `/export/install/Solaris_9/Tools/Boot` directory.
The installation image	JumpStart clients install the Solaris OE from the installation image. For example, if you spool the Solaris 9 OE into a directory called `/export/install`, the installation image would be located in the `/export/install/Solaris_9/Product` directory.

The `setup_install_server` script enables you to spool the boot and installation images from the 1 of 2 CD-ROM or from the DVD.

The `add_to_install_server` script enables you to spool additional installation image data from the 2 of 2 CD-ROM.

The `setup_install_server` script with the `-b` option enables you to spool only the boot image from the 1 of 2 CD-ROM or from the DVD. The script supports creating a boot image on a boot server. The boot server directs the JumpStart client to a separate install server for the installation image.

A Flash Install Image

Flash installation is significantly faster than the current JumpStart installation or a Web Start™ network installation methods. Flash allows detailed customization of the Solaris OE, hardware configuration, and third-party software packages prior to creation of the clones. In addition, Flash installation can provide enterprise-level disaster recovery when necessary.

Implementing a Basic JumpStart Server

A JumpStart server configuration includes:

- A single server that provides boot, identification, configuration, and installation services
- Boot and installation services provided by the Solaris 9 OE boot and installation images spooled to the local disk of the server
- Identification services provided by files on the server and a `sysidcfg` file, with no name service in place
- Configuration services provided by a `rules` file that contains an entry for a single JumpStart client, and a profile file that installs the entire Solaris 9 OE distribution into a single slice on the JumpStart client

The following tasks are required to configure a single JumpStart server to provide basic software installation services using the JumpStart procedures:

1. Spool the operating system image.
2. Edit the `sysidcfg` file.
3. Edit the `rules` and profile files.
4. Run the `check` script.
5. Run the `add_install_client` script.
6. Boot the client.

Spooling the Operating System Image

Spooling the Solaris OE boot and installation image to disk is the most common method of supplying boot and installation services to JumpStart clients. You can spool the boot image and installation image to different servers. The following example shows how one server provides both boot and installation services.

When you use the Solaris 9 CD-ROM source media, you must use the `setup_install_server` script to spool the Solaris 9 OE image from the 1 of 2 CD-ROM and use the `add_to_install_server` script to spool the Solaris 9 OE image from the 2 of 2 CD-ROM.

The 1 of 2 CD-ROM provides the boot image and the required portion of the installation image to install the *End User* configuration cluster. The 2 of 2 CD-ROM provides the remainder of the installation image, containing the data required to install the Developer, Entire Distribution, or the Entire Distribution with OEM Support configuration cluster.

When you use the Solaris 9 DVD source media, you are using the `setup_install_server` script to spool the Solaris 9 OE boot image and complete the installation image to disk.

When the spooling procedure is complete, the server has the data available to support boot and installation services for JumpStart clients. The spooled image also contains the `add_install_client` script that lets you establish boot and installation support for specific JumpStart clients.

To spool the Solaris 9 OE boot and installation images to a local disk, complete the following steps:

1. Create a directory with at least 800 Mbytes of space available to hold the Solaris OE image. Usually the `/export/install` directory is used.

```
# mkdir /export/install
```

2. Insert the Solaris 9 Software 1 of 2 CD-ROM in the CD-ROM drive or the Solaris 9 DVD in the DVD drive. Allow the `vold` daemon to automatically mount the media.
3. Change the directory to the location of the `setup_install_server` script.

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```

4. Run the `setup_install_server` script to copy the Solaris 9 OE boot and installation images to the local disk (this process can take about one hour).

```
# ./setup_install_server /export/install
```

5. When the `setup_install_server` script finishes, change the directory to root (`/`), and eject the CD-ROM or DVD.

```
# cd /
# eject cdrom
```

6. If you use CD-ROM media, insert the Solaris 9 Software 2 of 2 CD-ROM in the CD-ROM drive, and allow the `vold` daemon to automatically mount it.

   a. Change the directory to the location of the `add_to_install_server` script.

```
# cd /cdrom/cdrom0/Solaris_9/Tools
```

   b. Run the `add_to_install_server` script to copy the remainder of the installation image to the local disk (this process can take about 20 minutes).

```
# ./add_to_install_server /export/install
```

   c. When `add_to_install_server` finishes, change the directory to root (`/`), and eject the CD-ROM.

```
# cd /
# eject cdrom
```

Editing the sysidcfg File

To provide complete identification services in the absence of a name service, the JumpStart server must provide information in the `sysidcfg` file that answers the following questions:

- Will the client be configured to use IPv6 networking?
- What netmask will the client use?
- What is the Internet Protocol (IP) address of the default router?
- What security policy will the client implement?
- What name service will the client use?
- What time zone will the client use?
- What system locale will the client use?
- What system will provide the time-of-day information?
- What is the `root` password?

The following example shows the possible entries in a sysidcfg file that answers all of these questions. The example contains entries that can be used by any JumpStart client on the same subnet.

```
network_interface=primary {protocol_ipv6=no
                           netmask=255.255.255.0
                           default_route=192.10.10.100}
security_policy=none
name_service=none
timezone=US/Mountain
system_locale=en_US
timeserver=192.10.10.100
root_password=Hx23475vABDDM
```

For all of these items listed, you must specify values that are appropriate for your own systems, location, and network.

To configure a generic sysidcfg file on a JumpStart server, complete the following steps:

1. Create a directory to hold the sysidcfg file. Typically the /export/config directory holds the sysidcfg file.

```
# mkdir /export/config
```

2. Change the directory to /export/config, and create a file called sysidcfg using a text editor.

```
# cd /export/config
# vi sysidcfg
```

3. In the sysidcfg file, add the following lines. Substitute values that are appropriate for your systems, location, and network.

```
network_interface=primary {protocol_ipv6=no
                           netmask=netmask_value
                           default_route=router_IP}
security_policy=none
name_service=none
timezone=timezone
system_locale=locale
timeserver=timeserver_IP
root_password=Hx23475vABDDM
```

   a. For the *netmask_value*, enter the correct netmask for your network.

   b. For the *router_IP* value, enter the IP address of the system that will act as your default router.

c. For the *timezone* value, enter the correct time zone for your location. Time zones are listed in the directory structure below the `/usr/share/lib/zoneinfo` directory. For example, the `US/Mountain` time zone refers to the `/usr/share/lib/zoneinfo/US/Mountain` directory.

d. For the *locale* value, enter the correct system locale for your location. Locales are listed in the `/usr/lib/locale` directory.

e. For the *timeserver_IP* value, enter the IP address of the system that will provide the time-of-day to the JumpStart client.

Note – The `root` password string of `Hx23475vABDDM` represents the password `cangetin`.

4. Save the `sysidcfg` file, and exit your edit session.

Editing the `rules` and Profile Files

To provide configuration services, the JumpStart server must provide a `rules` file that allows the JumpStart client to select a profile file. The profile file must contain information that answers all of the configuration questions that the JumpStart client requires.

If the JumpStart client cannot obtain a response from a server for any configuration item, the client interrupts the automatic configuration process and asks for the information.

A very basic `rules` file can contain a single entry that allows a single client to select a profile file according to its host name. For example:

```
hostname client1 - profile1 -
```

This `rules` file causes a JumpStart client called `client1` to use a profile file called `profile1`. The dash (`-`) characters before and after the `profile1` file indicate that the `client1` system will not run a begin or a finish script.

The name of the profile file must match the name listed in the `rules` file, and the profile file must supply all of the configuration information that the client requires. For example, a simple profile file can contain the following information:

```
install_type    initial_install
system_type     standalone
partitioning    explicit
filesys         c0t0d0s0        free    /
filesys         c0t0d0s1        128     swap
cluster         SUNWCXall
```

This profile file declares that the JumpStart client will perform an initial installation as a standalone system, use partitioning that allocates 128 Mbytes to the swap area, and allocates the remainder of the disk space to the root (/) file system, and that the client will install the Entire Distribution with OEM support configuration cluster.

To configure a simple `rules` and profile file on a JumpStart server, complete the following steps:

1. Create a directory to hold the `rules` file if this directory does not already exist. Usually, the `/export/config` directory holds the `rules` file.

```
# mkdir /export/config
```

2. Change the directory to `/export/config`, and create a file called `rules` using a text editor.

```
# cd /export/config
# vi rules
```

3. In the `rules` file, add the following line. For *`client_name`*, substitute the name of your JumpStart client.

```
hostname client_name - profile1 -
```

4. Save the `rules` file, and exit your edit session.
5. Create a file called `profile1` by using a text editor.

```
# vi profile1
```

6. Add the following lines to the `profile1` file:

```
install_type    initial_install
system_type     standalone
partitioning    explicit
filesys         cxtxdxs0 free    /
filesys         cxtxdxs1 128     swap
cluster         SUNWCXall
```

   a. For *cxtxdxs0*, enter the correct designation for slice 0 on the JumpStart client's boot disk.

   b. For *cxtxdxs1*, enter the correct designation for slice 1 on the JumpStart client's boot disk.

7. Save the `profile1` file, and exit your edit session.

Running the check Script

Before a JumpStart client can use a configuration provided by a JumpStart server, you must run the `check` script to produce a file called `rules.ok`. The `check` script validates the syntax of the `rules` file and the profile files. If the validation completes successfully, the check script creates the `rules.ok` file.

This procedure assumes that the `rules` and profile file that you intend to use exist in the `/export/config` directory, and that the Solaris 9 OE has been spooled below the `/export/install` directory. To run the `check` script on a JumpStart server, complete the following steps:

1. Change the directory to the location of the `check` script.

```
# cd /export/install/Solaris_9/Misc/JumpStart_sample
```

2. Copy the `check` script to the `/export/config` directory.

```
# cp check /export/config
```

3. Change the directory to `/export/config`, and run the `check` script.

```
# cd /export/config
# ./check
Validating rules...
Validating profile profile1...
The custom JumpStart configuration is ok.
#
```

4. If the `check` script reports an error, edit the `rules` or profile file to correct the problem indicated. In the following example, the `profile1` file contains a spelling error. For the example, the misspelling of the keyword, `filesys`, causes the `check` script to report the following output:

```
Validating rules...
Validating profile profile1...

Error in file "profile1", line 4
        fileys c0t0d0s0 free /
ERROR: Invalid keyword
```

Running the `add_install_client` Script

The `add_install_client` script configures the boot server to provide the network boot services that JumpStart clients require. Options to the `add_install_client` script also let you specify what servers and what directories offer identification, configuration, and installation services.

The `add_install_client` script options and arguments must match how you have configured these services on the servers that you intend to use. In the following example, one server provides all the services for the JumpStart procedure. Run the `add_install_client` script only on the server that provides the boot image.

You must run the `add_install_client` script once for each JumpStart client.

Before you run the `add_install_client` script, edit the `/etc/inet/hosts` and `/etc/ethers` files on the boot server, and add a JumpStart client entry to each file. The following example shows how an entry for `client1` in the `/etc/inet/hosts` file appears:

```
192.10.10.4     client1
```

An entry for `client1` in `/etc/ethers` could appear as follows:

```
8:0:20:9c:88:5b client1
```

For this basic JumpStart configuration procedure, the `add_install_client` script requires that you specify the following information:

- The server and path where the `rules` and profile files are located (the `-c` option)
- The server and path where the `sysidcfg` file is located (the `-p` option)
- The name of the client
- The `kernel` architecture of the client

The following example supplies the required information for a client called `client1`:

```
# ./add_install_client -c server1:/export/config -p
server1:/export/config client1 sun4u
saving original /etc/dfs/dfstab in /etc/dfs/dfstab.orig
Adding "share -F nfs -o ro,anon=0 /export/install" to /etc/dfs/dfstab
making /tftpboot
enabling tftp in /etc/inetd.conf
starting rarpd
starting bootparamd
starting nfsd's
starting nfs mountd
updating /etc/bootparams
copying inetboot to /tftpboot
#
```

The `add_install_client` script automatically makes the changes required to support RARP, TFTP, the `bootparams` file, and NFS requests from the client, but it only causes the server to share the `/export/install` directory. Sharing the `/export/install` directory allows the JumpStart client to mount a root (`/`) file system during the network boot process, and to gain access to the installation image.

The following example shows that for the client to mount the configuration directory from the server, you must edit the `/etc/dfs/dfstab` file and add an entry to share the `/export/config` directory:

```
share -o ro /export/config
```

This line in the /etc/dfs/dfstab file would share the /export/config directory as a read-only directory.

Once this line exists in the /etc/dfs/dfstab file, you must run the shareall command to make the share take effect.

The following procedure assumes that the Solaris 9 OE boot and installation images have been spooled below the /export/install directory, and that the rules, profile, and sysidcfg files you intend to use exist in the /export/config directory. To run the add_install_client script on a JumpStart server, complete the following steps:

1. Edit the /etc/inet/hosts file, and add an entry for the JumpStart client.
2. Edit the /etc/ethers file, and add an entry for the JumpStart client.
3. Change the directory to the location of the add_install_client script on the server.

```
# cd /export/install/Solaris_9/Tools
```

4. Run the add_install_client script, and specify server and client information as follows:

```
# ./add_install_client -c server_name:/export/config -p
server_name:/export/config client_name platform_group
saving original /etc/dfs/dfstab in /etc/dfs/dfstab.orig
Adding "share -F nfs -o ro,anon=0 /export/install" to /etc/dfs/dfstab
making /tftpboot
enabling tftp in /etc/inetd.conf
starting rarpd
starting bootparamd
starting nfsd's
starting nfs mountd
updating /etc/bootparams
copying inetboot to /tftpboot
#
```

   a. For *server_name*, enter the name of the server where you are running the add_install_client script.
   b. For *client_name*, enter the name of the JumpStart client.
   c. For *platform_group*, enter the correct kernel architecture for the JumpStart client, for example, sun4u.

5. Edit the /etc/dfs/dfstab file to add the following line:

```
share -o ro /export/config
```

6. Run the `shareall` command to share the `/export/config` directory.

```
# shareall
```

7. Verify that the `/export/config` and `/export/install` directories are currently shared.

```
# share
-               /export/install   ro,anon=0   ""
-               /export/config   ro   ""
```

Booting the JumpStart Client

After the JumpStart server has been configured to provide all of the required services, you can initiate the installation process on the JumpStart client.

To boot the JumpStart client, perform the following steps:

1. Bring the JumpStart client to run state `0`.

```
# init 0
```

2. Boot the client to initiate the software installation using the JumpStart procedure.

```
ok boot net - install
```

Exercise: Configuring a Software Installation Procedure Using JumpStart

In this lab, you configure a JumpStart server to support one install client.

Preparation

This exercise requires a functioning NIS environment. Use the NIS master server as the JumpStart server, and the NIS client as the install client. Do not use any existing NIS slave servers as JumpStart clients.

Task Summary

Perform the following tasks:

- Except as noted otherwise, perform all steps on the NIS master server.
- The exercise requires adding the `locale` map to the NIS `Makefile` file. Steps in the exercise describe how to do this. Alternately, your instructor might have an updated `Makefile` file available. Verify that the `aliases` entry is removed from the target `all` in the `Makefile` file.
- Verify that the `/etc/bootparams`, `/etc/timezone`, `/etc/ethers`, and `/etc/netmasks` files exist and are used by NIS.
- Locate the Solaris 9 Software 1 of 2 CD-ROM. The JumpStart server will share this CD to allow the client to install the OE.
- Determine the Ethernet (MAC) address of the client system.
- Unshare any NFS shared directories and remove any share commands from the `/etc/dfs/dfstab` file.
- This exercise demonstrates loading the End User configuration cluster from a shared Solaris 9 Software 1 of 2 CD-ROM. You can only load the Core and End User configuration clusters using JumpStart procedures in this way. Software installations using the Developer, Entire Distribution, or Entire Distribution with OEM support configuration clusters require loading a Solaris 9 image to disk from the Solaris 9 Software 1 of 2 and 2 of 2 CD-ROMs, and using that image to load JumpStart clients. Refer to your lecture notes as necessary to perform the steps listed.

Worksheet for Configuring a Software Installation Procedure Using JumpStart Software

Complete the following worksheet before you begin.

Install server name: _____

Timehost server name: _____

Note – Without an assigned `timehost` entry for one of the Solaris OE, the JumpStart process becomes interactive, prompting you for time information. The NIS master is a good candidate for this exercise.

Directory containing the Solaris OE installation image:
_____

Configuration server name: _____

Configuration directory: _____

Boot server name: _____

Directory containing the boot image: _____

JumpStart client's name: _____

JumpStart client's IP address: _____

JumpStart client's Ethernet address: _____

JumpStart client's architecture: _____

Tasks

Complete the following steps:

1. On the NIS master server, log in as `root`. Open a terminal window, and change the directory to the `/etc` directory.

```
# cd /etc
```

2. Edit the `/etc/ethers` file, and add an entry for the JumpStart client; for example:

```
8:0:20:2f:90:3d    client1
```

3. Edit the `/etc/hosts` file, and add an entry for the JumpStart client, if one does not already exist. Add the `timehost` alias to the JumpStart server's entry; for example:

```
192.9.200.1      server1   loghost    timehost
192.9.200.100    client1
```

4. Edit or check the `/etc/netmasks` file to be certain that it contains the network number and subnet mask for your network, for example:

```
192.9.200.0 255.255.255.0
```

5. Edit the `/etc/timezone` file, and add an entry that associates your local time zone with the name of your NIS domain. Entries in this file are case sensitive, for example:

```
US/Mountain       yourdomain
```

6. Edit the `/etc/locale` file, and add an entry that associates your locale with the name of your NIS domain. Entries in this file are case sensitive; for example:

```
yourdomain        en_US
```

7. Update or replace the `/var/yp/Makefile` file so that it includes entries to support using a map for the `/etc/locale` file.

Note – Your instructor might have an updated `Makefile` file available.

To update `/var/yp/Makefile` file so that it includes the `locale` map, make the following changes. This procedure also removes the `aliases` entry from the target `all`.

a. Change the directory to `/var/yp`, and edit the `Makefile` file.

```
# cd /var/yp
# vi Makefile
```

b. Add the following text after the existing `timezone.time` entry; all beginning white space must be tabs. Because the entry in the `Makefile` file for the `timezone` map contains identical code except for the map name, you can duplicate the `timezone` entry, and replace `timezone` with `locale`.

```
locale.time:  $(DIR)/locale
        -@if [ -f $(DIR)/locale ]; then \
                sed -e "/^#/d" -e s/#.*$$// $(DIR)/locale \
                | awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \
                | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/locale.byname; \
                touch locale.time; \
                echo "updated locale"; \
                if [ ! $(NOPUSH) ]; then \
                        $(YPPUSH) locale.byname; \
                        echo "pushed locale"; \
                else \
                : ; \
                fi \
        else \
                echo "couldn't find $(DIR)/locale"; \
        fi
```

c. Add the word `locale` after the word `timezone` on the line beginning with the word `all`.

d. Following the `timezone:  timezone.time` line, add the line:

```
locale:  locale.time
```

e. Save the file, and exit the editor.

8. Update the NIS maps by running the `make` command.

```
# cd /var/yp
# /usr/ccs/bin/make
...
<Control>C
```

Note – The `make` command hangs when it tries to push the new `locale` map to the slave server. Press Control-C to stop the `make` command when this happens.

9. On the NIS slave server, use the `ypxfr` command to transfer the `locale.byname` map for the first time.

```
# /usr/lib/netsvc/yp/ypxfr locale.byname
```

10. On the NIS master server, update the NIS maps by running the `make` command. This time, the `make` command should complete successfully.

```
# cd /var/yp
# /usr/ccs/bin/make
```

11. Insert the Solaris 9 Software 1 of 2 CD-ROM in the CD-ROM drive. Create the `/export/config` directory.

```
# mkdir /export/config
```

12. Change the directory to `/cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample`.

```
# cd /cdrom/cdrom0/s0/Solaris_9/Misc/jumpstart_sample
```

13. Copy the content of the `jumpstart_sample` directory to the `/export/config` directory. This step places sample configuration files, used by the JumpStart procedure, in the `/export/config` directory, which you will use to complete the exercise.

```
# cp -r  *  /export/config
```

14. Change the directory to `/export/config`. Move the `rules` file to `rules.orig`.

```
# cd /export/config
# mv rules rules.orig
```

15. Create a new file called `rules` that contains the following entry. Enter the name of your JumpStart client instead of `client1`:

```
hostname  client1    -   host_class    finish_script
```

16. Edit the /export/config/host_class file so that it specifies an initial install; a standalone system type; explicit partitioning; the End User software cluster; and partitions for root (/), swap, and /usr. Use partition sizes and device names appropriate for the JumpStart client system; for example:

```
install_type    initial_install
system_type     standalone
partitioning    explicit
cluster         SUNWCuser
filesys         c0t0d0s0 300 /
filesys         c0t0d0s1 128 swap
filesys         c0t0d0s6 free /usr
```

17. In the /export/config directory, create a file called finish_script that contains the following lines. Replace *yourdomain* with your NIS domain name.

```
#!/bin/sh
touch /a/noautoshutdown
rm /a/etc/defaultdomain
rm -r /a/var/yp/binding/yourdomain
cp /a/etc/nsswitch.files /a/etc/nsswitch.conf
```

These commands configure the JumpStart client to avoid using the autoshutdown power-saving feature, and they remove the NIS client configuration.

18. Change the permissions on finish_script to 555.

```
# chmod 555 finish_script
```

19. Run the /export/config/check program, and correct any problems in the rules or host_class files that it reports. Verify that the rules.ok file exists after the check program completes successfully.

```
# ./check
```

20. In the /export/config directory, create a file called sysidcfg that contains the following lines. The string Hx23475vABDDM is a 13-character encrypted string for the password cangetin. You could replace this string with a different encrypted password string by copying one from your own /etc/shadow file.

```
security_policy=none
network_interface=primary {protocol_ipv6=no
                           default-route=<client_IP> }
root_password=Hx23475vABDDM
```

Note – These lines answer the installation questions about implementing Kerberos security and the IPv6 protocol and supply a `root` password.

21. Edit the `/etc/dfs/dfstab` file to add an entry for the `/export/config` directory as follows:

```
share -o ro /export/config
```

22. If the NFS server daemons are not running, start them.

```
# /etc/init.d/nfs.server start
```

23. If the NFS server daemons are already running, run the `shareall` command:

```
# shareall
```

24. Change the directory to `/cdrom/cdrom0/s0/Solaris_9/Tools`.

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```

25. Use the `add_install_client` program to add support for your JumpStart client. The following command example is appropriate for a server that will provide access to the operating environment using a mounted Solaris 9 Software 1 of 2 CD-ROM. Replace *`server1`* with the name of your JumpStart server, *`client1`* with the name of your JumpStart client, and *`sun4x`* with either `sun4u`, `sun4m`, or `sun4c`, depending on the type of client system that you are using.

```
# ./add_install_client -c server1:/export/config \
-p server1:/export/config client1 sun4x
```

What action does the `add_install_client` program report that it takes regarding the files and daemons in Table 17-3:

Table 17-3 Results of `add_install_client` Program

File or Daemon	Action
`/etc/dfs/dfstab` file	
`/etc/inetd.conf` file	
`/etc/nsswitch.conf` file	
`/tftpboot` file	
`rarpd` daemon	
`bootparamd` daemon	

26. Update the NIS maps by running the `make` command.

```
# cd /var/yp
# /usr/ccs/bin/make
```

27. Boot the JumpStart client.

```
ok boot net - install
```

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Task Solutions

26. What actions does the `add_install_client` program report that it takes regarding the files and daemons in Table 17-4:

Table 17-4 Results of `add_install_client` Program

File or Daemon	Action
`/etc/dfs/dfstab` file	*Copies the original to* `dfstab.orig`*, and adds a line to share slice 0 of the CD*
`/etc/inetd.conf` file	*Enables* `tftp`
`/etc/nsswitch.conf` file	*Changes the* `bootparams` *entry*
`/tftpboot` file	*Creates the directory, copies* `inetboot.SUN4U.Solaris_9-1` *into it*
`rarpd` daemon	*Starts this daemon*
`bootparamd` daemon	*Starts this daemon*

Setting Up JumpStart Software Configuration Alternatives

The JumpStart procedure supports a range of alternative server and client configurations. Depending on your network configuration, available server resources, and the client configurations that you want, you can:

- Set up all JumpStart services on a single server
- Configure one server per subnet to provide boot services separately from the other JumpStart services
- Configure boot, identification, configuration, and installation services on separate servers
- Configure a name service to provide identification information
- Configure begin scripts and finish scripts to further customize software installation on JumpStart clients

The flexibility in server and client configuration lets you build JumpStart services to meet your specific software installation needs.

Introducing the JumpStart Client Boot Sequence

To understand the services that a boot server provides, it is useful to know how a JumpStart client boots using the network (Figure 17-3).

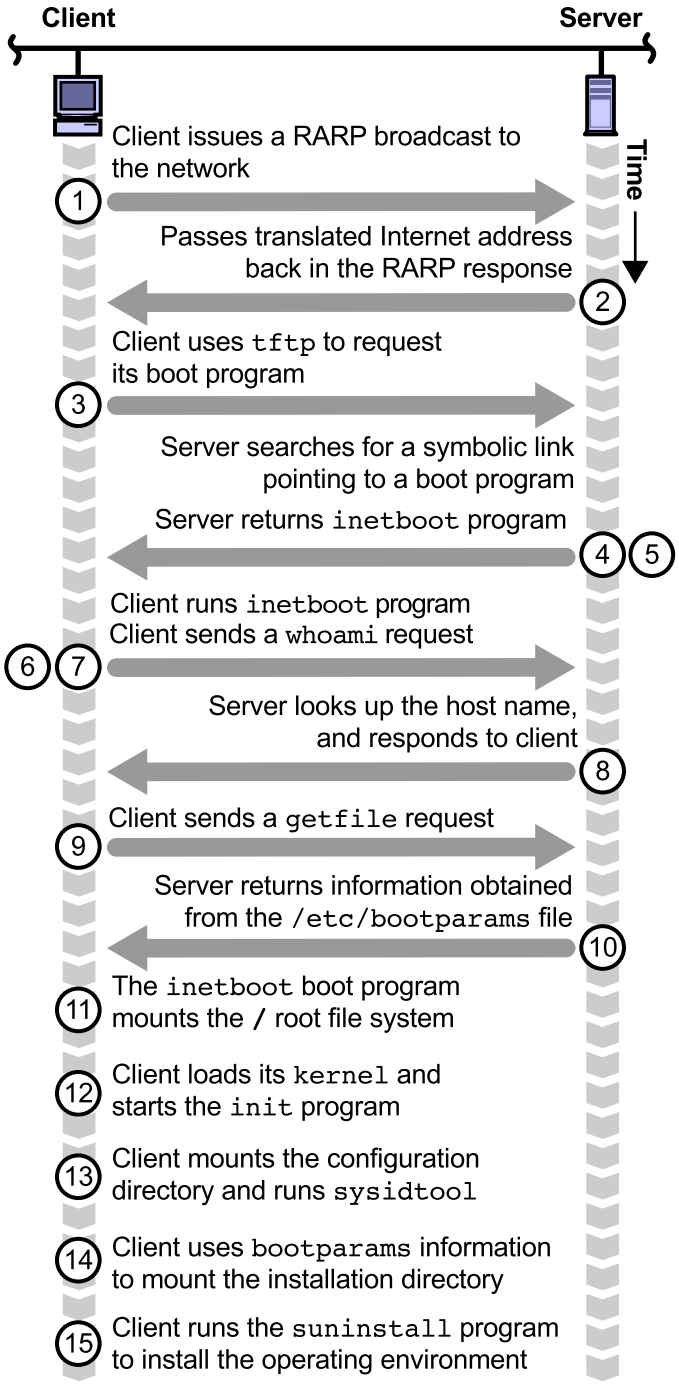


Figure 17-3 The JumpStart Boot Process

Figure 17-3 on page 17-34 shows the JumpStart client boot process. The following steps describe how a JumpStart client boots from a boot server, and starts the installation process:

1. When a JumpStart client boots, the boot PROM broadcasts a RARP request to the local subnet.
2. The `in.rarpd` daemon on the boot server processes the client's RARP request by:
   a. Looking up the client's Ethernet address and host name in the `/etc/ethers` file
   b. Checking for a corresponding host name in the `/etc/hosts` file
   c. Returning the associated IP address to the client
3. The client's boot programmable read-only memory (PROM) sends a TFTP request for a network bootstrap program.
4. The `in.tftpd` daemon on the boot server processes the client's TFTP request. The daemon searches the `/tftpboot` directory for a file with a hexadecimal representation of the client's IP address. The hexadecimal representation is the name of the file. This file is a symbolic link that points to a network bootstrap program.
5. The `in.tftpd` daemon on the boot server returns the network bootstrap program to the JumpStart client.
6. The JumpStart client runs the network bootstrap program.
7. The network bootstrap program issues a `whoami` request to discover the JumpStart client's host name.
8. The `rpc.bootparamd` daemon on the boot server looks up the client's host name, and returns it to the client.
9. The network bootstrap program issues a `getfile` request to obtain the location of the root (`/`) file system.
10. The server responds with the location of the root (`/`) file system, obtained from the `/etc/bootparams` file.
11. After the client obtains its boot parameters, the network bootstrap program mounts the root (`/`) file system from the boot server.
12. The client loads its `kernel` and starts the `init` program. When the JumpStart client finishes booting, it attempts to find configuration information.
13. The client searches for the configuration server using `bootparams` information. The client mounts the configuration directory, and runs the `sysidtool` daemon.

14. The client uses `bootparams` information to locate and mount the Solaris OE installation image.
15. The client runs the `suninstall` program and installs the Solaris OE.

For boot operations to continue, the following files and directories must be properly configured on the boot server:

- The `/etc/ethers` file
- The `/etc/inet/hosts` file
- The `/tftpboot` directory
- The `/etc/bootparams` file
- The `/etc/dfs/dfstab` file

Introducing the `/etc/ethers` and `/etc/inet/hosts` Files

A JumpStart client initially obtains its IP address through a RARP request while it boots. To obtain the RARP request, an entry for the client must exist in the `/etc/ethers` and `/etc/inet/hosts` files on the boot server, or exist in a name service.

The `/etc/ethers` file associates a Media Access Control (MAC) address with a client's host name. For example:

```
8:0:20:9c:88:5b client1
```

The `/etc/inet/hosts` file associates an IP address with a client's host name. For example:

```
192.10.10.4     client1
```

Generally, you configure this information by editing these files manually, and by updating the name service, if one is in place. With this information available in either the `/etc/ethers` and `/etc/inet/hosts` files on a boot server or in a name service, such as NIS or NIS+, the JumpStart client should be able to obtain the IP address and host name it needs to continue the boot process.

Introducing the /tftpboot Directory

JumpStart clients retrieve a network bootstrap program from the /tftpboot directory when they issue requests to the in.tftpd daemon running on the boot server. The in.tftpd daemon uses a symbolic link that is a hexadecimal representation of the client's IP address. This symbolic link locates a network bootstrap program to return to the /tftpboot directory. Different network bootstrap programs exist for different Solaris OE releases and client architectures.

In the following example, the symbolic link called C00A0A04 points to the network bootstrap program called inetboot.SUN4U.Solaris_9-1.

```
# cd /tftpboot
# ls -l
total 280
lrwxrwxrwx   1 root     other         26 Nov 19 17:31 C00A0A04 ->
inetboot.SUN4U.Solaris_9-1
```

The add_install_client script creates the required files in the /tftpboot directory when you run it to configure boot support for a JumpStart client. The platform group argument that you specify to the add_install_client script selects the bootstrap program appropriate for the client's kernel architecture. Running the add_install_client script from a Solaris 9 OE image automatically selects a bootstrap program specific to the Solaris 9 OE.

Note – Use the bc utility for a quick conversion from IP numbers to hexadecimal numbers. Run the bc utility, and press the Return key. Then enter obase=16. Enter each of the IP fields, one at a time, to get the hexadecimal conversion. Thus, 192 = C0, 10 = 0A, 10 = 0A, and 4 = 04. Putting it all together, the resultant hexadecimal IP number is C00A0A04. Press Control-D to exit the bc utility.

Describing the /etc/bootparams File

JumpStart clients retrieve information from the /etc/bootparams file when they issue requests to the rpc.bootparamd daemon that runs on the boot server. The rpc.bootparamd daemon references the /etc/bootparams file and returns the information to the client. The client system uses this information to mount the directories that it requires using the NFS service.

The `add_install_client` script updates the `/etc/bootparams` file when you run it to configure boot support for a JumpStart client. The `/etc/bootparams` file contains one entry for each JumpStart client that the boot server supports. Each entry lists the servers and directories that provide boot, identification, configuration, and installation services.

The options and arguments that you specify when you run the `add_install_client` script determine the content of the `/etc/bootparams` file. The following example describes an example entry in the `/etc/bootparams` file for a JumpStart client named `client1`:

```
client1
root=server1:/export/install/Solaris_9/Tools/Boot
install=server1:/export/install
boottype=:in
sysid_config=server1:/export/config
install_config=server1:/export/config
rootopts=:rsize=32768
```

The `add_install_client` command that creates the `/etc/bootparams` entry in the following example is:

```
# cd /export/install/Solaris_9/Tools
# ./add_install_client -c server1:/export/config -p
server1:/export/config client1 sun4u
```

Table 17-5 describes the example entries in the `/etc/bootparams` file.

Table 17-5 Entries in the `/etc/bootparams` File

Entry	Definition
`client1`	Specifies the JumpStart client name.
`root=server1:/export/install/Solaris_9/Tools/Boot`	Lists the boot server name and directory where the root (/) file system is found. This path is derived from the server and directory where you run the `add_install_client` script.
`install=server1:/export/install`	The server name and directory where the Solaris OE installation image is found. Unless you use the `-s` option, this path is derived from the server and directory where you run the `add_install_client` script.
`boottype=:in`	Indicates that `client1` is an install client. This entry is the default client type created by the `add_install_client` script.
`sysid_config=server1:/export/config`	Lists the server name and directory where the `sysidcfg` file is found. This path is taken from the `-p` option and argument to the `add_install_client` script.
`install_config=server1:/export/config`	Lists the server name and directory where the `rules` and profile files are found. This path is taken from the `-c` option and argument to the `add_install_client` script.
`rootopts=:rsize=32768`	Lists the mount options for the root (/) file system and the NFS read size.

The /etc/dfs/dfstab File

JumpStart clients require access to directories that servers make available using NFS. Placing an entry for a directory in the /etc/dfs/dfstab file on a server lets the server automatically share the directory when it boots. The add_install_client script creates only one entry in the /etc/dfs/dfstab file on the boot server. This entry shares the location of the boot and installation images. For example:

```
share -F nfs -o ro,anon=0 /export/install
```

The ro and anon=0 options for the share directory in this example let JumpStart clients mount the directory as read-only and retain their root user privileges for the mount.

You must share any other directory that JumpStart clients require with the server that provides it. Generally, you must manually edit the /etc/dfs/dfstab file to create entries for these directories. For example, if a separate server provides JumpStart configuration information, the /etc/dfs/dfstab file on that server must contain an entry for it:

```
share -o ro /export/config
```

Before a JumpStart client can boot and obtain all of the NFS resources it requires, every directory listed as an argument to the add_install_client script must be shared by the server on which it resides.

Setting Up a Boot-Only Server

Network configuration considerations or limits on server resources might require that you create JumpStart boot-only servers. A boot server responds to RARP, TFTP, and `bootparams` requests from JumpStart clients and provides a boot image using the NFS service.

In the `bootparams` information that the boot server offers, it identifies identification, configuration, and installation services.

Two main configuration steps are required to create a JumpStart boot server:

- Running the `setup_install_server` script with the `-b` option to spool a boot image from CD-ROM or DVD
- Running the `add_install_client` script with options and arguments that show a list of servers and the identification, configuration, and installation services that they provide

It is also possible to provide boot services from a shared CD-ROM or DVD, but this is not the most common or practical configuration.

Subnet Restrictions

JumpStart clients broadcast RARP requests when they attempt to boot from the network. Broadcast network traffic is normally not forwarded to networks other than the one where the broadcast traffic originated. This situation requires that a JumpStart boot server exist on the same subnet to which JumpStart clients are directly connected.

The initial network requests for boot-related services are the only JumpStart client requests that are limited by these subnet restrictions. Identification services can be provided by a `sysidcfg` file made available to the client by using NFS or by binding the JumpStart client to a name service in use. Configuration and installation services are also made available using the NFS service. The NFS service and name services generally allow for network traffic to route among subnets, but the services that depend on them can be provided by servers on different subnets from the one to which the client is directly attached.

Often, a single JumpStart server provides all JumpStart services. It might be necessary for various reasons to configure servers other than the boot server to respond to identification, configuration, or installation requests from JumpStart clients. In these cases, it is useful to create a boot server on the subnet where JumpStart clients reside.

Figure 17-4 shows a JumpStart network configuration with a separate boot server.

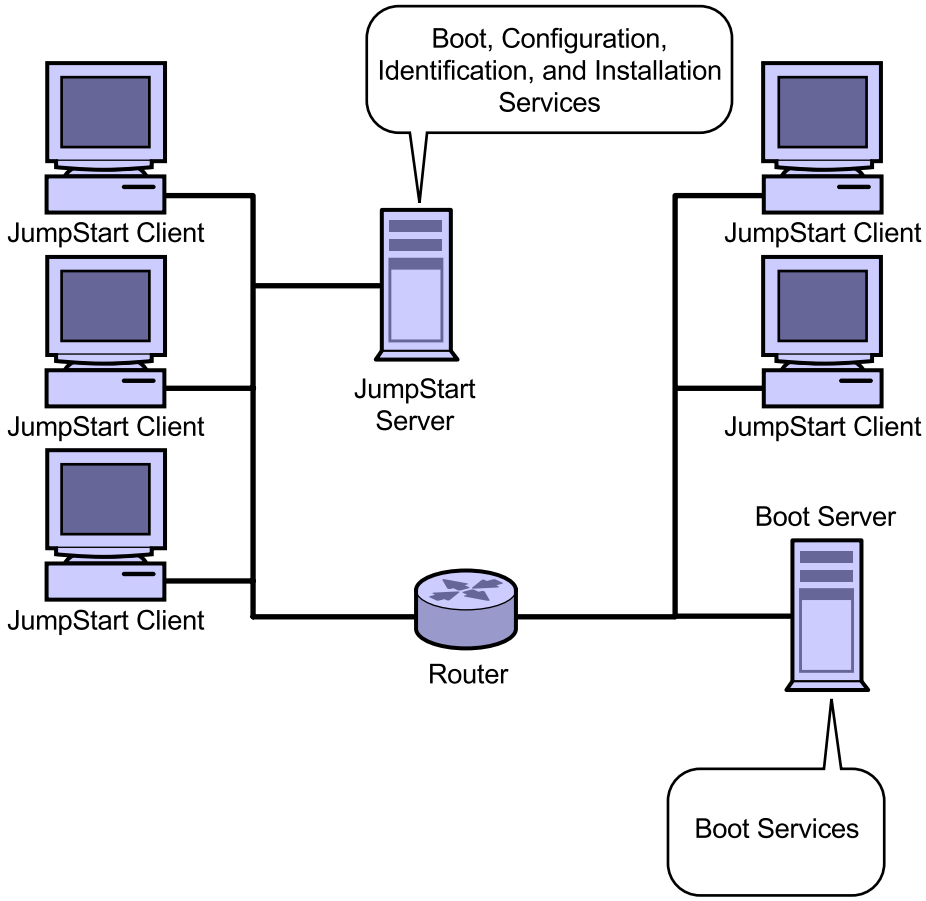


Figure 17-4 The JumpStart Boot Server

Executing the `setup_install_server` Script

To spool only the boot image from a Solaris 9 Software 1 of 2 CD-ROM or from the DVD, run the `setup_install_server` script with the `-b` option. In the Solaris 9 OE, the `setup_install_server` script spools a boot image that occupies about 260 Mbytes of disk space. All JumpStart clients that boot from this server use the same boot image.

To spool the Solaris 9 OE boot image to a local disk, complete the following steps on the system chosen as a boot server:

1. Create an empty directory with at least 280 Mbytes of space available to hold the Solaris OE boot image. The `/export/install` directory is usually used for this purpose.

```
# mkdir /export/install
```

2. Insert the Solaris 9 Software 1 of 2 CD-ROM in the CD-ROM drive, or the Solaris 9 DVD in the DVD drive. Allow the `vold` command to automatically mount the media.

3. Change the directory to the location of the `setup_install_server` script.

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```

4. Run the `setup_install_server` script with the `-b` option to copy the Solaris 9 OE boot image to the local disk (this process can take about 30 minutes).

```
# ./setup_install_server -b /export/install
```

5. When `setup_install_server` finishes, change directory to root (`/`), and eject the CD-ROM or DVD.

```
# cd /
# eject cdrom
```

Executing the `add_install_client` Script

The `add_install_client` script configures the boot server to offer the network boot services that JumpStart clients require. When you configure a boot-only server, you must specify options to the `add_install_client` script to indicate which servers and which directories will provide identification, configuration, and installation services.

The `add_install_client` options you use will *reflect*, rather than determine, how you have configured all of the required JumpStart services on other servers.

You must run the `add_install_client` script once for each JumpStart client.

Before you run the `add_install_client` script, update the `hosts` and `ethers` information for the JumpStart client.

If a name service is not in use, edit the `/etc/inet/hosts` and `/etc/ethers` files on the boot server, and add an entry to each file for the JumpStart client. For example, an entry for `client1` in the `/etc/inet/hosts` file could appear as follows:

```
192.10.10.4     client1
```

An entry for `client1` in the `/etc/ethers` file could appear as follows:

```
8:0:20:9c:88:5b client1
```

If a name service is in use, you must edit the `/etc/inet/hosts` and `/etc/ethers` files on the appropriate name service server, and run the commands required to update the name service maps or tables.

The `/etc/inet/hosts` file on the boot server must also contain an entry for each server you specify when you run the `add_install_client` script.

To add support for a JumpStart client to the boot server, assuming other servers provide all other JumpStart services, you would use `add_install_client` options that specify the following information:

- The server and path where the `rules` and profile files are located (the `-c` option)
- The server and path where the `sysidcfg` file is located (the `-p` option)
- The server and path where the installation image is located (the `-s` option)
- The name of the client
- The `kernel` architecture of the client

The `add_install_client` script automatically makes the changes required for the boot server to support RARP, TFTP, `bootparams`, and NFS requests from the client. The `add_install_client` script automatically causes the boot server to share the `/export/install` directory, if that is where the boot image is spooled. Sharing the `/export/install` directory lets the JumpStart client mount the boot image during the network boot process.

For the client to gain access to identification, configuration, and installation services, the following conditions must also exist:

- The server that provides the `sysidcfg` file must share the directory where the server is located.
- If you use a name service to provide identification information, it must be active and updated to provide the information that the client requires.
- The server that provides the `rules` and profile files must share the directory where these files are located.
- The server that provides the installation image must share the directory where it is located.

The following procedure assumes that the Solaris 9 OE boot image has been spooled below the `/export/install` directory on the boot server. To run the `add_install_client` script on a boot server, complete the following steps:

1. Update the `/etc/inet/hosts` information to add an entry for the JumpStart client.
2. Update the `/etc/ethers` information to add an entry for the JumpStart client.
3. Change the directory to the location of the `add_install_client` script on the server.

```
# cd /export/install/Solaris_9/Tools
```

4. Run the `add_install_client` script, and specify server and client information as follows:

```
# ./add_install_client -c server:/config_path -p server:/sysid_path -s
server:/install_path client_name platform_group
saving original /etc/dfs/dfstab in /etc/dfs/dfstab.orig
Adding "share -F nfs -o ro,anon=0 /export/install/Solaris_9/Tools/Boot"
to /etc/dfs/dfstab
making /tftpboot
enabling tftp in /etc/inetd.conf
starting rarpd
starting bootparamd
starting nfsd's
starting nfs mountd
updating /etc/bootparams
copying inetboot to /tftpboot
#
```

   a. For the *server*:*/config_path* field, enter the name of the server and path where the `rules` and profile files are located.

   b. For the *server*:*/sysid_path* field, enter the name of the server and path where the `sysidcfg` file is located.

   c. For the *server*:*/install_path* field, enter the name of the server and path where the installation image is located.

   d. For the *client_name* field, enter the name of the JumpStart client.

   e. For the *platform_group* field, enter the correct `kernel` architecture for the JumpStart client, for example, `sun4u`.

When you complete this procedure, and meet conditions on the other servers, you can initiate the installation process on a JumpStart client.

Setting Up Identification Service Alternatives

JumpStart clients can obtain the identification information that they require from different sources, including the `/etc/inet/hosts` file on a boot server, the `sysidcfg` file, or a name service, such as NIS or NIS+. Identification information provided in a `sysidcfg` file takes precedence over information provided by other sources.

Configuring /etc/inet/hosts and /etc/ethers Files

If a name service is not in use, a JumpStart client obtains its IP address and host name from the /etc/inet/hosts file found on the boot server. If a name service is in use, a JumpStart client obtains its IP address and host name from the hosts map or table in that source.

Configuring the /etc/inet/hosts file only requires that you create an entry appropriate for the JumpStart client in that file. For example:

```
192.10.10.4      client1
```

For a JumpStart client to initially obtain its IP address from a RARP request as it boots, an entry for the client must also exist in the /etc/ethers file on the boot server. For example:

```
8:0:20:9c:88:5b client1
```

If a name service is in use, the maps or tables that contain /etc/inet/hosts and /etc/ethers information must include entries for the JumpStart client.

Configuring the sysidcfg File

JumpStart clients use information in the sysidcfg file to answer identification questions. Information in this file replaces identification information available to the client from other sources. If the JumpStart client cannot obtain a response for an identification question, the client interrupts the automatic identification process and asks for the information.

In the Solaris 8 OE, JumpStart clients require a sysidcfg file to answer IPv6 and Kerberos-related identification questions. In the Solaris 9 OE, the sysidcfg file must also contain an entry identifying a default router.

The sysidcfg file allows you to specify nearly all of the identification information that a JumpStart client requires. The sysidcfg file can contain:

- Identification information that all JumpStart clients can use
- Information that is client-specific

If you supply client-specific information in the `sysidcfg` file, you must create a separate `sysidcfg` file for each client. You must name the file `sysidcfg` on each system. Therefore, if you specify client-specific information in the `sysidcfg` file, you must place each unique `sysidcfg` file in a separate directory.

Locating the `sysidcfg` File

Typically, you would create a generic `sysidcfg` file in the `/export/config` directory on a JumpStart server. The `sysidcfg` files that contain client-specific information must exist in separate directories. For example, the `/export/config/client1/sysidcfg` directory.

JumpStart clients learn of the location of the `sysidcfg` file from `bootparams` information that they obtain from the boot server. When you run the `add_install_client` script on the boot server, use the `-p` option, and specify the server and path where the `sysidcfg` file is stored, for example, the following command:

```
# ./add_install_client -c server1:/export/config -p
server1:/export/config client1 sun4u
```

indicates that the `sysidcfg` file that `client1` will use is found on the server, `server1` in the `/export/config` directory. The server, `server1`, must share the `/export/config` directory by using the NFS service before the client can mount it.

Constructing the `sysidcfg` File

The `sysidcfg` file lets you specify many different identification items. Entries in the `sysidcfg` file must conform to the following rules:

- Independent keywords can be listed in any order.
- Keywords are not case sensitive.
- Keyword values can be optionally enclosed in single (') or double (") quotation marks.
- Dependent keywords must be enclosed in curly braces ({}) to tie them to their associated independent keyword.
- Only the first instance of a keyword is valid. If a keyword is specified more than once, only the first keyword specified is used.

Table 17-6 lists the keywords and arguments used in the construction of the `sysidcfg` file.

Table 17-6 Keywords and Arguments Used in Constructing the `sysidcfg` File

Keywords	**Arguments**
`name_service {domain_name}`	`name_service=NIS, NIS+, DNS, LDAP, OTHER, NONE`
	Options for NIS and NIS+: `{domain_name=`*`domain_name`* `name_server=`*`hostname`*`(`*`ip_address`*`)}`

	Options for DNS:
	`{domain_name=`*`domain_name`*
	`name_server=`*`ip_address`*`,`*`ip_address`*`,`
	`ip_address` (three maximum)
	`search=`*`domain_name`*`,`*`domain_name`*`,`
	`domain_name``,`*`domain_name`*`,`
	`domain_name``,`*`domain_name`*
	(six maximum, the total length is less than or equal to 250 characters)`}`

	Options for LDAP: `{domain_name=`*`domain_name`* `profile=`*`profile_name`* `profile_server=`*`ip_address`*`}`

`network_interface`, `hostname`, `ip_address` (Internet Protocol [IP] address), `netmask`, DHCP, IPv6	`network_interface=NONE,PRIMARY,`
	or *`value`*
	`{hostname=`*`hostname`*
	`ip_address=`*`ip_address`*
	`netmask=`*`netmask`* `protocol_ipv6=yes/no}`

	If DHCP *is* used, specify: `{dhcp protocol_ipv6=yes/no}`
	If DHCP is *not* used, specify: `{hostname=`*`host_name`* `default_route=`*`ip_address`* `ip_address=`*`ip_address`* `netmask=`*`netmask`* `protocol_ipv6=yes/no}`
`root_password`	`root_password=`*`root_password`* (encrypted password from `/etc/shadow`)

Table 17-6 Keywords and Arguments Used in Constructing the `sysidcfg` File (Continued)

Keywords	Arguments
`security_policy`	`security_policy=kerberos`, `NONE`
	Options for `kerberos`: `{default_realm=`*`FQDN`* `admin_server=`*`FQDN`* `kdc=`*`FQDN1`*`,`*`FQDN2`*`,`*`FQDN3`*`}` where *`FQDN`* is a fully qualified domain name. You can list a maximum of three key distribution centers (KDCs), but at least one is required.
`system_locale`	`system_locale=`*`locale`* (entry from the `/usr/lib/locale` file)
`terminal`	`terminal=`*`terminal_type`* (entry from the `/usr/share/lib/terminfo` database)
`timezone`	`timezone=`*`timezone`* (entry from `/usr/share/lib/zoneinfo` file)
`timeserver`	`timeserver=localhost`, *`hostname`*, or *`ip_addr`*

Example of the sysidcfg File

The following is an example of the `sysidcfg` file:

```
network_interface=primary {protocol_ipv6=no
                           netmask=255.255.255.0
                           default_route=192.10.10.1}
security_policy=none
name_service=none
timezone=US/Mountain
system_locale=en_US
timeserver=192.10.10.1
root_password=Hx23475vABDDM
```

Note – The encrypted `root_password` entry in this example represents the password `cangetin`.

Configuring NIS for JumpStart Procedures

JumpStart clients can use the NIS to obtain most of the identification information that they would otherwise obtain from the `/etc/inet/hosts` file on the boot server and the `sysidcfg` file on a configuration server. Configuring NIS to support JumpStart procedures involves editing files and running commands on the NIS master server in use.

In the Solaris 9 OE, name services cannot provide responses for the IPv6, Kerberos, default route, and `root` password questions that clients ask. The `sysidcfg` file offers the only means of automatically supplying these responses to clients. NIS can supply all of the other essential identification information that clients require.

Information supplied in the `sysidcfg` file overrides any information you make available in NIS. The following sections describe how to configure the files that NIS uses to create maps, and the procedures required to update NIS with the information you provide in those files. The following sections assume that a functional NIS domain exists, and that all JumpStart servers participate in the NIS domain as NIS clients.

A change to any file that is represented by a map in an NIS domain requires that you complete the following steps on the NIS master server.

1. Edit and save the file that requires the change.
2. Change the directory to `/var/yp`.

```
# cd /var/yp
```

3. Enter the `make` command.

```
# /usr/ccs/bin/make
```

Configuring the `/etc/inet/hosts` File

The NIS map that represents the `/etc/inet/hosts` file can hold three identification items that JumpStart clients use:

- The JumpStart client's IP address
- The JumpStart client's host name
- The `timehost` alias

JumpStart clients recognize the `timehost` alias if it exists in a NIS map. JumpStart clients do not use the `timehost` alias directly from the `/etc/inet/hosts` file.

To configure NIS to respond to RARP requests from the JumpStart client, edit the `/etc/inet/hosts` file on the NIS master server to include an entry for the JumpStart client. The following example shows an entry for `client1` in the `/etc/inet/hosts` file:

```
192.10.10.4      client1
```

Note – Enabling RARP support in NIS also requires changes to the `/etc/ethers` file on the NIS master server.

To configure NIS to supply time-of-day information that the JumpStart clients require, you must add a `timehost` entry to the `/etc/inet/hosts` file. For example, the following entry would let JumpStart clients obtain their time-of-day information from the system that uses the IP address `192.10.10.1`.

```
192.10.10.1 server1 timehost
```

Usually, you would associate the `timehost` alias with a JumpStart server or the NIS master server.

After you complete the changes to the `/etc/inet/hosts` file, you must update the associated NIS map by running the `/usr/ccs/bin/make` command.

Configuring the `/etc/ethers` File

To configure NIS to respond to RARP requests that JumpStart clients issue, you must edit the `/etc/ethers` file on the NIS master server to include an entry for the JumpStart client. For example, an entry for `client1` in the `/etc/ethers` file could appear as follows:

```
8:0:20:9c:88:5b client1
```

After you complete the changes to the `/etc/ethers` file, you must update the associated NIS map by running the `/usr/ccs/bin/make` command.

Configuring the /etc/locale File

To configure NIS to respond to localization requests issued by JumpStart clients, you must create and configure an `/etc/locale` file on the NIS master server, and update the NIS `Makefile` to use it. The `/etc/locale` file does not exist in a default Solaris 9 OE installation, and no reference to this file exists in the default `/var/yp/Makefile` file.

Use a text editor to create an `/etc/locale` file with the appropriate content. The following example shows an entry for `client1` in the `/etc/locale` file:

```
client1       en_US
```

An entry for all systems in the NIS domain called `Central.Sun.Com` in the `/etc/locale` file could appear as follows:

```
Central.Sun.COM       en_US
```

Note – For a list of possible `locale` entries for this file, run the `locale -a` command, or list the locales found in the `/usr/lib/locale` directory.

To update the /var/yp/Makefile file on the NIS master server so that it includes the locale map, make the following changes:

1. Change the directory to /var/yp, and edit the Makefile file.

```
# cd /var/yp
# vi Makefile
```

   a. Add the following text after the existing *.time entries; all beginning white space must be tabs. The entry in the Makefile file for the timezone map contains identical code except for the map name; therefore, duplicate the timezone entry, and replace timezone with locale.

```
locale.time:  $(DIR)/locale
        -@if [ -f $(DIR)/locale ]; then \
               sed -e "/^#/d" -e s/#.*$$// $(DIR)/locale \
               | awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \
               | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/locale.byname; \
               touch locale.time; \
               echo "updated locale"; \
               if [ ! $(NOPUSH) ]; then \
                       $(YPPUSH) locale.byname; \
                       echo "pushed locale"; \
               else \
               : ; \
               fi \
        else \
               echo "couldn't find $(DIR)/locale"; \
        fi
```

   b. Append the word locale to the line beginning with the word all.

   c. Add the following line after the auto.home: auto.home.time entry:

```
locale:  locale.time
```

   d. Save the file, and exit the editor.

2. Update the NIS maps by running the make command.

```
# cd /var/yp
# /usr/ccs/bin/make
...
<Control>-C
#
```

The make command hangs when it tries to push the new locale map to slave servers. Press Control-C to stop the make command if the command hangs.

3. On any slave servers that exist in this NIS domain, run the `ypxfr` command to transfer the `locale.byname` map for the first time.

```
# /usr/lib/netsvc/yp/ypxfr locale.byname
```

4. On the NIS master server, again update the NIS maps by running the `make` command.

```
# cd /var/yp
# /usr/ccs/bin/make
```

The `make` command should complete successfully.

Configuring the `/etc/timezone` File

To configure NIS to respond to time zone requests that JumpStart clients issue, you must create or edit the `/etc/timezone` file on the NIS master server to include an entry for the client. The `/etc/timezone` file does not exist in a default Solaris 9 OE installation. For example, an entry for `client1` in `/etc/timezone` could appear as follows:

```
US/Mountain        client1
```

An entry for all systems in the NIS domain called `Central.Sun.COM` in `/etc/timezone` could appear as follows:

```
US/Mountain        Central.Sun.COM
```

After you have completed the changes to the `/etc/timezone` file, you must update the associated NIS map by running the `/usr/ccs/bin/make` command.

Note – Possible time zone entries for this file exist in the `/usr/share/lib/zoneinfo` directory.

Configuring the `/etc/netmasks` File

To configure NIS to respond to requests for netmask information that JumpStart clients issue, you must edit the `/etc/netmasks` file on the NIS master server. The `/etc/netmasks` file must include an entry for the network to which the JumpStart client is directly connected.

The /etc/netmasks file contains network masks that implement IP subnets. This file supports both standard subnetting, as specified in RFC-950, and variable length subnets, as specified in RFC-1519. Each line in the /etc/netmasks file should consist of the network number, any number of spaces or tab characters, and the network mask to use on that network. You can specify network numbers and masks in the conventional IP '.' (dot) notation (such as IP host addresses, but use zeros for the host section). For example, you could use:

```
192.9.200.0     255.255.255.0
```

to specify that the Class C network 192.9.200.0 should use 24 bits to identify the network, and 8 bits to identify the host.

Note – Refer to the man page for the netmasks nouns for more examples of subnet masks.

After you complete the changes to the /etc/netmasks file, enter the /usr/ccs/bin/make command to update the associated NIS map.

Configuring the /etc/bootparams File

Even though it is possible for NIS to provide bootparams file information to JumpStart clients, the bootparams file information is obtained from the JumpStart boot server. The boot server is often not the same system that acts as the NIS master server.

Each time you run the add_install_client script on a boot server to provide boot support for a JumpStart client, the script checks the /etc/nsswitch.conf file for the bootparams entry. If the bootparams entry in the /etc/nsswitch.conf file lists the nis source before the files source, the add_install_client script reverses their order. For example, the following entry in the /etc/nsswitch.conf file before running the add_install_client script:

```
bootparams: nis files
```

would change to the following entry after running the add_install_client script:

```
bootparams: files nis
```

Typically, a JumpStart boot server would participate in NIS as a client. The `add_install_client` script changes the `/etc/nsswitch.conf` file to cause JumpStart clients to obtain their `bootparams` file information from the `/etc/bootparams` file on the boot server, instead of changing the file from NIS. In most JumpStart configurations, this is the most practical situation.

Configuring the `sysidcfg` File With NIS

If you use NIS to supply all of the identification items it can possibly offer to JumpStart clients, only four items are required in the `sysidcfg` file. These items answer the IPv6, Kerberos, default router, and `root` password questions in the Solaris 9 OE.

The following example `sysidcfg` file causes the client to not implement IPv6 nor Kerberos security, sets the default route to be `192.10.10.100`, and sets the `root` password to `cangetin`. Use values that are appropriate for your own systems and network.

```
network_interface=primary {protocol_ipv6=no
                           default_route=192.10.10.100}
security_policy=none
root_password=Hx23475vABDDM
```

The absence of the `root_password` entry does not interfere with the system identification process the client performs before installing the Solaris OE. Without this entry in the `sysidcfg` file however, the client will ask for a `root` password the first time it reboots after the Solaris OE installation completes. NIS cannot supply the `root` password.

Setting Up Configuration Service Alternatives

You can customize how JumpStart clients load and configure the Solaris OE. Entries in the `rules` and profile files establish the basic Solaris OE configuration that a JumpStart client will use. Begin and finish scripts further customize the software installation process.

Matching Clients to Configurations

The `rules` file contains entries that allow JumpStart clients to select an installation profile. Each entry in the `rules` file lists one or more identifying characteristics that JumpStart clients can match. When a client finds an entry in `rules` that it matches, it uses the profile associated with that entry. Clients use only the first entry in the `rules` file that they match.

Each entry in the `rules` file is known as a *rule*. Profile files contain the configuration information JumpStart clients use to load the Solaris OE.

If a JumpStart client checks all the entries in `rules` but does not find a match, the client begins an interactive configuration session.

The `rules` File Syntax

Entries in the `rules` file conform to the following syntax:

```
[!] match_key match_value [&& [!] match_key match_value]* \
begin profile finish
```

where:

match_key	A predefined keyword that describes an attribute of the system being installed. The keyword can be: any, `hostname`, `model`, `arch`, `installed`, `network`, `domainname`, `karch`, `totaldisk`, `disksize`, or `memsize`.
match_value	The value (or range of values) selected by the system administrator for the *match_key*. You can use multiple keywords in a rule. Join multiple keywords with the logical AND symbol, (&&). You can use the logical NOT symbol (!) in front of a keyword to express negation. In other words, to express that the install client's value for *match_key* does not equal the *match_value* specified in the rule.
begin	The name of the begin script. Use a dash (-) to indicate that no begin script will be run.
profile	The name of the profile file.
finish	The name of the finish script. Use a dash (-) to indicate that no finish script will be run.

Examples of rules File Entries

The following is an example of the rules file entries.

```
#
# The first five rules listed here demonstrate specifics:
#
hostname   client1     -     host_class     set_root_pw
hostname   client2   -   class_basic_user     -
network   192.43.34.0  &&  ! model 'SUNW,Ultra-5_10'  -   class_net3   -
model    'SUNW,Ultra-5_10'   -   class_ultra    complete_ultra
memsize   64-96   &&   arch   sparc   -   class_prog_user   -
#
# The following rule matches any system.
any   -    -   class_generic   -
```

In this rules file example:

- The first rule matches a machine on a network called client1. The class file is host_class. The finish script is set_root_pw.
- The second rule matches a machine with host name client2. The class file is class_basic_user.
- The third rule matches a machine on network 192.43.34 that is not an Ultra™ 5 or 10 system architecture. The class file is class_net3. This rule does not specify begin or finish script.
- The fourth rule matches a machine that is an Ultra 5 or 10 system architecture. The class file is class_ultra. There is a finish script called complete_ultra.
- The fifth rule matches a machine using SPARC architecture and with a memory between 64 and 96 Mbytes. The class file is class_prog_user.
- The sixth rule matches any machine. The class file is class_generic. This rule does not specify a begin or finish script.

Begin Scripts

Begin scripts are Bourne scripts that JumpStart clients run *before* installing the Solaris OE. Begin scripts allow you to perform a variety of tasks on the JumpStart client. Typically, you would use a begin script to back up data from the client before proceeding with the Solaris OE installation.

The following example begin script causes the JumpStart client to copy its existing `/etc/passwd` and `/etc/shadow` files to a directory on an NFS server:

```
#!/bin/sh

HOSTNAME=`/bin/uname -n`
mount 192.10.10.100:/backup /mnt

if [ ! -d /mnt/${HOSTNAME} ]; then
       mkdir /mnt/${HOSTNAME}
fi

if [ -d /mnt/${HOSTNAME} ]; then
       mount /dev/dsk/c0t0d0s0 /a
       cp /a/etc/passwd /a/etc/shadow /mnt/${HOSTNAME}
       umount /a
fi

umount /mnt
```

This example script works only if the following conditions exist:

- The server using the IP address `192.10.10.100` shares the `/backup` directory in read-write mode and with the `anon=0` option set
- The JumpStart client has a previously installed `root` file system available as `/dev/dsk/c0t0d0s0`

This example script shows that a begin script can mount disk resources from other systems, mount resources from the client itself, and copy files between those mounted directories. File systems that exist on the client are available using their standard logical device names. NFS provides access to shared directories on the network. The mount points `/a` and `/mnt` are available in the `root` file system when the JumpStart client mounts from the boot server.

For a client to use a begin script, the script must be associated with a rule that the client selects from the `rules` file. For example, the rule:

```
hostname client1 begin1 config1 -
```

would cause a JumpStart client called `client1` to use the begin script called `begin1`.

Profile (Class) File

A profile file is a text file that determines how the Solaris OE installation will proceed on a JumpStart client. Profile files are sometimes called *class* files. Rules listed in the `rules` file allow clients to select an appropriate profile file. Although you usually associate a different profile with every rule, you can use the same profile for multiple rules.

The following example shows that for a client to use a profile file, the profile must be associated with the rule the client selects from the `rules` file:

```
hostname client1 - config1 -
```

The `rule` file would cause a JumpStart client called `client1` to use the profile file called `config1`.

An entry in a profile file consists of one keyword and its associated parameters. Each keyword controls one element of the Solaris OE software installation. Each profile consists of multiple entries. Profile file names must match the names used in the `rules` file.

Keywords and Arguments

Table 17-7 lists the keywords and parameters used in a profile file to specify how the Solaris OE installation proceeds on the JumpStart client.

Table 17-7 Keywords and Arguments for Profile Files

Keywords	**Arguments**
`install_type`	`initial_install | upgrade`
`system_type`	`standalone | dataless | server`
`partitioning`	`default | existing | explicit`
`cluster` *`cluster_name`*	`add | delete`

Table 17-7 Keywords and Arguments for Profile Files (Continued)

Keywords	Arguments
package *package_name*	add | delete
usedisk	*disk_name*
dontuse	*disk_name*
locale	*locale_name*
num_clients	*number*
client_swap	*size*
client_arch	*kernel_architecture*
filesys	*device size file_system optional_parameters*

The `cluster` keyword requires a parameter that lists name of the configuration cluster you want to install. Table 17-8 defines the configuration cluster names according to the common names used for them during the interactive installation routine.

Table 17-8 Possible Entries for the `cluster` Keyword

Interactive Installation Name	Configuration Cluster Name
Core	SUNWCreq
End User	SUNWCuser
Developer	SUNWCprog
Entire Distribution	SUNWCall
Entire Distribution Plus OEM Support	SUNWCXall

Appendix A of the *Solaris™ 9 System Installation and Configuration Guide* contains a description of the clusters and packages available on the Solaris 9 Software Distribution CD-ROM.

Examples of Profile Files

The following example describes a profile file that uses default partitioning, except that the swap partition size set to 128 Mbytes. The client installs the developer configuration cluster (SUNWCprog) and adds the NIS packages, SUNWypr and SUNWypu. The manual pages from this cluster (SUNWman) are deleted because the client mounts them from the server named server1.

```
# Select software for programmers
install_type       initial_install
system_type        standalone
partitioning       default
filesys            any   128 swap  # specify size of swap
filesys            server1:/usr/share/man - /usr/share/man ro,soft
cluster            SUNWCprog
package            SUNWman  delete
package            SUNWypr  add
package            SUNWypu  add
```

The following example describes a profile file that installs the Entire Distribution configuration cluster (SUNWCall), and removes the SUNWman package. The example uses explicit partitioning and declares the slices and sizes assigned to the /, swap, /usr, /var, and /opt file systems.

```
install_type    initial_install
system_type     standalone
partitioning    explicit
filesys         c0t0d0s0  150   /
filesys         c0t0d0s1  128   swap
filesys         c0t0d0s6  800   /usr
filesys         c0t0d0s7  free  /var
filesys         c0t1d0s7  all   /opt
cluster         SUNWCall
package         SUNWman   delete
```

Finish Scripts

Finish scripts are Bourne scripts that JumpStart clients run *after* installing the Solaris OE but *before* they reboot. Finish scripts allow you to perform a variety of post-installation tasks on the JumpStart client, including:

- Setting the power-management configuration
- Retrieving backed-up data from a server on the network
- Copying selected files from a JumpStart server to the client

The following example finish script causes the JumpStart client to turn off automatic shutdown for power management, retrieve its backed-up `/etc/passwd` and `/etc/shadow` files from a directory on an NFS server, and copy a file from the configuration server to the JumpStart client.

```
#!/bin/sh

touch /a/noautoshutdown

HOSTNAME=`/bin/uname -n`

mount 192.10.10.100:/backup /mnt

if [ -d /mnt/${HOSTNAME} ]; then
        echo "Copying passwd and shadow..."
        cp /mnt/${HOSTNAME}/passwd /a/etc/passwd
        cp /mnt/${HOSTNAME}/shadow /a/etc/shadow
fi

umount /mnt

mkdir /a/labfiles
cp ${SI_CONFIG_DIR}/files/SA118_setup.tar /a/labfiles
```

This example script works if the following conditions exist:

- The server using the IP address `192.10.10.100` shares the `/backup` directory.
- The `passwd` and `shadow` files exist in the `/backup/`*`client_name`* directory on the server that shares it, where *`client_name`* is the host name of the JumpStart client.
- The configuration server has the file called `SA118_setup.tar` in the `files` directory. The `files` directory must exist in the directory that this server shares, and the client uses it as `${SI_CONFIG_DIR}`.

Typically `${SI_CONFIG_DIR}` refers to the `/export/config` directory on the configuration server. `${SI_CONFIG_DIR}` specifically refers to the directory associated with the `install_config` item that the client found in the `/etc/bootparams` directory. The `${SI_CONFIG_DIR}` variable is one of several JumpStart software-specific variables that you can use in begin and finish scripts.

Note – For more information on JumpStart software variables available for use in begin and finish scripts, refer to the *Solaris 9 OE Advanced Installation Guide*.

In the Solaris 9 OE and earlier releases back to Solaris 2.5.1, JumpStart clients automatically mount all of their file systems below the `/a` directory, before the finish script runs. The client uses its boot image to construct the directory that it will use on reboot. The directory hierarchy is mounted under the `/a` directory in the boot image. This temporary mount point allows finish scripts to make changes to the client's directory hierarchy by prefixing the absolute path name of the files and directories to be modified, created, or deleted with the `/a`. This directory allows you to write finish scripts that copy files into the client's file systems without mounting them within the script.

The `touch /a/noautoshutdown` command is the only method available to automatically disable the power management feature on the JumpStart client. Without this file in the client's root (`/`) directory, the client asks power management configuration questions when it boots.

For a client to use a finish script, the script must be associated with the rule that the client selects from the `rules` file. For example, consider the rule:

```
hostname client1 begin1 config1 finish1
```

This rule would cause a JumpStart client called `client1` to use the finish script called `finish1`.

Setting Up Installation Service Alternatives

In addition to the standard JumpStart installation configurations, you can create alternatives for installation.

Using CD and DVD Sources

You can set up boot and installation services directly from the Solaris 9 Software 1 of 2 CD-ROM or from the Solaris 9 Software DVD. To do this, you must also configure identification and configuration services in the same manner as when you use a spooled Solaris OE image.

The installation image found on the Solaris 9 Software 1 of 2 CD-ROM only supports installing the Core and End User configuration clusters. The Solaris 9 Software 2 of 2 CD-ROM contains the remainder of the installation image, but there is no support for changing CD-ROMs in the middle of a JumpStart installation procedure.

The Solaris 9 DVD contains an installation image that supports installing all configuration clusters through the Entire Distribution with OEM support.

To set up boot and installation services from CD-ROM or DVD, complete the following steps:

1. Insert the Solaris 9 Software 1 of 2 CD-ROM in the CD-ROM drive or the Solaris 9 Software DVD in the DVD drive. Allow the `vold` daemon to automatically mount the media.
2. Change the directory to the location of the `add_install_client` script.

```
# cd /cdrom/cdrom0/s0/Solaris_9/Tools
```

3. Run the `add_install_client` script, and specify the server and client information as follows:

```
# ./add_install_client -c server:/config_path -p server:/sysid_path
client_name platform_group
```

   a. For the *server*`:`*/config_path* value, enter the name of the server and path where the `rules` and profile files are located.

   b. For the *server*`:`*/sysid_path* value, enter the name of the server and path where the `sysidcfg` file is located.

c. For the *client_name* field, enter the name of the JumpStart client.

d. For the *platform_group* field, enter the correct `kernel` architecture for the JumpStart client, for example, `sun4u`.

The `add_install_client` script automatically makes the changes required to support RARP, TFTP, and `bootparams` file and NFS requests from the client, but this script only causes the server to share the `/cdrom/sol_9_sparc/s0` directory. Sharing the `/cdrom/sol_9_sparc/s0` directory lets the JumpStart client to mount a root (/) file system during the network boot process and to gain access to the installation image.

You must manually configure the appropriate servers to share the other directories you name in the `add_install_client` command.

Using the `modify_install_server` Script

The `modify_install_server` script, located on the Solaris 9 Software Installation CD-ROM, enables an interactive Solaris Web Start™ style of installation on the client. The `modify_install_server` script replaces the JumpStart boot image in the directory you specify with a Web Start boot image.

Caution – Running the `modify_install_server` script actually defeats the purpose of custom JumpStart procedure. It disables the noninteractive benefit of the JumpStart procedure. The resulting installation process will be *interactive*.

To use the `modify_install_server` script to spool a Web Start boot image for JumpStart clients, complete the following steps:

1. Insert the Solaris 9 Software Installation CD-ROM into the CD-ROM drive. Allow the `vold` daemon to automatically mount it.
2. Change directories to the location of the `modify_install_server` script.

```
# cd /cdrom/cdrom0/s0
```

3. Run the `modify_install_server` script, specify the location of the Solaris 9 OE boot image, and specify the slice on the CD-ROM that holds the Web Start boot image.

```
# ./modify_install_server /export/install ../s1
```

Using a Flash Source

You can also use a Flash source as an alternative installation service. The Web Start™ Flash (Flash) installation feature lets you to create a single reference installation of the Solaris 9 OE on a master system. You can replicate the installation on other systems known as clones.

The Flash installation utilities are available starting with Solaris 8 OE update 4, and are installed as part of the Solaris OE. Before the Flash archive is created and deployed, you must decide how to integrate the installation process into your specific environment. Some items to consider are:

- Building support for custom hardware and driver configurations at installation time, which eliminates the need to re-create the archive in the future. The recommended installation for the required level of support on the master is Entire Distribution + OEM support.
- Selecting the name conventions for each archive in advance.
- Allocating the contents of each archive or customized multiple archives, including third-party software and package additions or deletions. At least one archive must contain the Solaris 9 OE files.
- Using the Web Start Flash archive.

Troubleshooting the JumpStart Procedure

If any of the four main JumpStart services are improperly configured, the JumpStart clients can:

- Fail to boot
- Fail to find a Solaris OE image to load
- Ask questions interactively for configuration
- Fail to partition disks or create file systems, and fail to load the OE

Resolving Boot Problems

Problems in the JumpStart client boot process are usually associated with RARP, TFTP, or `bootparams` file-related configuration issues. If the client issues error messages or fails to proceed with the boot process, it usually means that one of these services is not properly configured.

Resolving RARP Problems

If the JumpStart client fails to boot and repeatedly issues the following message:

```
Timeout waiting for ARP/RARP packet
```

then the JumpStart client cannot obtain RARP services from a boot server. This message indicates that the `/etc/ethers` or `/etc/inet/hosts` file on the boot server is not correctly configured. To correct this problem, edit these files, and ensure that the MAC address and host name for the client in the `/etc/ethers` file, and that the IP address and host name for the client in the `/etc/inet/hosts` file are correct.

Other problems to check for that can cause this error message:

- Name service not updated to reflect new entries in the `/etc/ethers` or `/etc/inet/hosts` files
- Physical network connections

Enter the commands required to update the name service in use. Usually, the messages these commands issue will indicate whether an update for the `/etc/ethers` or `/etc/inet/hosts` files was successful.

Check all of the physical network connections between the client and the boot server to eliminate a potential source of the updating problem.

Resolving TFTP Problems

If the JumpStart client issues the following message once and stops booting:

```
Timeout waiting for ARP/RARP packet
```

this message indicates that the JumpStart server cannot obtain TFTP services from a boot server.

Usually, this error message indicates that there is no entry for the JumpStart client in the `/tftpboot` directory on the boot server. An easy way to solve this problem is to run the `add_install_client` script for this client. For example:

```
# cd /export/install/Solaris_9/Tools
# ./add_install_client -c server1:/export/config -p
server1:/export/config client1 sun4u
```

Other problems to check for that can cause this message to appear:

- The incorrect platform group argument to the `add_install_client` script was used (For example, specifying `sun4m` for a `sun4u` system).
- The boot server is not configured to allow the `in.tftpd` daemon to run on demand.

If you specify the incorrect platform group for the client when you run the `add_install_client` script, the client might hang, or issue additional error messages and panic early in the boot process. To solve this problem, run the `add_install_client` script, and specify the correct platform group.

If the boot server is not configured to allow the `in.tftpd` daemon to run on demand, the client hangs. Usually, the `add_install_client` script automatically modifies the boot server to provide this service. To correct this problem, edit the `/etc/inetd.conf` file on the boot server, and remove the comment (#) character from the following line:

```
#tftp   dgram   udp6    wait    root    /usr/sbin/in.tftpd      in.tftpd
-s /tftpboot
```

After making this change, send a HUP signal to the inetd process:

```
# pkill -HUP inetd
```

Running the inetd daemon allows the client to resolve TFTP requests properly. The inetd daemon starts the in.tftpd daemon on demand, so usually you would not see the in.tftpd process in the list of running processes.

Resolving bootparams File Problems

If the JumpStart client obtains RARP and TFTP responses, but stops booting after displaying a numeric value, such as:

```
23e00
```

the JumpStart client is unable to obtain bootparams file information from a boot server. This value indicates that the client was able to load its network bootstrap program. If no information for the client exists in /etc/bootparams, or if the rpc.bootparamd daemon is not running, this portion of the boot process will fail.

If no entry exists in the /etc/bootparams file for the JumpStart client, create an entry by running the add_install_client script that automatically starts the rpc.bootparamd daemon.

The /etc/rc2.d/S27boot.server script starts the rpc.bootparamd daemon when the boot server boots. Logic in the /etc/rc2.d/S27boot.server script checks for the /tftpboot directory, and starts the rpc.bootparamd daemon if the directory exists. Check if the rpc.bootparamd daemon is running:

```
# pgrep -l bootparamd
```

If the rpc.bootparamd process is not running, check whether the /tftpboot directory exists. If it exists, manually start the rpc.bootparamd process with the following script:

```
# /etc/init.d/boot.server start
```

Resolving Identification Problems

Problems in the JumpStart client identification process usually relate to identification information missing from the `sysidcfg` file or from a name service. If a JumpStart client cannot obtain a response from a server for any identification item, the client interrupts the automatic identification process and asks for the information. The client usually indicates what information is missing, but not necessarily from what source.

Resolving `sysidcfg` Problems

In the absence of a name service, if the JumpStart client interrupts the identification or installation process to obtain any of the following identification items, check the `sysidcfg` file on the JumpStart server, and correct the problem you find:

- Will the client be configured to use IPv6 networking?
- What netmask will the client use?
- What is the IP address of the default router?
- What security policy will the client implement?
- What name service will the client use?
- What time zone will the client use?
- What system locale will the client use?
- What system will provide the time-of-day information?
- What is the `root` log in password?

Resolving Name Service Problems

If you use a name service, and the JumpStart client interrupts the identification process to obtain identification items *other than* the following, check the corresponding map or table information in the name service, and correct the problem you find:

- Will the client implement IPv6 protocols?
- What is the IP address of the default router?
- What security policy will the client implement?
- What is the `root` log in password?

The previous items can only be provided using the `sysidcfg` file.

You can use the `sysidcfg` file to provide information that a name service could otherwise provide. You must verify the content of the `sysidcfg` file or any information that it provides. Information provided in the `sysidcfg` file overrides information in name services.

Resolving Configuration Problems

Problems in the JumpStart client configuration process usually relate to improperly configured `rules` or profile files. If a JumpStart client cannot obtain a response from a server for any configuration item, or if the configuration information it finds is incompatible with the client's hardware, it interrupts the automatic configuration process.

The information that the client requests usually indicates what is missing or improperly configured. Incompatible configuration information causes the client to display a panel that describes the problem.

Resolving `rules` File Problems

Sometimes the JumpStart client completes its identification tasks, but then issues the following messages:

```
Checking rules.ok file...
Warning: Could not find matching rule in rules.ok
Press the return key for an interactive Solaris install program...
```

These messages indicate that it cannot find an entry in the `rules.ok` file that it matches.

Usually this happens because administrators fail to run the `check` script to generate an up-to-date `rules.ok` file. To correct this problem, verify that the `rules.ok` file contains an entry that will match the client, and then run the `check` script. For example:

```
# ./check
Checking validity of rules...
Checking validity of profile1 file...
The auto-install configuration is ok.
#
```

Resolving Profile (Class) File Problems

If the JumpStart client completes its identification tasks, but then displays an error message, such as:

```
ERROR: Field 2 - Disk is not valid on this system (c0t4d0s0)
```

it indicates that a configuration error exists in the profile file it has selected.

To correct this error, edit the profile file that the client uses, and correct the problem indicated.

Resolving Installation Problems

Problems in the JumpStart client installation process usually relate to NFS configuration problems. If a server fails to share a directory that a JumpStart client requires, the installation cannot proceed.

Resolving NFS Problems

If the JumpStart client obtains RARP and TFTP responses, but panics and displays an error message similar to the following:

```
panic - boot: Could not mount filesystem
Program terminated
ok
```

the client cannot mount the root (/) file system defined in the `/etc/bootparams` file.

To correct this problem, edit the `/etc/dfs/dfstab` file on the boot server to ensure that it contains an entry that shares the required directory structure. Check the `/etc/bootparams` file on the boot server to determine what directory to share. For example, the `/etc/dfs/dfstab` file could contain the following entry to share the `/export/install` directory:

```
share -F nfs -o ro,anon=0 /export/install
```

The `-o ro,anon=0` options are required for the client to use the root (/) file system properly.

Run the following script to stop and start the NFS daemons on the boot server:

```
# /etc/init.d/nfs.server stop
# /etc/init.d/nfs.server start
```

If the JumpStart client issues an error message that indicates that it cannot mount any directory it requires or automatically begins an interactive installation session, verify the configuration of the `/etc/dfs/dfstab` file on the servers that provide the directories that the client requires. Make any required change in the servers'`/etc/dfs/dfstab` files, and stop and restart the NFS server daemons on those servers.

Any directory listed in the `/etc/bootparams` file on the boot server must be shared by the server providing the directory.

Resolving Begin and Finish Script Problems

Begin and finish script problems can be the most troublesome of all issues related to the JumpStart procedure. Any error possible in a shell script is possible in one of these. Debugging begin and finish scripts might involve multiple attempts at booting the JumpStart client, or otherwise performing trial runs of the scripts.

After writing begin or finish scripts, you must verify that these scripts are referenced in the appropriate rule in the `rules` file. You must also remember to run the `check` script to regenerate the `rules.ok` file.

Resolving Syntax Problems

If the JumpStart client boots, displays the GUI interface in one window, and then the window disappears after the begin script runs, a syntax error might exist in your begin script.

To check for this problem on the JumpStart client, open a terminal window, and examine the `/tmp/begin.log` file. This file contains standard output and error messages that the begin script generates. Correct any error it reports in the begin script and try booting the client again.

The JumpStart client behaves similarly when it encounters errors in finish scripts. If the JumpStart client abruptly closes the window in which the finish script is running, it is probable that a syntax error exists in your finish script.

To check for this problem, after the JumpStart client reboots, examine the `/var/sadm/system/logs/finish.log` file. This file contains standard output and error messages that the finish script generates. Correct any error it reports in the finish script, and try booting the client again.

Identifying Log Files

JumpStart clients retain the following log files during the installation process:

```
/tmp/begin.log
/tmp/finish.log
/tmp/install_log
/var/sadm/system/logs/sysidtool.log
```

These logs contain standard output and error messages from begin scripts, finish scripts, the Solaris OE software installation process, and the system identification process that the client performs.

JumpStart clients retain a corresponding set of log files after the installation process completes and the system reboots:

```
/var/sadm/system/logs/begin.log
/var/sadm/system/logs/finish.log
/var/sadm/system/logs/install_log
/var/sadm/system/logs/sysidtool.log
```

Module 18

Performing a Flash Installation

Objectives

The Solaris™ Web Start Flash (Flash) installation feature enables you to create a single reference installation of the Solaris™ Operating Environment (Solaris OE) on a system, which is called the master system. You can replicate this OE installation on a number of systems, called clone systems.

Upon completion of this module, you should be able to:

- Describe the Flash installation feature
- Manipulate a Flash archive
- Use a Flash archive for installation

The following course map shows how this module fits into the current instructional goal.

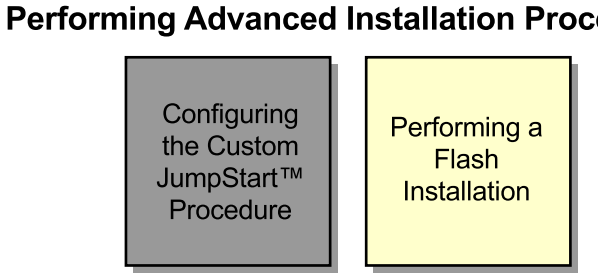


Figure 18-1 Course Map

Introducing the Flash Installation Feature

The Flash installation feature lets you create a single reference installation of the Solaris 9 OE on a master system, and then replicate the installation on other systems known as clones.

The Flash installation utilities are installed as part of the Solaris 9 OE. Before the Flash archive is created and deployed, you must decide how to integrate the installation process into your specific environment. Some items for consideration are:

- Including support for custom hardware and driver configurations at installation time, eliminating the need to re-create the archive in the future. The recommended installation for the required level of support on the master is Entire Distribution + OEM support.
- Selecting the naming conventions for each archive in advance.
- Deciding upon the contents of each archive or customized multiple archives, including third-party software and package additions or deletions. At least one archive must contain the Solaris 9 OE files.
- Using the Solaris Web Start Flash archive.

Note – The master and clone systems must be of like architectures, whether they use `sun4m`, `sun4u`, or Intel Architecture (IA).

Uses of the Flash Installation Feature

You can build multiple customized configurations on the master system by using packages from a predefined pool. Flash installation is significantly faster than the current JumpStart or Solaris Web Start network installation methods. Flash allows detailed customization of the Solaris OE, hardware configuration, and third-party software packages prior to the creation of the clones. In addition, Flash installation can act as an enterprise-level disaster recovery when necessary.

Flash Deployment Methods

The Flash installation process is integrated into the existing custom JumpStart software framework. The installation process is specified by keywords in the JumpStart profile on the JumpStart server during JumpStart setup. You can also deploy Flash using the Solaris Web Start installation method from the Solaris 9 OE installation CD, which requires an interactive installation. Flash archive extraction includes the copying of files from the archive to the clone. The Flash installation bypasses procedural scripts in the package-based JumpStart installation, making the process of building a clone machine extremely fast. Flash eliminates the need for finish scripts or for the customization of the JumpStart software image.

Flash Installation Process

Flash installation is a three-stage process involving:

- Creating and installing the master system
- Creating a Flash archive
- Deploying the Flash archive to the clone system

Installing the Master

The Flash installation feature uses one or more archives created from a master system that acts as a reference configuration. The master system is an installed system that has been customized as required. Customization can include adding or removing software packages, adding third-party or unbundled software products, and modifying configuration files, such as the run control scripts and the `/etc/inetd.conf` file.

Creating the Flash Archive

The Flash archive is identical to the current installation on the master system. You can easily transfer the archive as a large file from server to server to deploy it to the clone systems. To make managing multiple archives easier to manage, you can add identification information using the command line. You can create the archive when the system is running in single-user mode, multiuser mode, or being booted from the Solaris 9 OE Installation CD-ROM.

During installation you must specify a directory and a location where the Flash archive resides. Options during installation are:

- Network file system (NFS) server
- Hypertext Transfer Protocol (HTTP) server
- Local or remote tape
- Compact Disc (CD)
- Local drive of clone machine

Having a planning sheet serves as an important tool to help you make decisions and to document the archive creation and installation process. After you determine the content of the archive, you can proceed to the actual installation process.

Deploying the Flash Archive to the Clone

You can install the Flash archive on to the clone using an interactive install, the Solaris Web Start 3.0 installer, or custom JumpStart procedure. The interactive method requires you to boot the system to be cloned from the Solaris 9 OE Software CD-ROM 1 of 2. The Web Start installation requires the Solaris 9 OE Installation CD-ROM. To initiate the JumpStart procedure, the required JumpStart services must be configured on an appropriate server. The Flash archive is extracted on to the clone, replacing the package-based installation process.

Note – Although most files on the master system are configured before the archives are created, some network files might need re-configuration after being deployed to the clone systems.

Flash Installation Requirements

The following sections describe the Flash installation hardware and software requirements, dependencies, and limitations.

Hardware Requirements

The recommended system specifications for a Flash installation are:

- A SPARC system for the clone and UltraSPARC® system for the master system. Although minimum requirements for IA installation have not been listed, a Pentium processor or equivalent architecture is recommended.
- The master and the clone must have the same `kernel` architecture, such as `sun4u`, `sun4m`, or IA. The Flash installation supports both SPARC and IA platforms, although a SPARC master cannot be used for an IA clone installation, and an IA master cannot be used for a SPARC clone installation.
- Before you create the archive, you must install and configure the master with the exact software, hardware, and peripheral device package that you want on the clone. For example, to create a clone that uses an Elite3D framebuffer, (even if the master does not use the Elite3D card), you must include the necessary Solaris OE software support in the archive.

Software Requirements

The recommended software specifications for a Flash installation is:

The Flash utility comes with Solaris 9 OE and is installed as part of the Solaris OE. Flash utilities are also available with the minimum Solaris software group (`SUNWusr`). The Entire Distribution + OEM software group is recommended for you to be able to include all files and driver support when creating the Flash archive.

```
# more /var/sadm/system/admin/CLUSTER
CLUSTER=SUNWCXall
```

Limitations of the Flash Utility

There are certain limitations to the Flash utility, including, but not limited to, the configuration of the Solaris™ Volume Manager software and the current versions of the Solaris OE:

- Flash does not support metadevices or non-UFS file systems (except for installation for IA-type disks).
- You can only create the archive from material available on the master's original configuration. The Flash utility will not install archives of differing Solaris OE versions.

Manipulating a Flash Archive

The Flash installation process involves creation of the Flash archive prior to the deployment of the Flash archive to the clones.

Note – Ensure that the master is running as stable as possible during archive creation.

The Flash installation utility comprises two commands:

- You can use the `/usr/sbin/`*`flarcreate`* command to create an archive on the master.
- You can use the `/usr/sbin/flar` archive administration command to extract information from an archive, to split an archive, or to combine archives.

For additional information about the Flash archive process, view the online man pages.

The next section introduces the various Flash utility commands.

Create a Flash Archive

The syntax for the `flarcreate` command is:

```
flarcreate -n name [-R root] [-S] [-c] [-t] [-m master] [-a author]
[-e descr] [-x exclude] archive
```

where:

`-n`	Specify the name of the archive
`-R`	Specify the root of the Flash archive
`-S`	Do not include sizing information in the archive
`-c`	Compress the archive using the `compress` command
`-t`	Create an archive on a tape device
`-m`	Specify the name of the master on which you created the archive
`-a`	Specify the author of the archive

`-e`	Specify the description of the archive
`-x`	Exclude the named directory from the archive
`archive`	Specify the path to the Flash archive

The following example shows the creation of a Flash archive:

```
# flarcreate -n flash_root_archive -c -R / -e root_archive -x /flash \
-a admin_operator flash_archive1
Determining which filesystems will be included in the archive...
Determining the size of the archive...
The archive will be approximately 517.98MB.
Creating the archive...
2034098 blocks
Archive creation complete.
```

Performing Flash Archive Administration

You use the `/usr/sbin/flar` command to perform archive administration. The syntax for the `flar` command is:

```
flar -i archive

flar -c archive

flar -s archive
```

where:

- `i` — Retrieves information about archives that have been created
- `c` — Combines the individual sections that make up an existing archive into a new archive
- `s` — Splits an archive into one file for each section of the archive

Keywords exclusive to Flash and identification of the archive can be viewed from the online manual pages.

To list the header data that is created with the archive, use the `flar` command:

```
# flar -i flash_archive1
archive_id=12c58ec3286dc08ab07beda7339399c9
files_archived_method=cpio
creation_date=20020320202034
creation_master=sys44
content_name=flash_root_archive
creation_node=sys44
creation_hardware_class=sun4u
creation_platform=SUNW,Ultra-5_10
creation_processor=sparc
creation_release=5.9
creation_os_name=SunOS
creation_os_version=Generic
files_compressed_method=compress
files_archived_size=543105559
content_description=root_archive
content_author=admin_operator
content_architectures=sun4u
```

The header of the archive file contains the following identification parameters for the archive:

- `content_name` – The name of the archive (in this case, `flash_`*`directoryname`*`_archive`)
- `creation_date` – The date that the archive is created (from the master)
- `creation_master` – The name of the master (in this case, `sys44` and `instructor`)
- Other information about the archive

You can also use additional keywords for administering the archive.

Using a Flash Archive for Installation

The third and final stage of the Flash installation is the deployment of the archive onto the clone. This process can create multiple clones of the master.

You can use any of the Solaris OE installation methods to install Flash archives. This module describes the procedures to:

- Install Flash archives with the Solaris Web Start program
- Install Flash archives with the Solaris OE `suninstall` program
- Install Flash archives with a JumpStart installation

Using a Flash Archive With Solaris™ Web Start

If you are using the Solaris Web Start software (Solaris OE Installation CD), there is no need for a JumpStart configuration, but you must manually adjust the installation program for system identification and disk layout specifications.

To create a Flash archive with the Solaris Web Start software, complete the following steps:

1. Shut down the system to the OpenBoot™ PROM prompt. Insert the Solaris 9 OE Installation CD, and boot the CD-ROM.

```
ok boot cdrom
```

   The system searches for a location to store the installation software during the system boot. The installation software is stored on the swap partition, and you are presented with the Welcome screen to begin the Web Start installation process.

Note – If you followed the setup instructions for the this course, then the root disk `swap` partition will be allocated for this installation image. The `swap` partition is allocated to slice 1 of the system disk, but slice 1 begins on cylinder 0. If you did not put the `swap` partition on cylinder 0, you must correct the disk partitioning during the Flash archive installation procedure.

A Welcome window appears, as shown in Figure 18-2. The Welcome window lists the items to be configured using this installation procedure.

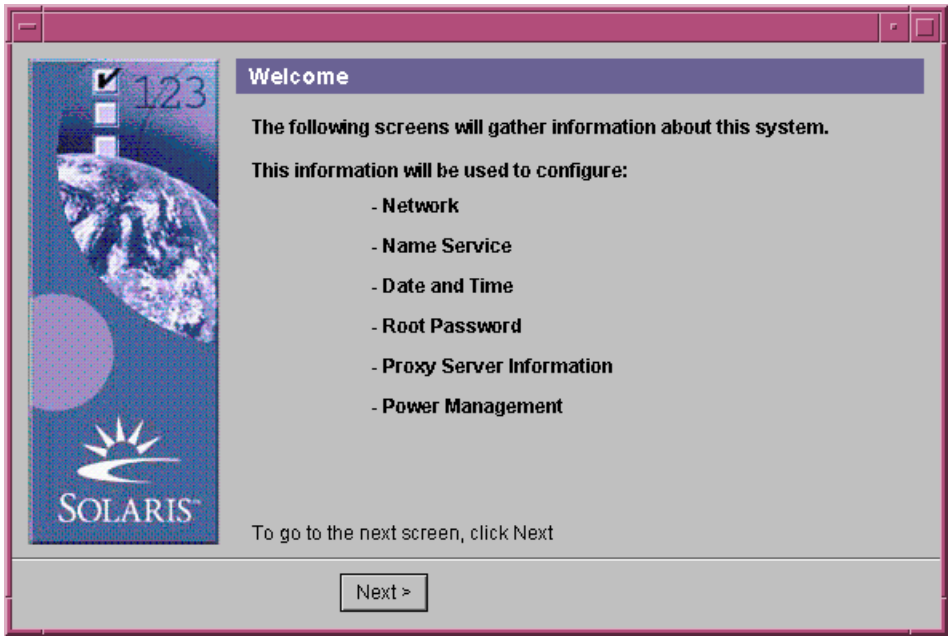


Figure 18-2 Welcome Window

2. Click Next to continue.

3. In the Network Connectivity window, as shown in Figure 18-3, select Networked.

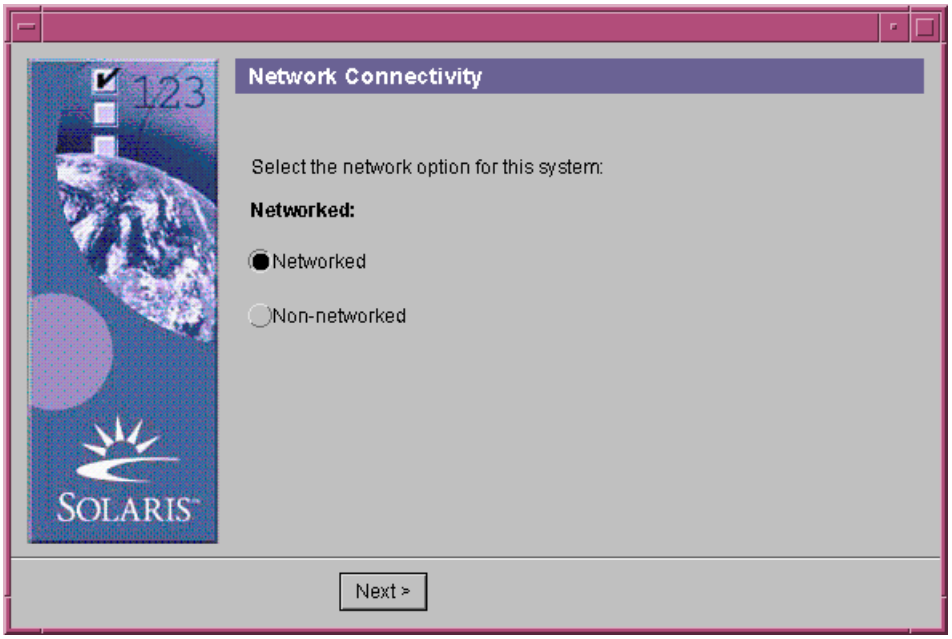


Figure 18-3 Network Connectivity Window

The system prompts you to answer a series of network configuration questions.

4. Click Next to continue.

Dynamic Host Configuration Protocol (DHCP) enables automatic host configuration.

5. The DHCP feature is not used in this lab; therefore, select the default selection of No in the DHCP window, as shown in Figure 18-4.

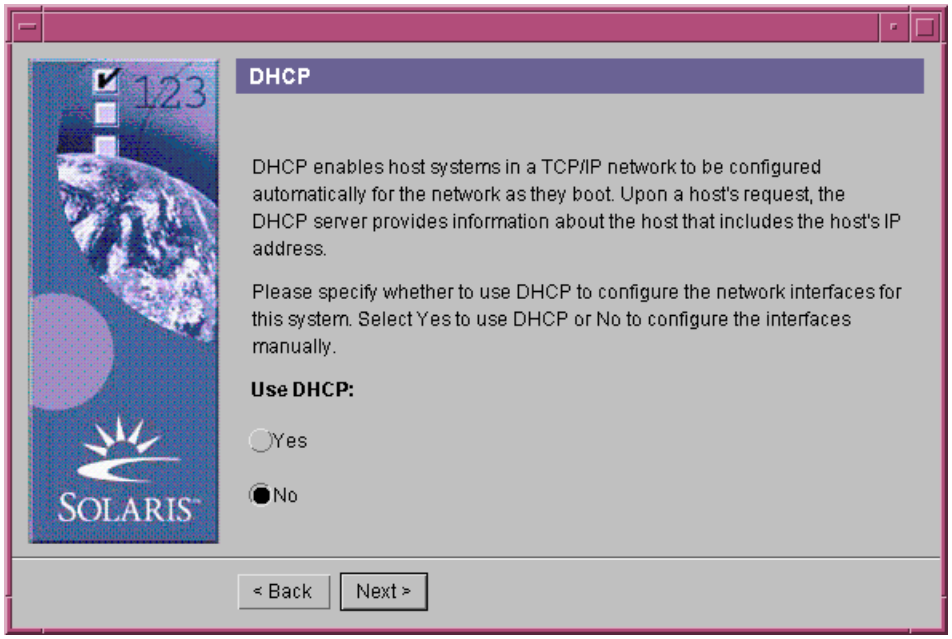


Figure 18-4 DHCP Window

6. Click Next to continue.

7. Enter the host name of the clone in the Host Name field in the Host Name window, as shown in Figure 18-5.

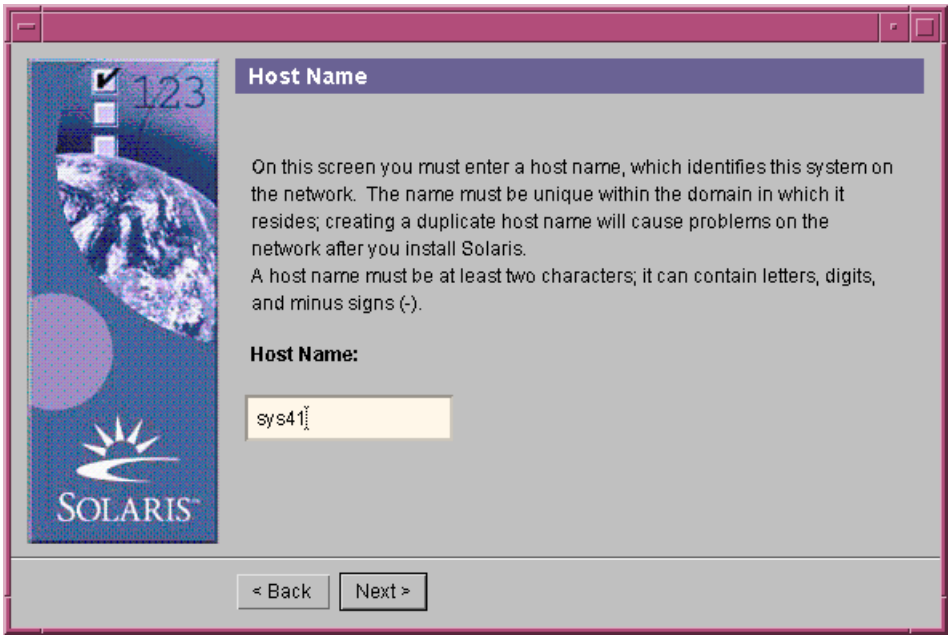


Figure 18-5 Host Name Window

8. Click Next to continue.

9. From the IP Address window, as shown in Figure 18-6, type the IP address in the IP address field to which this system will respond when it is configured.

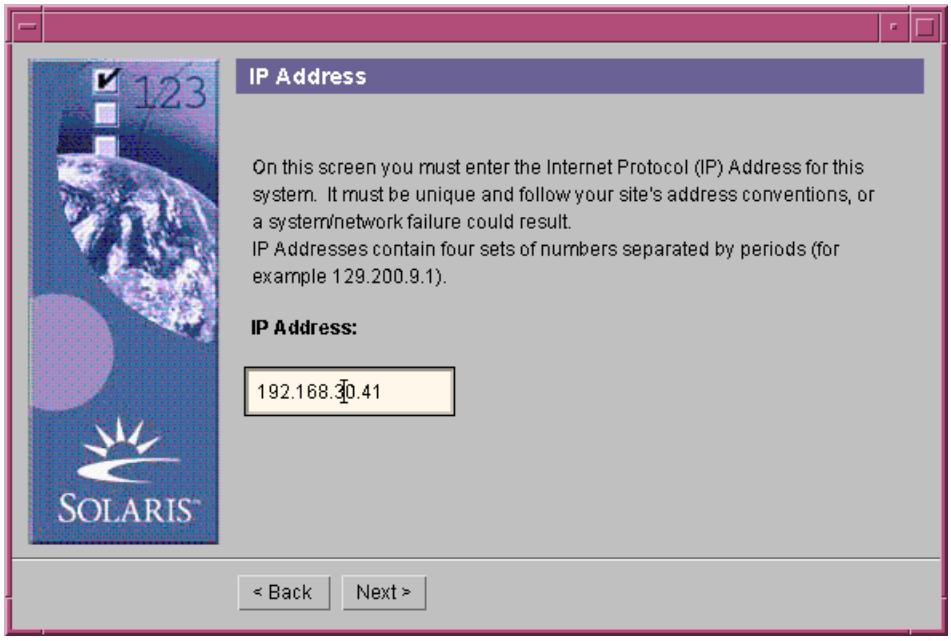


Figure 18-6 IP Address Window

10. Click Next to continue.

The Netmask Window window shows that the first three octets are used for the network and subnets address segments, and the right-most octet is reserved for the host address segment, as shown in Figure 18-7.

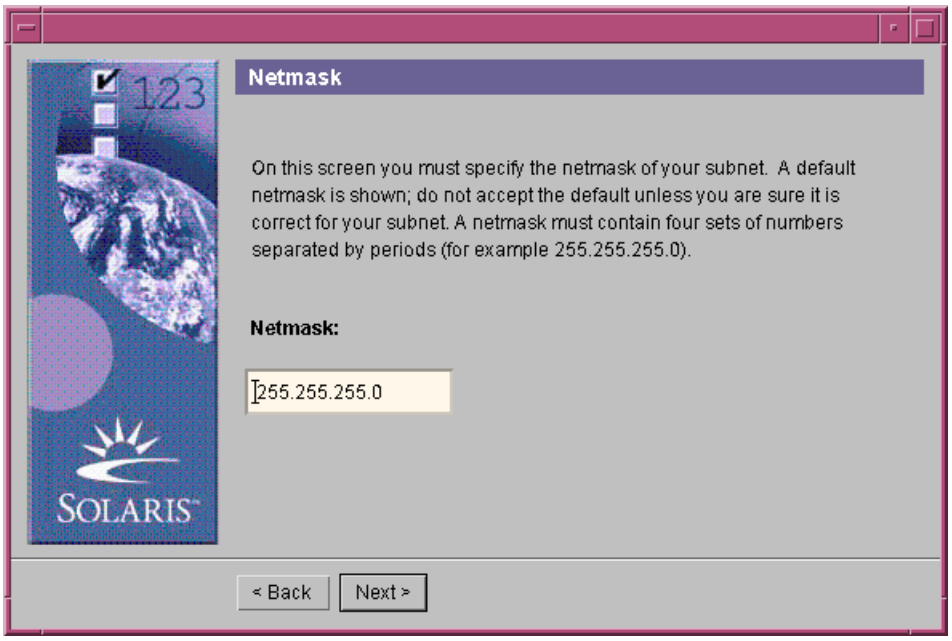


Figure 18-7 Netmask Window

11. Accept the default value. (The system's default configures IPv4.)
12. Click Next to continue.

To configure Internet Protocol version 6 (IPv6) during installation, you can select Yes in the IPv6 window (Figure 18-8). However, IPv6 is a topic for a network configuration course, so this selection is not explored in this course.

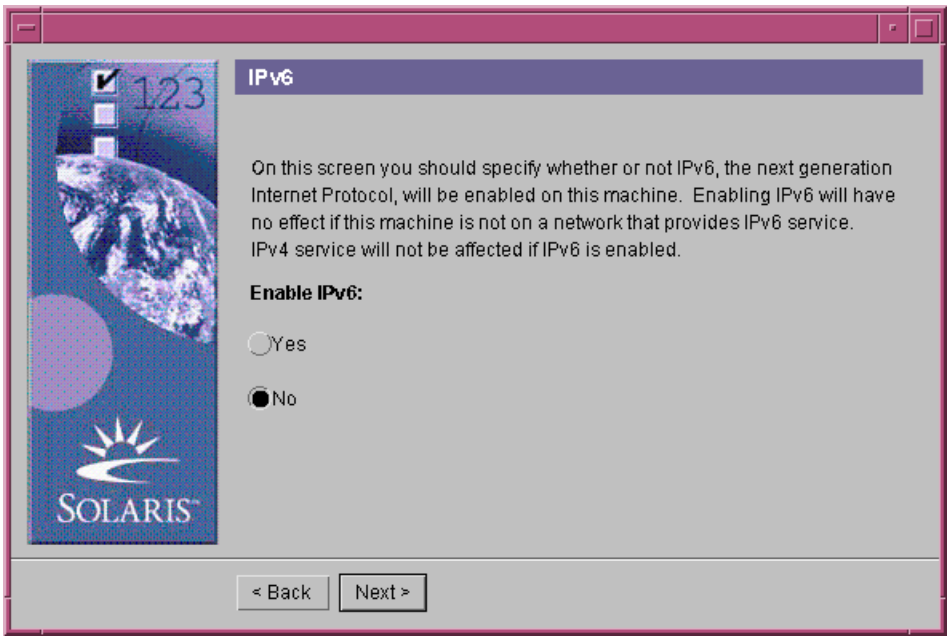


Figure 18-8 IPv6 Window

13. Select No.
14. Click Next to continue.

You use the Name Service window to select the name service for which you want to configure the system. If you select None, as shown in Figure 18-9, the system performs name resolution using the local files.

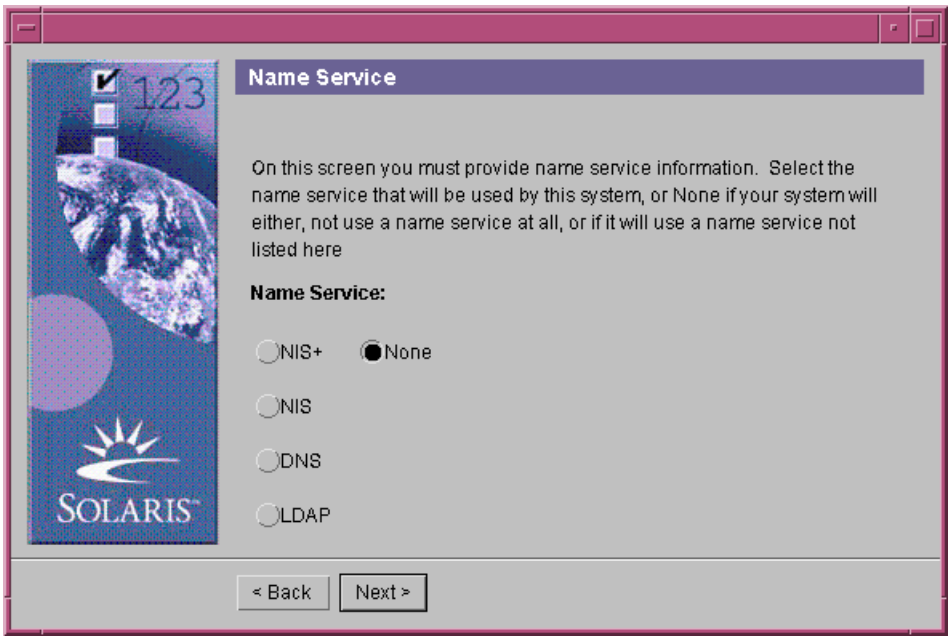


Figure 18-9 Name Service Window

15. Select None.
16. Click Next to continue.

You can let the system select a default router. You can also specify a router if you know the router address that you want to select, as shown the Default Router window in Figure 18-10.

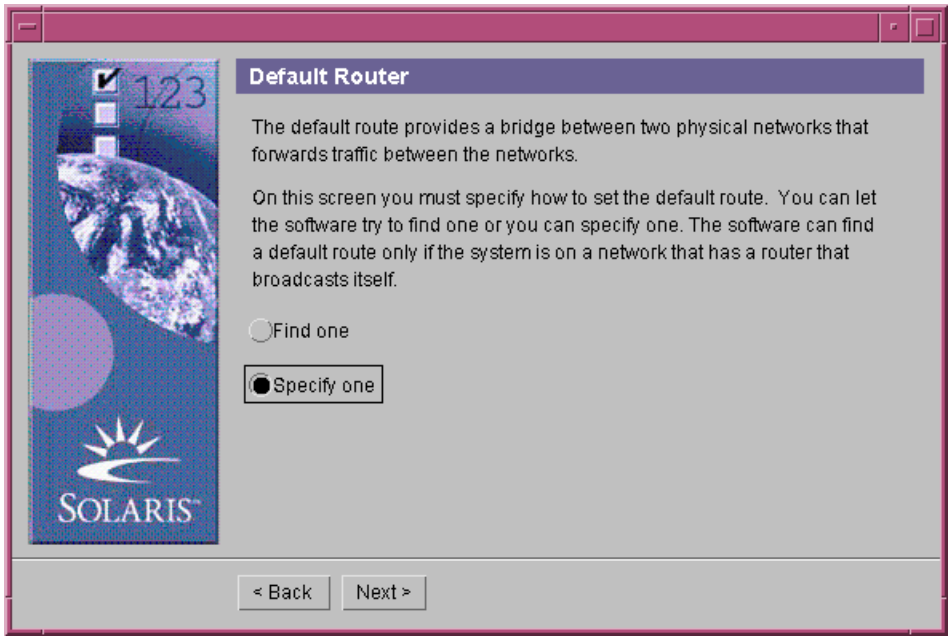


Figure 18-10 Default Router Window

17. Select Specify one to let the system assign a default router.
18. Click Next to continue.

19. Determine the default router address from other systems that are configured on the master system, and type this value in the Router IP Address field in the Default Router window, as shown in Figure 18-11.

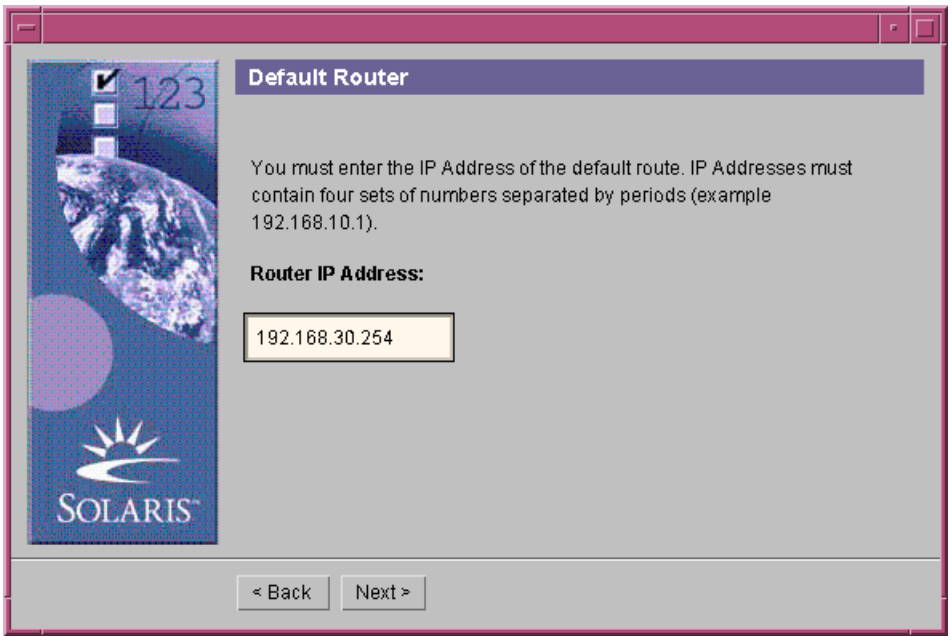


Figure 18-11 Default Router Window

20. Click Next to continue.

21. From the Time Zone window, shown in Figure 18-12, select the Geographic region to specify time zone information.

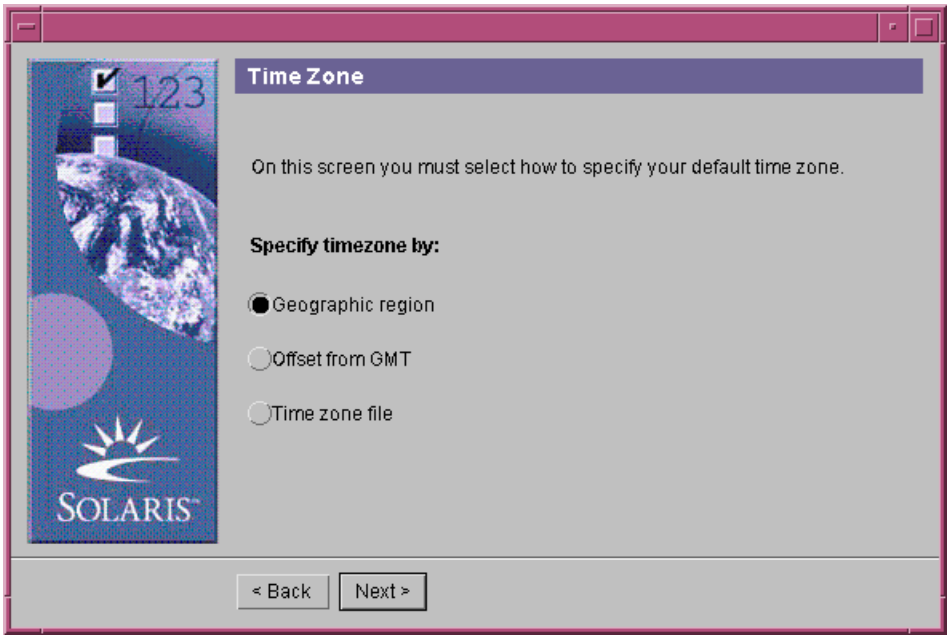


Figure 18-12 Time Zone Window

22. Click Next to continue.

The Geographic Region window displays the various geographic regions of the world, as shown in Figure 18-13. When you select a geographic region from the left column, its corresponding time zones appear in the right column.

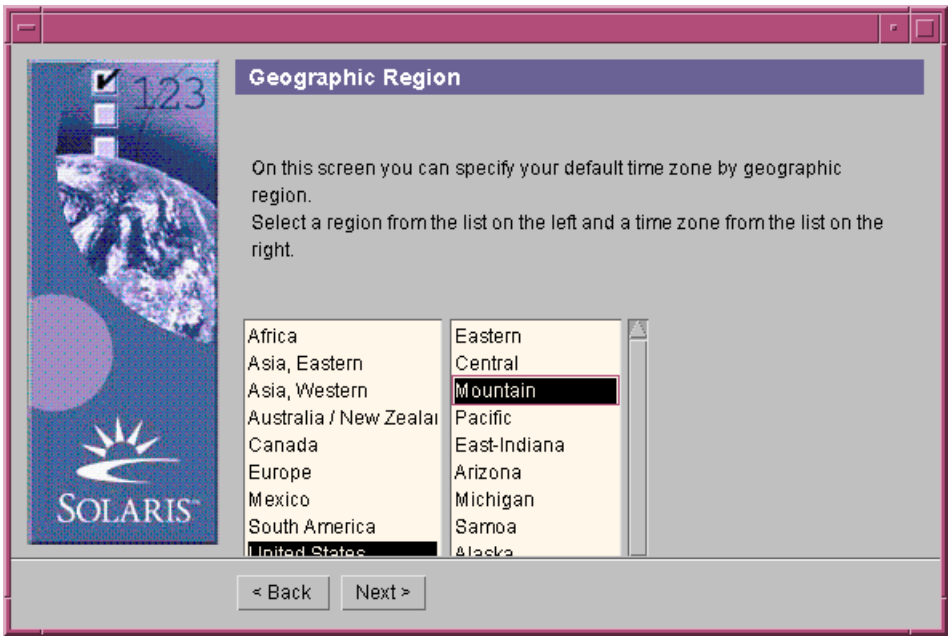


Figure 18-13 Geographic Region Window

23. Select your geographic region of the world.
24. Select your time zone within the specified geographic region.

    When you click a time zone, the time zone variable that appears in the right window is set to display your selected time zone.
25. Click Next to continue.

The Date and Time Window window, as shown in the Figure 18-14, displays the current time in the selected geographic region and time zone.

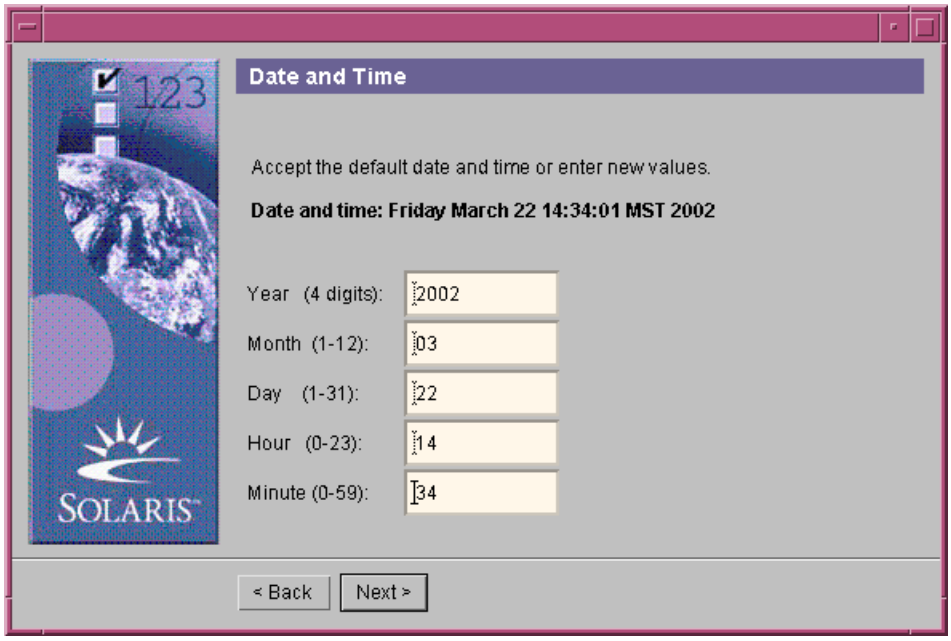


Figure 18-14 Date and Time Window

26. Change the current date and time, if necessary.
27. Click Next to continue.

28. To preset the root password, enter a password in the first blank field in the Root Password Window window, as shown in Figure 18-15. Confirm the password in the next blank field. If the password field currently contains an entry, delete it and then enter your new password to ensure root user access.

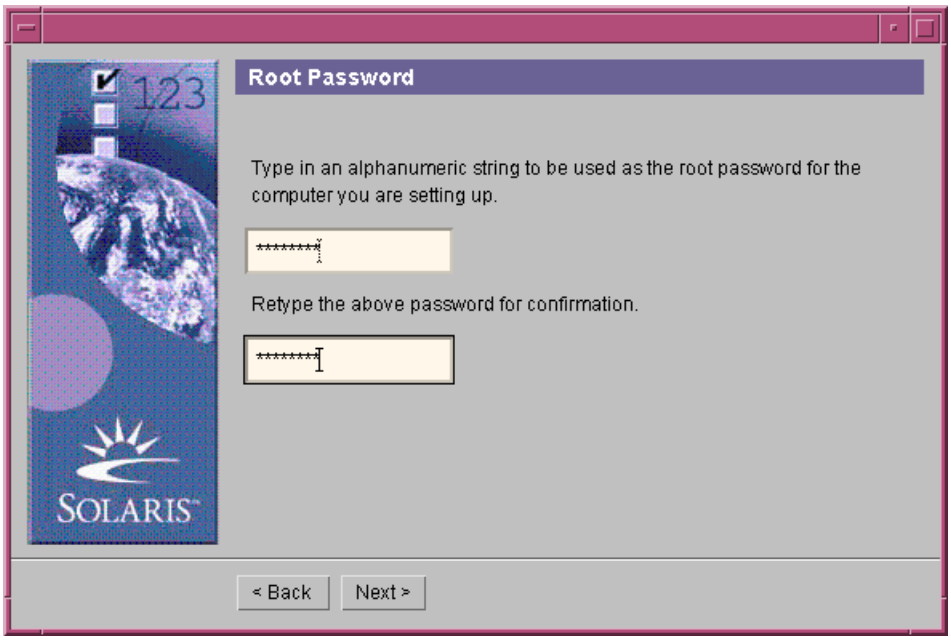


Figure 18-15 Root Password Window

29. Click Next to continue.

30. In the Power Management window, as shown in Figure 18-16, select Turn Power Management Off.

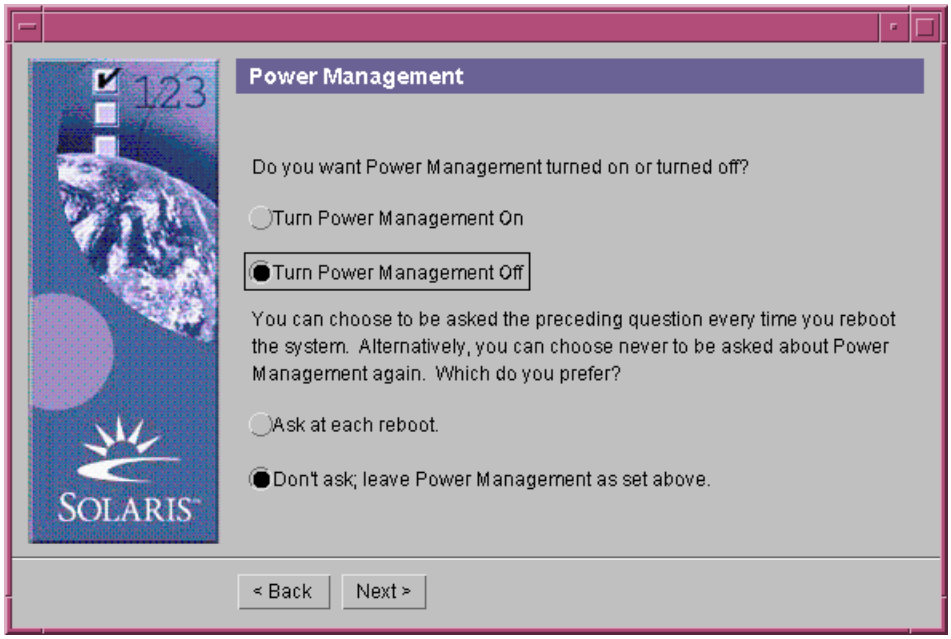


Figure 18-16 Power Management Window

The Power Management feature automatically shuts down the system after a predetermined period of system inactivity. To use this system as an application server, make sure that it does not shut itself down. Turning off the power management ensures that the system stays up through periods of low system activity. It also keeps the system up for remote logins.

31. Click Next to continue.

The Proxy Server Configuration window, as shown in Figure 18-17, shows that the proxy server provides an additional layer of security between the Internet and your system.

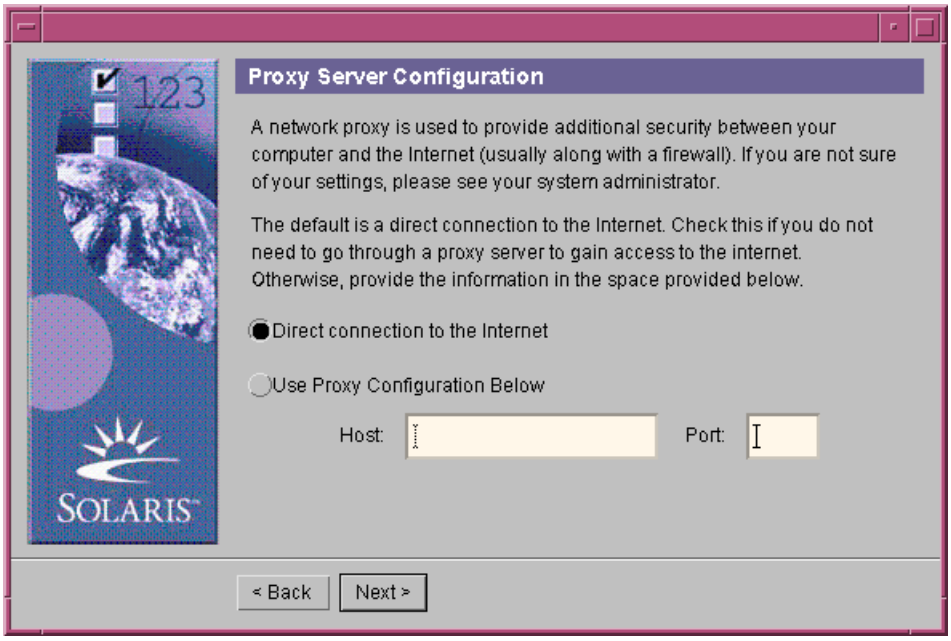


Figure 18-17 Proxy Server Configuration Window

32. Select Direct connection to the Internet for this installation procedure.
33. Click Next to continue.

The Confirm Information window, as shown in Figure 18-18, summarizes the selections you made during this entire procedure.

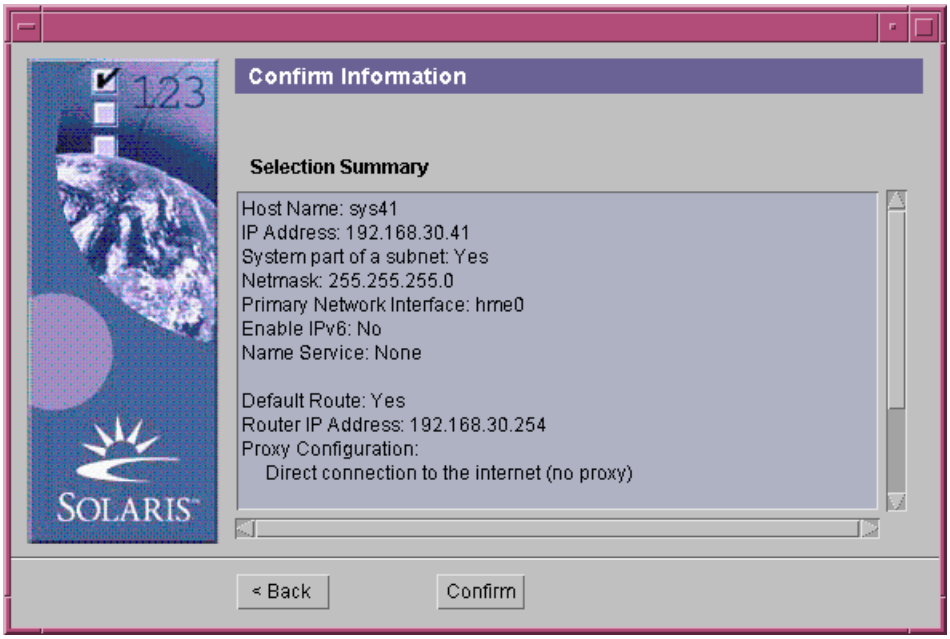


Figure 18-18 Confirm Information Window

34. After making your selections, click Confirm to accept your selections and continue, or click Back to retrace your steps and change a selection.
35. Click Next to continue.

After you confirm your selections, the system welcomes you to the first window of the Solaris Web Start process, as shown in Figure 18-19, which is the second phase of the process.

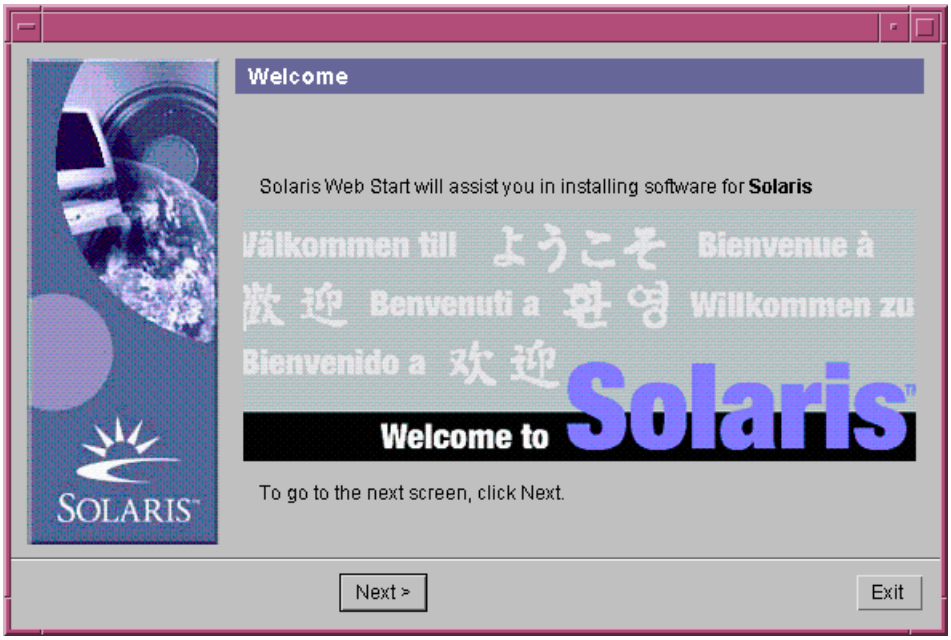


Figure 18-19 Welcome Window

36. Click Next to continue.

37. In the Specify Media Window window, select Network File System to specify the source of the Flash archive files, as shown in Figure 18-20.

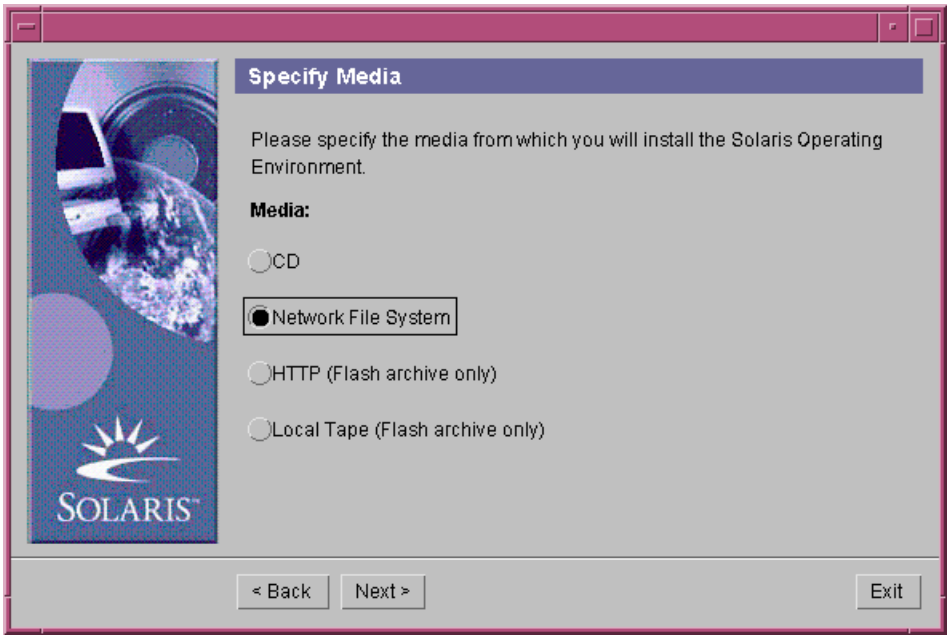


Figure 18-20 Specify Media Window

These files currently are located in a file system on the archive system, so that you can access them across an NFS file system.

Note – If the Flash archive directory on the server has been shared using NFS, the archives should be accessible.

38. Click Next to continue.

The Specify Network File System Path window, as shown in Figure 18-21, displays a list of sample Flash archive files.

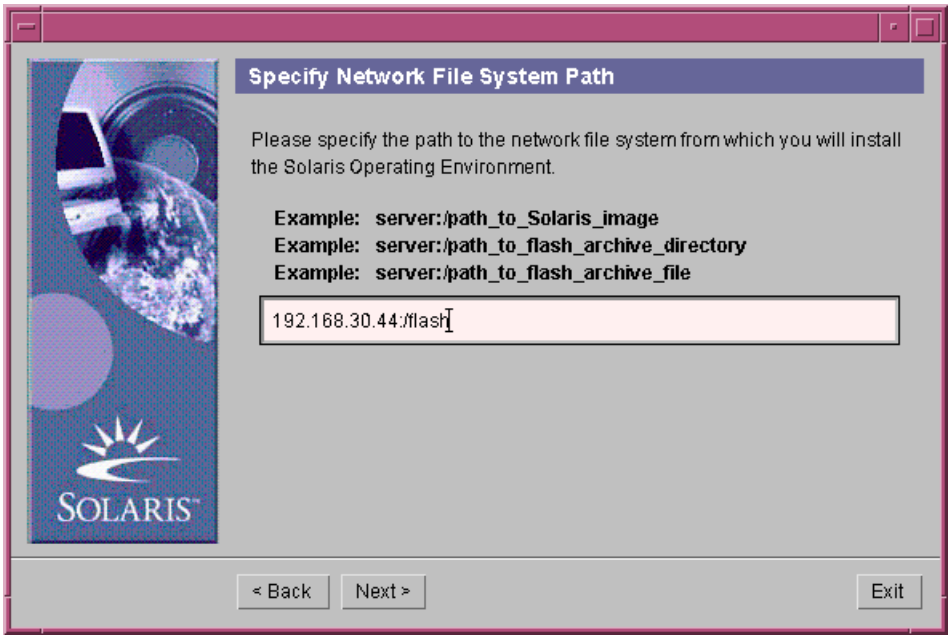


Figure 18-21 Specify Network File System Path Window

In the Specify Network File System Path window, you enter the name of the server storing the archive files in the blank field. If you have not configured the system for name resolution, use the IP address of the master server, followed by the absolute path name to the Flash archive directory.

For example, for a server named `sys44`, with an IP address of `192.168.30.44`, and an archive directory at `/flash`, the entry is:

```
192.168.30.44:/flash
```

39. Enter **`192.168.30.44:/flash`** in the entry field.
40. Click Next to continue.

The Select Flash Archives window, as shown in Figure 18-22, shows that when you select an archive, an information box appears containing more detailed information about the archive. You can append this information to the archive during archive creation by using various options to the `flarcreate` command. You can select any number of archives, depending on what you want to configure the system to do after the Flash archive installations are complete.

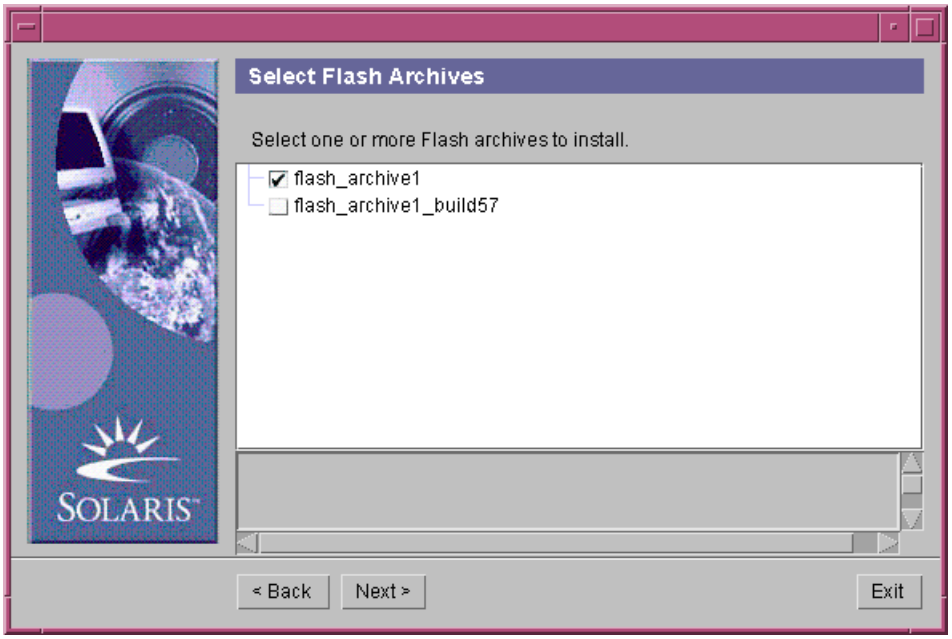


Figure 18-22 Select Flash Archives Window

41. Select the box for the appropriate archive to continue.
42. Click Next to continue.

The Flash Archive Summary window, as shown in Figure 18-23, displays the type of Flash archive and the archive location.

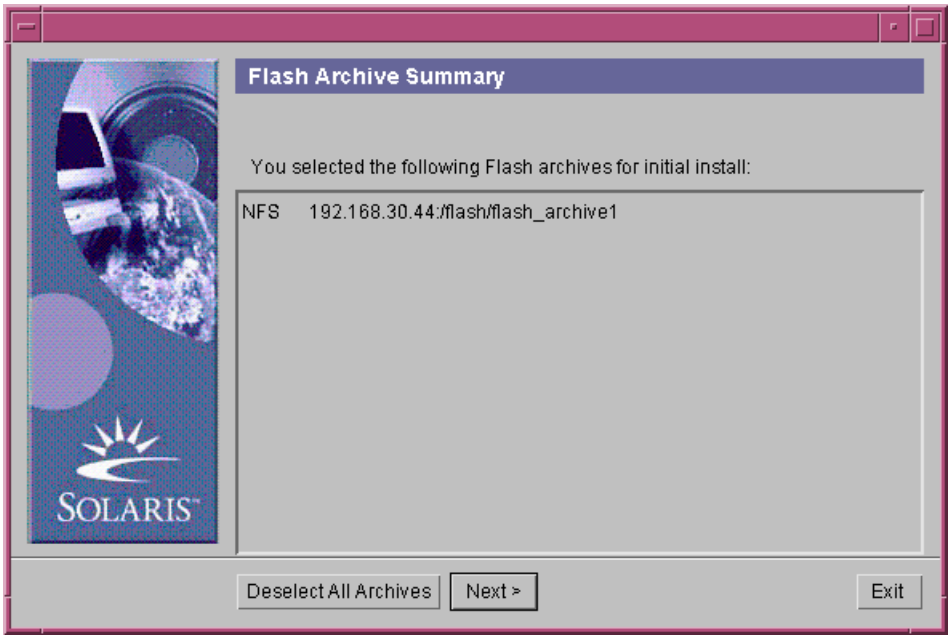


Figure 18-23 Flash Archive Summary Window

43. Click Next to continue.

In the Additional Flash Archives window, as shown in Figure 18-24, you can also select alternative sources for additional Flash archives. The alternative sources can be in another file system on the same master or on a different source altogether.

Figure 18-24 Additional Flash Archives Window

44. Because you are selecting a single archive, select None – Archive Selection Complete, as shown in Figure 18-24, to show that the archive selection is complete.
45. Click Next to continue.

46. Select the destination disk device from the Installer: Flash archive – Disk Selection window, as shown in Figure 18-25.

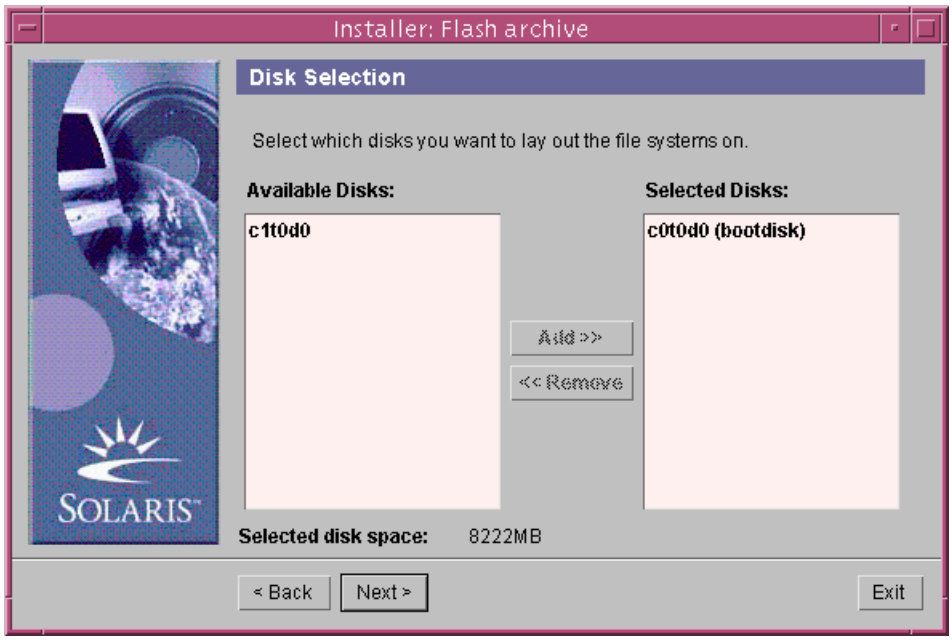


Figure 18-25 Installer: Flash archive – Disk Selection Window

47. Click Next to continue.

The size allocations are displayed in the Installer: Flash archive – Lay Out File Systems window, as shown in Figure 18-26, are sufficient for the archives that must be stored on the system disk.

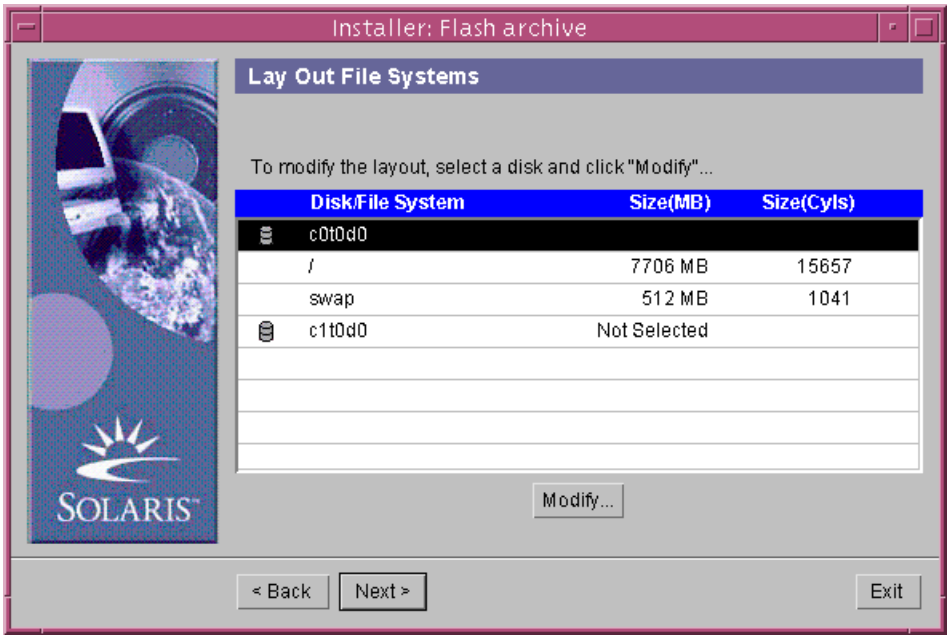


Figure 18-26 Installer: Flash archive – Lay Out File Systems Window

48. Click Next to continue.

The Installer: Flash archive – Ready to Install window, as shown in Figure 18-27, shows that the system evaluates your entries, and presents you with a summary for approval.

Figure 18-27 Installer: Flash archive – Ready to Install Window

49. Perform one of the following actions:
    - Click Install Now to accept your selections, and continue.
    - Click Back to change a selection.

The Installer: Flash archive – Installing window, as shown in Figure 18-28, shows that during the Flash archive installation, you see a standard installation slide-bar mechanism to trace the progress.

Figure 18-28 Installer: Flash archive – Installing Window

The Installer: Flash archive – Installation Summary window, as shown in Figure 18-29, shows the standard installation summary window when the installation is complete.

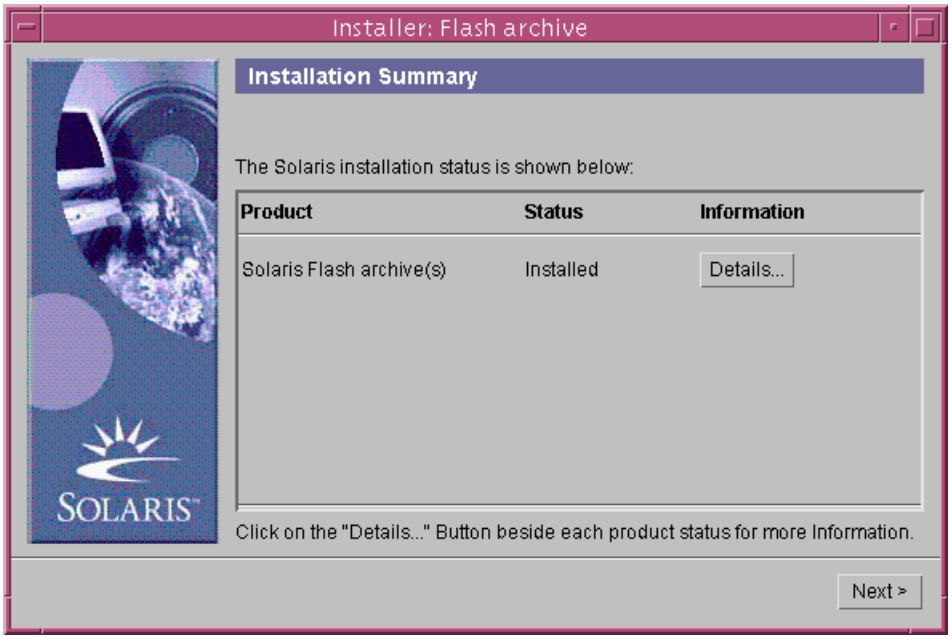


Figure 18-29 Installer: Flash archive – Installation Summary Window

50. Click Details to view the installation log file.

    The `/tmp/install_log` window provides a detailed summary of the flash installation process, and shows that the Flash archive installation is complete, as shown in Figure 18-30.

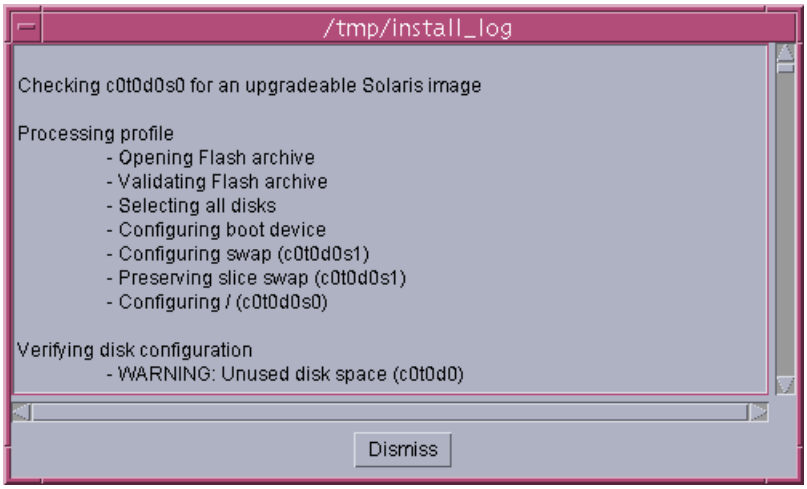


Figure 18-30 The `/tmp/install_log` Window

51. Click Dismiss to continue.

    At this point, the system prompts you to reboot, which terminates the installation process, and boots the system using the Solaris OE image that you just loaded using the Flash archive.

Using a Flash Archive With Interactive Install

You can perform interactive installation of the Solaris OE by using the `suninstall` program. The Solaris `suninstall` program only installs the Solaris OE software. After you install the Solaris OE software, you must use other installation programs to install additional software.

1. Insert Solaris 9 OE Installation 1 of 2 CD-ROM.
2. Boot the Flash clone system from the Boot PROM prompt as follows:

```
ok boot cdrom - nowin
```

After the pre-installation phase completes, a series of character-based curses screens appear. Figure 18-31 shows the window with information for the installation.

```
---------- The Solaris Installation Program ----------------------------

The Solaris installation program is divided into a series of short
sections where you'll be prompted to provide information for the
installation. At the end of each section, you can change the selections
you've made before continuing.
```

Figure 18-31 Solaris Installation Program – Step 1 Window

3. Read the `curses`-based content, answer any relevant prompts, and use the function or escape key sequences to progress to the next prompt.

As shown in Figure 18-32, you can select either an upgraded installation or an initial installation.

4. The Flash archive installation requires that you to follow the initial installation path, so press F4.

```
---------- Solaris Interactive Installation ----------------------------

This system is upgradable, so there are two ways to install the Solaris
software. The Upgrade option updates the Solaris software to the new
release,saving as many modifications to the previous version of Solaris
software as possible. Back up the system before using the Upgrade
option.

The Initial option overwrites the system disks with the new version of
Solaris software. This option allows you to preserve any existing file
systems. Back up any modifications made to the previous version of
Solaris software before starting the Initial option.

After you select an option and complete the tasks that follow, a summary
of your actions will be displayed. If you want to install the system with
a Flash archive, select Initial.


     F2_Upgrade    F4_Initial    F5_Exit    F6_Help
```

Figure 18-32 Solaris Interactive Installation – Step 2 Window

As shown in Figure 18-33, the installation provides more options for your Flash installation process. When you select the initial installation path, you can either perform a standard initial installation, which loads packages from standard installation media, or you can choose to use a Flash archive file to load a pre-configured Solaris OE archive file that provides a ready-to-use, bootable image of Solaris OE.

```
---------- Solaris Interactive Installation ----------------------------

You'll be using the initial option for installing Solaris software on the
system. The initial option overwrites the system disks when the new
Solaris software is installed.

On the following screens, you can accept the defaults or you can
customize how Solaris software will be installed by:
- Selecting the type of Solaris software to install
- Selecting disks to hold software you've selected
- Specifying how file systems are laid out on the disks

After completing these tasks, a summary of your selections (called a
profile) will be displayed.

There are two ways to install your Solaris software:

- "Standard" installs your system from a standard Solaris Distribution.
- "Flash" installs your system from one or more Flash Archives.


 Esc-2_Standard    F3_Go Back    Esc-4_Flash    F5_Exit    F6_Help
```

Figure 18-33 Solaris Interactive Installation – Step 3 Window

5. Press Esc-4 to select a Flash installation.

Note – Some system architectures recognize function keys, while others recognize the Escape key sequences. For example, `F2 = Esc2` and F4 = Esc4. In Figure 18-33, Esc2 = F2.

When performing Flash archive installations, you can select any one of five retrieval methods. Figure 18-34 shows the five retrieval methods. One commonly used version is to retrieve the archive from the master as NFS-shared files.

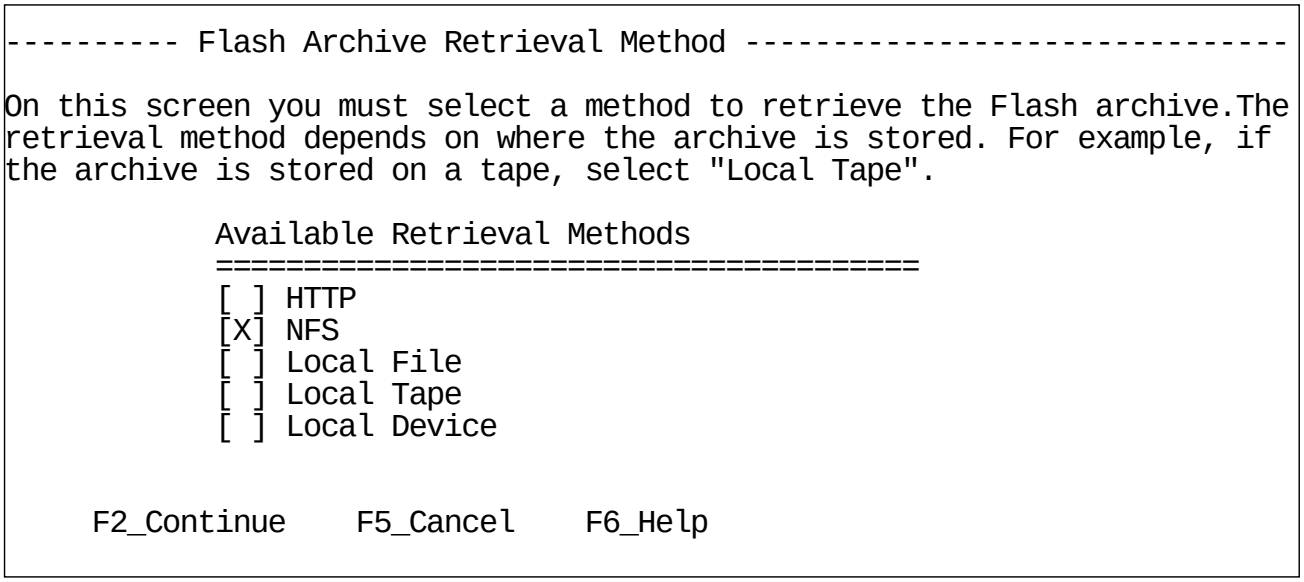


Figure 18-34 Flash Archive Retrieval Method Window

6. Select NFS, and press F2 to continue.

   When you select a retrieval method, you must select a specific location. In the NFS retrieval method, the next screen prompts you for the server and location. Remember to use the IP address of the server instead of the server name.

Next, you add a Flash archive, as shown in Figure 18-35. If the NFS file system is mounted and shared, and if you can locate the Flash archive within the file system, you are prompted for additional Flash archive names. A Solaris OE image must exist on a clone system before you can install additional Flash archives. The first Flash archive you install must also contain a bootable Solaris OE image.

```
---------- Flash Archive Addition ------------------------------------

Please specify the path to the network file system where the Flash
archive is located. For example:

     NFS Location: syrinx:/export/archive.flar


================================================================================

NFS Location: host:/path/to/flash_archive.flar
192.168.30.30:/flash/flash_archive1


     F2_Continue    F5_Cancel    F6_Help
```

Figure 18-35 Flash Archive Addition Window

7. Press F2 to continue.

Figure 18-36 shows the options available when selecting a Flash archive.

```
---------- Flash Archive Selection ------------------------------------

You selected the following Flash archives to use to install this system.
If you want to add another archive to install select "New".

        Retrieval Method      Name

=================================================================
        NFS                   flash_archive


     F2_Continue    F3_Go Back    F4_Edit    F5_New    F6_Help
```

Figure 18-36 Flash Archive Selection Window

8. Press F2 to continue.

   Figure 18-37 shows the steps in selecting a disk where you want to install the Flash archive. This disk is now the boot disk for the clone system.

```
---------- Select Disks -----------------------------------------------

On this screen you must select the disks for installing Solaris software
Start by looking at the Suggested Minimum field; this value is the
approximate space needed to install the software you've selected. Keep
selecting disks until the Total Selected value exceeds the Suggested
Minimum value.

           Disk Device (Size)        Available Space
        ==========================================
        [X] c0t0d0   (4102 MB) boot disk      4102 MB (F4 to edit)

                           Total Selected:   4102 MB
                       Suggested Minimum:    664 MB


     F2_Continue    F3_Go Back    F4_Edit    F5_Exit    F6_Help
```

Figure 18-37 Select Disks Window

9. Press F2 to continue.

Figure 18-38 shows the options that appear on the screen to preserve data. The system is queried and you are prompted to preserve any existing data on the target disk.

```
---------- Preserve Data? ----------------------------------------------

Do you want to preserve existing data? At least one of the disks you've
selected for installing Solaris software has file systems or unnamed
slices that you may want to save.


     F2_Continue    F3_Go Back    F4_Preserve    F5_Exit    F6_Help
```

Figure 18-38 Preserving Data Window

10. Press F4 to preserve the existing data and to continue.

    Figure 18-39 shows the layout of the file system and disk. This screen varies according to your disk partition specification in the preconfigured profile files. Explicit partitioning configures the disk as specified in the profile file, while existing partitioning specifies that you should leave the disk as currently configured. The existing specification brings up the next screen where you are prompted to customize the existing partitions.

```
---------- File System and Disk Layout ---------------------------------

The summary below is your current file system and disk layout, based on
the information you've supplied.

NOTE: If you choose to customize, you should understand file systems,
their intended purpose on the disk, and how changing them may affect the
operation of the system.

        File system/Mount point           Disk/Slice             Size
        ============================================================
        /                                 c0t0d0s0            1115 MB
        swap                              c0t0d0s1             589 MB
        overlap                           c0t0d0s2            4102 MB
        /export                           c0t0d0s7            2398 MB


    F2_Continue    F3_Go Back    F4_Customize    F5_Exit    F6_Help
```

Figure 18-39 File System and Disk Layout Window

11. Press F2 to continue.

Figure 18-40 shows the Mount Remote File Systems window.

```
---------- Mount Remote File Systems? ----------------------------------

Do you want to mount software from a remote file server? This may be
necessary if you had to remove software because of disk space problems.


    F2_Continue    F3_Go Back    F4_Remote Mounts    F5_Exit    F6_Help
```

Figure 18-40 Mount Remote File Systems Window

12. If your Flash archives are stored on the master Flash archive server, press F2 to continue.

    Figure 18-41 shows that the profiling phase of the Flash installation is now complete.

```
---------- Profile -----------------------------------------------------

 The information shown below is your profile for installing Solaris
software. It reflects the choices you've made on previous screens.
============================================================================

               Installation Option: Flash
                       Boot Device: c0t0d0
                   Client Services: None

                          Software: 1 Flash Archive
                                      NFS: flash_archive

       File System and Disk Layout: /                    c0t0d0s0 1115 MB
                                    swap                 c0t0d0s1  589 MB
                                    /export              c0t0d0s7 2398 MB

          * preserved file systems



     F2_Continue    F4_Change    F5_Exit    F6_Help
```

Figure 18-41 Profile Window

13. Review your selections and make changes, if necessary. If you are satisfied with the selections, press F2 to continue.

Figure 18-42 shows that the final step before beginning the Flash installation is to select either an automatic reboot or a manual reboot after the installation is done.

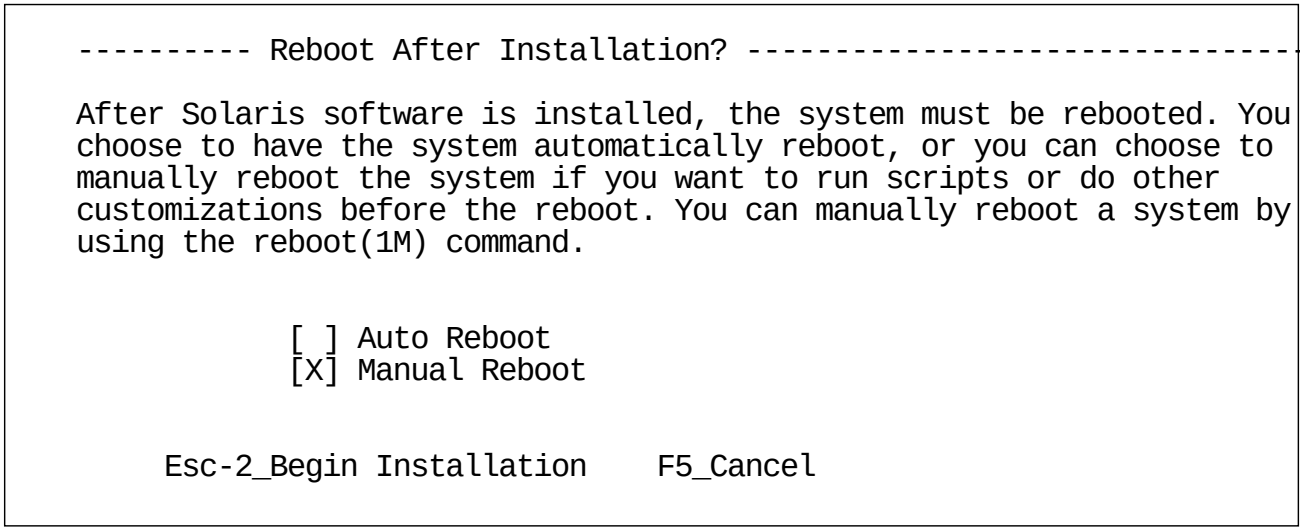


Figure 18-42 Reboot After Installation Window

14. Press Esc-2 to begin the installation.

When you start the installation, you see the volume table of contents (VTOC) information. Figure 18-43 shows the Preparing system for Solaris install window, which provides a progress slide bar and numerical indication of how far the installation has progressed.

```
Preparing system for Solaris install

Configuring disk (c0t0d0)
        - Creating Solaris disk label (VTOC)

Creating and checking UFS file systems
        - Creating / (c0t0d0s0

Beginning Solaris Installation

---------- Solaris Flash Install---------------------------------------


           MBytes Installed:    0.00
           MBytes Remaining:    511.00
                 Installing: Extracting Archive:
                             NFS
                             flash_archive



         |          |          |          |          |          |
         0          20         40         60         80         100
```

Figure 18-43 Preparing system for Solaris install Window

Figure 18-44 shows the steps involved in completing the Flash installation. After you install the Flash archive, the cleanup scripts complete the installation housekeeping tasks, and the system either reboots or prompts you to reboot, depending on your earlier configuration.

```
Customizing system files
        - Mount points table (/etc/vfstab)
        - Network host addresses (/etc/hosts)

Cleaning devices

Customizing system devices
        - Physical devices (/devices)
        - Logical devices (/dev)

Installing boot information
        - Installing boot blocks (c0t0d0s0)

Installation log location
        - /a/var/sadm/system/logs/install_log (before reboot)
        - /var/sadm/system/logs/install_log (after reboot)

Flash installation complete
Executing JumpStart postinstall phase...

The begin script log 'begin.log'
is located in /var/sadm/system/logs after reboot.
```

Figure 18-44 Completing the Flash Installation Window

Figure 18-45 on page 18-51 shows the reboot screen.

15. Reboot the system to complete the installation operation.

    Notice that the device configuration might not correspond to the devices on the system. It is usual to encounter errors on the first reboot after a Flash install, because the actual device configuration might differ between master and clone systems. The first reboot reconfigures the devices.

```
# reboot
syncing file systems... done
rebooting...
Resetting...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 300MHz), No Keyboard
OpenBoot 3.31, 256 MB (60 ns) memory installed, Serial #9685423.
Ethernet address 8:0:20:93:c9:af, Host ID: 8093c9af.

Rebooting with command: boot
Boot device: disk:a  File and args:
SunOS Release 5.9 Version Beta 64-bit
Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.
configuring IPv4 interfaces: hme0.
Hostname: sys41
Configuring /dev and /devices
devfsadm: opendir failed for /devices/pci@1f,4000/ebus@1: No such file or
directory
devfsadm: opendir failed for /devices/pci@1f,4000/ebus@1: No such file or
directory
devfsadm: opendir failed for /devices/pci@1f,4000/ebus@1: No such file or
directory
Configuring the /dev directory (compatibility devices)
The system is coming up. Please wait.
checking ufs filesystems
/dev/rdsk/c0t0d0s7: is clean.
Configuring network interface addresses: hme0.
Machine is an IPv4 router.
starting rpc services: rpcbind keyserv ypserv done.
Setting netmask of hme0 to 255.255.255.0
starting internet domain name server.
Setting default IPv4 interface for multicast: add net 224.0/4: gateway
sys41
syslog service starting.
Print services started.
Oct 18 15:26:51 sys41 sendmail[317]: My unqualified host name (sys41)
unknown; sleeping for retry
volume management starting.
The system is ready.

******************************************************************************
* Starting Desktop Login on display :0...
* Wait for the Desktop Login screen before logging in.
******************************************************************************
sys41 console login:
```

Figure 18-45 Rebooting the System After the Flash Installation

Using a Flash Archive With JumpStart Software

When you use Flash archives as the input source for JumpStart software, you must reconfigure a few of the JumpStart software configuration files to point to the Flash archive locations.

Network Files

When you use the JumpStart software method to deploy an archive to a clone, the clone follows a standard network boot process. The boot process uses Reverse Address Resolution Protocol (RARP) to get its Internet address and host name that are preconfigured on the master system files (`/etc/ethers` and `/etc/hosts`).

```
# more /etc/ethers
8:0:20:93:c9:af sys41
8:0:20:9e:dc:04 sys42
8:0:20:b5:98:25 sys43
8:0:20:99:f2:22 sys44

# more /etc/inet/hosts
.
.(output truncated)
.
192.168.30.30   instructor
192.168.30.41   sys41
192.168.30.42   sys42
192.168.30.43   sys43
192.168.30.44   sys44
```

JumpStart Keywords

The JumpStart process uses keywords (added to the JumpStart profile installation file) to determine specific configurations for the JumpStart process. The Flash keywords new to JumpStart are:

- `install_type flash_install`

  where:

`install_type`	Standard keyword
`flash_install`	The type of installation being performed

- `archive_location [`*`retrieval_type`*`] [`*`location`*`]`

  where:

`archive_location`	The new keyword for JumpStart software
`retrieval_type`	The file system type argument
`location`	The absolute path to the archive

Examples of locations:

```
archive_location nfs [server_name:/path/filename]
archive_location http [server_name:port_path/filename]
archive_location [local_tape_device_position]
```

Certain JumpStart profile keywords are incompatible with the deployment of the Flash archives. The following keywords support the selection of packages to be installed. Because a Flash deployment does not use the package add process, these keywords are incompatible in a Flash environment. Other packages are used for upgrades, which are not a part of Flash. Incompatible keywords are:

- `cluster`
- `package`
- `isa_bits`
- `geo`
- `backup_media`
- `layout_constraint`

The `flash_S9` file is a sample profile file for JumpStart using a Flash archive.

```
# cat flash_S9
install_type flash_install
archive_location nfs 192.168.30.30:/flash/flash_archive1
partitioning explicit
filesys c0t1d0s1 512 swap
filesys c0t1d0s0 free /
```

The `rules` file now points to the `flash_S9` profile file. You must run the `check` script to rebuild the `rules.ok` file.

```
# cat rules
any - - flash_S9 -
# ./check
Validating rules...
Validating profile ultra1_prof
The custom JumpStart configuration is ok.
```

The `sysidcfg` file is a standard JumpStart configuration file.

```
# cat sysidcfg
system_locale=en_US
timezone=US/Mountain
timeserver=192.168.30.xx
network_interface=primary{netmask=255.255.255.0 protocol_ipv6=no}
name_service=none
security_policy=none
```

Note – Configuration files, such as the *`profile`* file, the `rules` files, and the `sysidcfg` file, are stored in the NFS directory that is invoked by using the `add_install_client` command.

Add the host name for the clone, the path to the `sysidcfg` file, and the path to the archive using the `add_install_client` utility from the Solaris OE Installation CD 1 of 2, under the `/cdrom/Solaris_9_sparc/s0/Solaris_9/Tools` directory:

```
# ./add_install_client -p instructor/export/config \
-c instructor/export/config sys41 sun4u
making /tftpboot
enabling tftp in /etc/inetd.conf
updating /etc/bootparams
copying inetboot to /tftpboot
```

The `add_install_client` command shares the Solaris OE CD for the clone to boot. You must still share the `/flash` directory prior to booting the clone:

```
# share /flash
```

Check the shares:

```
# share
-               /export    ro,anon=0   ""
-               /flash     ro,anon=0   ""
```

Check the `/etc/bootparams` file to make sure that the command points to the correct installation directories:

```
# more /etc/bootparams
sys41 root=instructor:/export/install/Solaris_9/Tools/Boot
install=instructor:/export/install boottype=:in
sysid_config=instructor:/flash install_config=instructor:/flash
rootopts=:rsize=32768
```

JumpStart Using a Flash Archive

Use either the `init 0` or the `shutdown` command to bring the clone to the `ok` prompt, then boot the clone to the network.

```
ok boot net - install

SunOS Release 5.9 Beta Version Generic_110517-03 64-bit
Copyright 1983-2000 Sun Microsystems, Inc. All rights reserved.
whoami: no domain name
Configuring /dev and /devices
Using RPC Bootparams for network configuration information.
Configured interface hme0
Using sysid configuration file
192.168.30.30:/flash/sysidcfg
The system is coming up. Please wait.
Starting remote procedure call (RPC) services: sysidns done.
Starting Solaris installation program...
Searching for JumpStart directory...
Using rules.ok from 192.168.30.30:/flash
Checking rules.ok file...
Using profile: flash_S9
Executing JumpStart preinstall phase...
Searching for SolStart directory...
Checking rules.ok file...
```

```
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.

Processing default locales
- Specifying default locale (en_US.ISO8859-1)

Processing profile
- Opening Flash archive
- Validating Flash archive
- Selecting all disks
    - Configuring boot device
    - Preserving existing geometry for c0t0d0s0 (/)
    - Preserving existing geometry for c0t0d0s1 (swap)
    - Preserving existing geometry for c0t0d0s2 (overlap)
    - Preserving existing geometry for c0t0d0s3 (unnamed)
    - Preserving existing geometry for c0t0d0s4 (unnamed)
    - Preserving existing geometry for c0t0d0s5 (unnamed)
    - Preserving existing geometry for c0t0d0s6 (unnamed)
    - Preserving slice c0t0d0s7 (/export/home)

Verifying disk configuration

Verifying space allocation
    NOTE: 1 archives did not include size information

Configuring disk (c0t0d0)
    - Creating Solaris disk label (VTOC)

Creating and checking UFS file systems
    - Creating / (c0t0d0s0)
/dev/rdsk/c0t0d0s0:2285010 sectors in 2418 cylinders of 15 tracks, 63
sectors
    1115.7MB in 76 cyl groups (32 c/g, 14.77MB/g, 3712 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 30336, 60640, 90944, 121248, 151552, 181856, 212160, 242464, 272768,
 303072, 333376, 363680, 393984, 424288, 454592, 483872, 514176, 544480,
 574784, 605088, 635392, 665696, 696000, 726304, 756608, 786912, 817216,
 847520, 877824, 908128, 938432, 967712, 998016, 1028320, 1058624,
1088928,
 1119232, 1149536, 1179840, 1210144, 1240448, 1270752, 1301056, 1331360,
 1361664, 1391968, 1422272, 1451552, 1481856, 1512160, 1542464, 1572768,
 1603072, 1633376, 1663680, 1693984, 1724288, 1754592, 1784896, 1815200,
 1845504, 1875808, 1906112, 1935392, 1965696, 1996000, 2026304, 2056608,
 2086912, 2117216, 2147520, 2177824, 2208128, 2238432, 2268736,
    - Checking /export/home (c0t0d0s7)
```

```
/dev/rdsk/c0t0d0s7: 2 files, 9 used, 2403075 free
/dev/rdsk/c0t0d0s7: (19 frags, 300382 blocks, 0.0% fragmentation)

Beginning Flash archive extraction

Extracting archive: flash_archive
    Extracted    0.00 MB (  0% of  511.45 MB archive)
    Extracted    1.00 MB (  0% of  511.45 MB archive)
    Extracted    2.00 MB (  0% of  511.45 MB archive)
.
. (output truncated)
.
    Extracted  510.00 MB ( 99% of  511.45 MB archive)
    Extracted  511.00 MB ( 99% of  511.45 MB archive)
    Extracted  511.45 MB (100% of  511.45 MB archive)
    Extraction complete

Customizing system files
    - Mount points table (/etc/vfstab)
      fd- /dev/fdfd-no-
      /proc-/procproc-no-
      /dev/dsk/c0t0d0s1--swap-no-
      /dev/dsk/c0t0d0s0/dev/rdsk/c0t0d0s0/ufs1no-
      /dev/dsk/c0t0d0s7/dev/rdsk/c0t0d0s7/export/homeufs2yes-
      swap-/tmptmpfs-yes-
    - Network host addresses (/etc/hosts)

Cleaning devices

Customizing system devices
    - Physical devices (/devices)
    - Logical devices (/dev)

Installing boot information
    - Installing boot blocks (c0t0d0s0)


Installation log location
        - /a/var/sadm/system/logs/install_log (before reboot)
        - /var/sadm/system/logs/install_log (after reboot)

Flash installation complete
Executing JumpStart postinstall phase...

The begin script log 'begin.log'
is located in /var/sadm/system/logs after reboot.
```

```
syncing file systems... done
rebooting...
```

After the clone system has completely rebooted, log in to the clone and use the `ping` command to verify connectivity to the master.

Locating the Installation Logs

The error and message log resides in the `/var/adm/messages` file. The detailed installation log resides in the `/var/sadm/install_data/install_log` file.

Exercise Summary

Discussion – Take a few minutes to discuss the experiences, issues, or discoveries that you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Bibliography

Sun Microsystem Publications

- *Solaris™ 9 Installation Guide*, Part Number 806-5205-05.
- *Solaris 9 Installation Guide,* Part Number 806-5205-05
- *Solaris 9 Beta Reference Manual Collection, man pages section 4: File Formats*, Sun Part Number 816-0219-05
- *Solaris Management Console help*
- *System Administration Guide: Basic Administration*, Part Number 806-4073-10
- *System Administration Guide: Advanced Administration*, Part Number 806-4074-10
- *System Administration Guide: IP Services*, Part Number 806-4075-11
- *System Administration Guide: Security Services*, Part Number 806-4078-11
- *Solaris Volume Manager Administration Guide*, Part Number 806-6111-10
- *Solaris Smartcard Administration Guide*, Part Number 806-7010-10
- *Solaris 9 Installation Guide*, Part Number 816-5102
- *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*, Part Number 806-4077-06
- *System Administration Guide: Resource Management and Network Services*, Part Number 806-4076-07

Books

- Kasper, Paul Anthony and Alan L. McClellan. *Automating Solaris™ Installations – A Custom JumpStart™ Guide.* ISBN 0-13-312505-X.

Online Help

- *Solaris Management Console Toolbox Editor help*

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T