System Administration Guide: Resource Management and Network Services

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

Part No: 806–4076–07
December 2001

011025@2471

Contents

Tables

Figures

Examples

Preface

System Administration Guide: Resource Management and Network Services is part of a multi-volume set that covers a significant part of the Solaris™ system administration information. This book assumes that you have already installed the SunOS™ 5.9 operating system, and you have set up any networking software that you plan to use. The SunOS 5.9 operating system is part of the Solaris 9 product family, which also includes many features, including the Solaris Common Desktop Environment (CDE).

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

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Solaris 9 release. To use this book, you should have 1-2 years of UNIX® system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Volumes Are Organized

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

System Administration Guide: Basic Administration

- "Solaris Administration Tool Roadmap" in *System Administration Guide: Basic Administration*
- "Working With the Solaris Management Console Tools (Tasks)" in *System Administration Guide: Basic Administration*
- "Managing Users and Groups Topics" in *System Administration Guide: Basic Administration*
- "Managing Server and Client Support Topics" in *System Administration Guide: Basic Administration*
- "Shutting Down and Booting a System Topics" in *System Administration Guide: Basic Administration*
- "Managing Removable Media Topics" in *System Administration Guide: Basic Administration*
- "Managing Software Topics" in *System Administration Guide: Basic Administration*
- "Managing Devices Topics" in *System Administration Guide: Basic Administration*
- "Managing Disks Topics" in *System Administration Guide: Basic Administration*
- "Managing File Systems Topics" in *System Administration Guide: Basic Administration*
- "Backing Up and Restoring Data Topics" in *System Administration Guide: Basic Administration*

System Administration Guide: Advanced Administration

- "Managing Printing Services Topics" in *System Administration Guide: Advanced Administration*

- "Managing Terminals and Modems Topics" in *System Administration Guide: Advanced Administration*
- "Managing System Resources Topics" in *System Administration Guide: Advanced Administration*
- "Managing System Performance Topics" in *System Administration Guide: Advanced Administration*
- "Troubleshooting Solaris Software Topics" in *System Administration Guide: Advanced Administration*

System Administration Guide: IP Services

- "TCP/IP Topics" in *System Administration Guide: IP Services*
- "DHCP Topics" in *System Administration Guide: IP Services*
- "IPv6 Topics" in *System Administration Guide: IP Services*
- "IP Security Topics" in *System Administration Guide: IP Services*
- "Mobile IP Topics" in *System Administration Guide: IP Services*
- "IP Network Multipathing Topics" in *System Administration Guide: IP Services*

System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)

- "Overview of Naming and Directory Services" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- "The Name Service Switch" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- "Introduction to DNS" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- "Network Information Service (NIS): An Overview" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- "NIS+: An Introduction" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- "LDAP: An Overview" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- "Administering FNS: Attributes Overview" in *System Administration Guide: Naming and Directory Services*

System Administration Guide: Resource Management and Network Services

- "Resource Management Topics" in *System Administration Guide: Resource Management and Network Services*
- "SLP Topics" in *System Administration Guide: Resource Management and Network Services*
- "Mail Services Topics" in *System Administration Guide: Resource Management and Network Services*
- "Accessing Remote File Systems Topics" in *System Administration Guide: Resource Management and Network Services*
- "Modem–Related Network Services Topics" in *System Administration Guide: Resource Management and Network Services*
- "Working With Remote Systems Topics" in *System Administration Guide: Resource Management and Network Services*

System Administration Guide: Security Services

- "Security Services Overview" in *System Administration Guide: Security Services*
- "Using Authentication Services (Tasks)" in *System Administration Guide: Security Services*
- "Using Secure Shell (Tasks)" in *System Administration Guide: Security Services*
- "Introduction to SEAM" in *System Administration Guide: Security Services*
- "Managing System Security Topics" in *System Administration Guide: Security Services*
- "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*
- "Using Automated Security Enhancement Tool (Tasks)" in *System Administration Guide: Security Services*
- "Auditing Overview" in *System Administration Guide: Security Services*

System Administration Guide: Naming and Directory Services (FNS and NIS+)

- "NIS+: An Introduction" in *System Administration Guide: Naming and Directory Services (FNS and NIS+)*
- "Federated Naming Service (FNS)" in *System Administration Guide: Naming and Directory Services (FNS and NIS+)*

Related Books

This is a list of related documentation that is refered to in this book.

- *System Administration Guide: Advanced Administration*
- *System Administration Guide: Basic Administration*
- *System Administration Guide: IP Services*
- *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
- *System Administration Guide: Naming and Directory Services (FNS and NIS+)*
- *System Administration Guide: Resource Management and Network Services*
- *System Administration Guide: Security Services*
- Anderson, Bart, Bryan Costales, and Harry Henderson. *UNIX Communications*. Howard W. Sams & Company, 1987.
- Costales, Bryan. *sendmail, Second Edition*. O'Reilly & Associates, Inc., 1997.
- Frey, Donnalyn and Rick Adams. *!%@:: A Directory of Electronic Mail Addressing and Networks*. O'Reilly & Associates, Inc., 1993.
- Krol, Ed. *The Whole Internet User's Guide and Catalog*. O' Reilly & Associates, Inc., 1993.
- O' Reilly, Tim and Grace Todino. *Managing UUCP and Usenet*. O' Reilly & Associates, Inc., 1992.

Ordering Sun Documents

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

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

Accessing Sun Documentation Online

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

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

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

CHAPTER 1

Resource Management and Network Service Overview

Topics for the Solaris 9 Release

The following services or utilities are covered in this book:

Chapter 5	Resource management functionality enhances your ability to allocate, monitor, and control system resources.
Chapter 13	NFS is a protocol which provides the ability to access file systems from a remote host.
Chapter 17	SLP is a dynamic service discovery protocol.
Chapter 23	Mail services allow for a message to be sent to one or more people, while routing the message over whatever networks are necessary.
Chapter 28	PPP is a protocol that provides point-to-point links between remote hosts.
Chapter 37	UUCP enables hosts to exchange files
Chapter 41	These commands are used to access files on remote systems. The commands include `ftp`, `rlogin` and `rcp`.
Chapter 2	NCA provides improved web server performance by caching web pages.
"Perl 5" on page 44	The Practical Extraction and Report Language (Perl) is a tool that can be used to generate scripts to assist with system administration tasks.

Perl 5

This Solaris release includes Practical Extraction and Report Language (Perl) 5.6.1, a powerful general-purpose programming language that is generally available as free software. Perl has emerged as the standard development tool for complex system administration tasks, such as graphic, network, and World Wide Web programming, because of its excellent process, file, and text manipulation features.

Perl 5 includes a dynamically loadable module framework, which allows the addition of new capabilities for specific tasks. Many modules are freely available from the Comprehensive Perl Archive Network (CPAN), at http://www.cpan.org.

Accessing Perl Documentation

Several sources of information about Perl are included in this Solaris release. The same information is available by using these two mechanisms.

You can access the man pages by adding `/usr/perl5/man` to your MANPATH environment variable. This example displays the Perl overview.

```
% set MANPATH=($MANPATH /usr/perl5/man)
% man perl
```

You can access additional documentation by using the `perldoc` utility. This example displays the same overview information.

```
% /usr/perl5/bin/perldoc perl
```

The `perl` overview page lists of all the documentation that is included with the release.

Perl Compatibility Issues

In general, the 5.6.1 version of Perl is compatible with the previous version, so that scripts do not have to be rebuilt or recompiled to function. However, any XSUB-based (.xs) modules require recompilation and reinstallation.

In the Solaris 9 release, you can access the older version of Perl as `/usr/perl5/5.00503/bin/perl`. The older version might not be supported in future releases and should only be used until the new modules are rebuilt.

Changes to the Solaris Version of Perl

The Solaris version of Perl was compiled to include system malloc, 64-bit integer and large file support. In addition, appropriate patches have been applied. For a full list of all configuration information, review the results from this command.

```
% /usr/perl5/bin/perlbug -dv
---
Flags:
    category=
    severity=
---
Site configuration information for perl v5.6.1:
 .
 .
```

You can generate a shorter list by using `perl -V`.

CHAPTER 2

Managing Web Cache Servers

This chapter provides an overview of the Solaris Network Cache and Accelerator (NCA). Also, procedures for using NCA and reference material about NCA are included.

Network Cache and Accelerator (Overview)

The Solaris Network Cache and Accelerator (NCA) increases web server performance by maintaining an in-kernel cache of web pages that are accessed during HTTP requests. This in-kernel cache uses system memory to significantly increase performance for HTTP requests that are normally handled by web servers. Using system memory to hold web pages for HTTP requests increases web server performance by reducing the overhead between the kernel and the web server. NCA provides a sockets interface through which any web server can communicate with NCA with minimal modifications.

In situations where the requested page is retrieved from the in-kernel cache (cache hit) performance improved dramatically. In situations where the requested page in not in the cache (cache miss) and must be retrieved from the web server, performance is also significantly improved.

This product is intended to be run on a dedicated web server. Running other large processes on a server that runs NCA can cause problems.

NCA provides logging support in that it logs all cache hits. This log is stored in binary format to increase performance. The `ncab2clf` command can be used to convert the log from binary to common log format (CLF).

The Solaris 9 release includes the following enhancements:

- Sockets interface.
- Support for vectored sendfile which provides support for AF_NCA. See the `sendfilev`(3EXT) man page for more information.
- New options for the `ncab2clf` command that support the ability to skip records before a selected date (`-s`) and to process a given number of records (`-n`).

Managing Web Cache Servers (Task Map)

The following table describes the procedures that are needed to use NCA.

Task	Description	For Instructions, Go To
Planning for NCA	A list of requirements to enable the use of NCA. Review before configuring NCA.	"System Requirements for NCA" on page 49
Enabling NCA	Steps to enable in-kernel caching of web pages on a web server.	"How to Enable Caching of Web Pages" on page 49
Disabling NCA	Steps to disable in-kernel caching of web pages on a web server.	"How to Disable Caching of Web Pages" on page 50
Administering NCA logging	Steps to enable or disable the NCA logging process.	"How to Enable or Disable NCA Logging" on page 51
Loading the NCA socket library	Steps to use NCA if the AF_NCA socket is not supported.	"How to Load the NCA Socket Utility Library" on page 51

Administering the Caching of Web Pages (Tasks)

The following sections cover the system requirement that are needed to use NCA and the procedures to enable or disable parts of the service.

System Requirements for NCA

To support NCA, the system must meet these requirements:

- 256 Mbytes RAM must be installed.
- The Solaris 9 release or one of the Solaris 8 upgrade releases must be installed.
- Apache support must be available. Apache support is available in the Solaris 9 and the Solaris 8 upgrade releases.

This product is intended to be run on a dedicated web server. Running other large processes on a server that runs NCA can cause problems.

▼ How to Enable Caching of Web Pages

1. **Become superuser.**

2. **Register the interfaces.**

   Enter the names of each of the physical interfaces in the `/etc/nca/nca.if` file. See the `nca.if`(4) man page for more information.

   ```
   # cat /etc/nca/nca.if
   hme0
   hme1
   ```

   Each interface must have an accompanying `hostname.`*interface-name* file and an entry in `/etc/hosts` file for the contents of `hostname.`*interface-name*. To start the NCA feature on all interfaces, place an asterisk, `*`, in the `nca.if` file.

3. **Enable the `ncakmod` kernel module.**

   Change the `status` entry in `/etc/nca/ncakmod.conf` to `enabled`.

   ```
   # cat /etc/nca/ncakmod.conf
   #
   # NCA Kernel Module Configuration File
   #
   status=enabled
   ```

```
httpd_door_path=/var/run/nca_httpd_1.door
nca_active=disabled
```

See the `ncakmod.conf`(4) man page for more information.

4. **Enable NCA logging.**

   Change the `status` entry in `/etc/nca/ncalogd.conf` to `enabled`.

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=enabled
logd_path_name="/var/nca/log"
logd_file_size=1000000
```

   You can change the location of the log file by changing the path that is indicted by the `logd_path_name` entry. See the `ncalogd.conf`(4) man page for more information.

5. **For IA only: Increase the virtual memory size.**

   Use the `eeprom` command to set the `kernelbase` of the system.

```
# eeprom kernelbase=0x90000000
# eeprom kernelbase
kernelbase=0x90000000
```

   The second command verifies that the parameter has been set.

Note – Setting the `kernelbase` of the system reduces the amount of virtual memory that user processes can use to less than 3 Gbytes, which means that the system is not ABI compliant. When the system boots, it displays a message that warns you about noncompliance. Most programs do not actually need the full 3–Gbyte virtual address space. If you have a program that needs more than 3 Gbytes, you need to run it on a system that does not have NCA enabled.

---

6. **Reboot the server.**

▼ How to Disable Caching of Web Pages

1. **Become superuser.**

2. **Disable the `ncakmod` kernel module.**

   Change the `status` entry in `/etc/nca/ncakmod.conf` to `disabled`.

```
# cat /etc/nca/ncakmod.conf
# NCA Kernel Module Configuration File
#
status=disabled
httpd_door_path=/var/run/nca_httpd_1.door
```

```
nca_active=disabled
```

See the `ncakmod.conf`(4) man page for more information.

3. **Disable NCA logging.**

   Change the `status` entry in `/etc/nca/ncalogd.conf` to `disabled`.

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=disabled
logd_path_name="/var/nca/log"
logd_file_size=1000000
```

   See the `ncalogd.conf`(4) man page for more information.

4. **Reboot the server.**

▼ How to Enable or Disable NCA Logging

NCA logging can be turned on or off as needed after NCA has been enabled. See "How to Enable Caching of Web Pages" on page 49 for more information.

1. **Become superuser.**

2. **Change NCA logging.**

   To permanently disable logging, you need to change the status in `/etc/nca/ncalogd.conf` to disabled and reboot the system. See the `ncalogd.conf`(4) man page for more information.

   a. **To stop logging:**

      Type the following command.

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

   b. **To start logging:**

      Type the following command.

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

▼ How to Load the NCA Socket Utility Library

Follow this process only if your web server does not provide native support of the AF_NCA socket.

Add a line to the web server startup script that causes the library to be preloaded. The line should resemble the following:

```
LD_PRELOAD=/usr/lib/ncad_addr.so /usr/bin/httpd
```

Caching Web Pages (Reference)

The following sections cover the files and components that are needed to use NCA. Also, specifics about how NCA interacts with the web server are included.

NCA Files

You need several files to support the NCA feature. Many of these files are ASCII, but some of them are binary. The following table lists all of the files.

TABLE 2–1 NCA Files

File Name	Function
`/etc/hostname.*`	File that lists all physical interfaces that are configured on the server.
`/etc/hosts`	File that lists all host names that are associated with the server. Entries in this file must match entries in `/etc/hostname.*` files for NCA to function.
`/etc/init.d/ncakmod`	Script that starts the NCA server. It is run when a server is booted.
`/etc/init.d/ncalogd`	Script that starts NCA logging. It is run when a server is booted.
`/etc/nca/nca.if`	File that lists the interfaces on which NCA is run. See the `nca.if`(4) man page for more information.
`/etc/nca/ncakmod.conf`	File that lists configuration parameters for NCA. See the `ncakmod.conf`(4) man page for more information.
`/etc/nca/ncalogd.conf`	File that lists configuration parameters for NCA logging. See the `ncalogd.conf`(4) man page for more information.
`/usr/bin/ncab2clf`	Command that is used to convert data in the log file to the common log format. See the `ncab2clf`(1) man page for more information.

TABLE 2–1 NCA Files *(Continued)*

File Name	Function
`/usr/lib/net/ncaconfd`	Command that is used to configure NCA to run on multiple interfaces during boot. See the `ncaconfd`(1M) man page for more information.
`/usr/lib/nca_addr.so`	Library that uses AF_NCA sockets instead of AF_INET sockets. This library must be used on web servers that use AF_INET sockets. See the `ncad_addr`(4) man page for more information.
`/var/nca/log`	File that holds the log file data. The file is in binary format, so do not edit it.

NCA Architecture

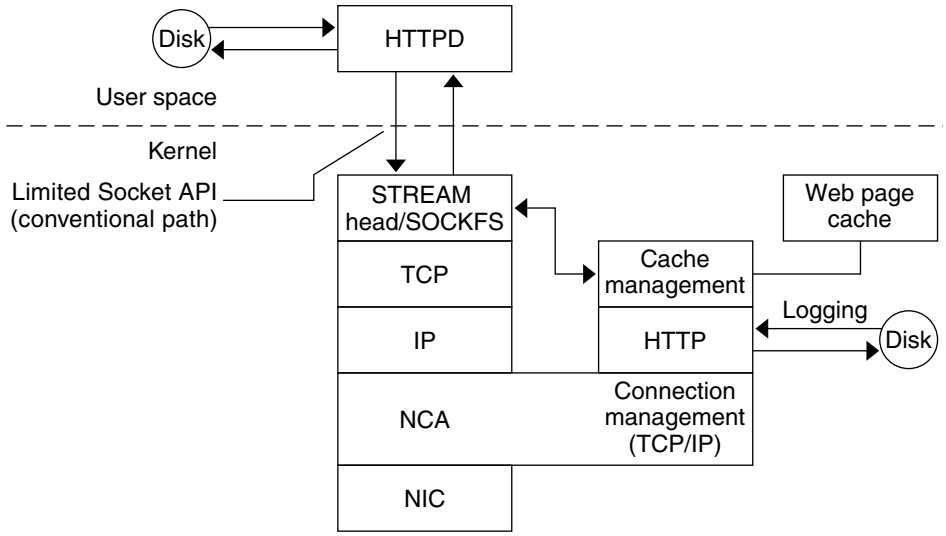
The NCA feature includes the following components.

- Kernel module, `ncakmod`
- Web server, `httpd`

The kernel module `ncakmod` maintains the cache of web pages in system memory. The module communicates with a web server, `httpd`, through a sockets interface (family type PF_NCA) .

The kernel module also provides a logging facility that logs all HTTP cache hits. NCA logging writes HTTP data to the disk in binary format. NCA provides a conversion utility for converting binary log files to common log format (CLF).

The following figure shows the flow of data for the conventional path and the path that is used when NCA is enabled.



NCA to Httpd Request Flow

The following list shows the request flow between the client and the web server.

1. An HTTP request is made from the client to the web server.
2. If the page is in cache, the in-kernel cache web page is returned.
3. If the page is not in cache, the request goes to the web server to retrieve or update the page.
4. Depending on the HTTP protocol semantics that are used in the HTTP response, the page is cached or not and then it is returned to the client. If the Pragma: No-cache header is included in the HTTP request, the page is not cached.

Interpositioning Library for Door Server Daemon Support

Many web servers use AF_INET sockets. By default, NCA uses AF_NCA sockets. To correct this situation, an interpositioning library is provided. The new library is loaded in front of the standard socket library, libsocket.so. The library call `bind()` is interposed by the new library, ncad_addr.so. If the status is enabled in `/etc/nca/ncakmod.conf`, the version of Apache that is included with the Solaris 9 release is already set up to call this library. If you are using IWS or Netscape servers, see "How to Load the NCA Socket Utility Library" on page 51 to use the new library.

CHAPTER 3

Time Related Services

Keeping system clocks synchronized within a network is required for many databases and authentication services. The following topics are covered in this chapter.

Clock Synchronization (Overview)

The Network Time Protocol (NTP) public domain software from the University of Delaware is included in the Solaris software starting with the Solaris 2.6 release. The `xntpd` daemon sets and maintains the system time-of-day. The `xntpd` daemon is a complete implementation of the version 3 standard, as defined by RFC 1305.

The `xntpd` daemon reads the `/etc/inet/ntp.conf` file at system startup. See `xntpd`(1M) for information about configuration options.

Keep the following in mind when using NTP in your network:

- The `xntpd` daemon takes up minimal system resources.
- An NTP client synchronizes automatically with an NTP server when it boots. If the client becomes unsynchronized, the client will resynchronize again when the client sees a time server.

Another way to synchronize clocks is to run `rdate` using `cron`.

Managing Network Time Protocol (Tasks)

The following procedures show how to set up and use the NTP service.

▼ How to Set Up an NTP Server

1. **Become superuser.**
2. **Create the** `ntp.conf` **file.**

   To ensure proper execution of the `xntpd` daemon, the `ntp.conf` file must first be created. The `ntp.server` file can be used as a template.

   ```
   # cd /etc/inet
   # cp ntp.server ntp.conf
   ```

3. **Start the** `xntpd` **daemon.**

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

▼ How to Set Up an NTP Client

1. **Become superuser.**
2. **Create the** `ntp.conf` **file.**

   To activate the `xntpd` daemon, the `ntp.conf` file must first be created.

   ```
   # cd /etc/inet
   # cp ntp.client ntp.conf
   ```

3. **Start the** `xntpd` **daemon.**

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

Using Other Time Related Commands (Tasks)

▼ How to Synchronize Date and Time From Another System

1. **Become superuser.**
2. **Reset the date and time to synchronize with another system, by using the `rdate` command.**

   `# rdate` *another-system*

   Where *another-system* is the name of another system.
3. **Verify that you have reset your system's date correctly by using the `date` command.**

   The output should show a date and time that matches that of the other system.

Example—Synchronizing Date and Time From Another System

The following example shows how to use `rdate` to synchronize the date and time of one system with another. In this example, the system `earth`, running several hours behind, is reset to match the date and time of the server `starbug`.

```
earth# date
Tue Jun  5 11:08:27 MDT 2001
earth# rdate starbug
Tue Jun  5 14:06:37 2001
earth# date
Tue Jun  5 14:06:40 MDT 2001
```

Network Time Protocol (Reference)

The following files are needed for the NTP service to run.

TABLE 3–1 NTP Files

File Name	Function
`/etc/inet/ntp.conf`	Lists configuration options for NTP.
`/etc/inet/ntp.client`	Sample configuration file for NTP clients.
`/etc/inet/ntp.server`	Sample configuration file for NTP servers.
`/etc/inet/ntp.drift`	Sets the initial frequency offset on NTP servers.
`/etc/inet/ntp.keys`	Sample configuration file for NTP servers.
`/etc/init.d/xntpd`	NTP startup script run when a host is booted.
`/usr/lib/inet/xntpd`	NTP daemon. See `xntpd`(1M) for more information.
`/usr/sbin/ntpdate`	Utility to set the local date and time based on NTP. See `ntpdate`(1M) for more information.
`/usr/sbin/ntpq`	NTP query program. See `ntpq`(1M) for more information.
`/usr/sbin/ntptrace`	Program to trace NTP hosts back to the master NTP server. See `ntptrace`(1M) for more information.
`/usr/sbin/xntpdc`	NTP query program for the xntpd daemon. See `xntpdc`(1M) for more information.
`/var/ntp/ntpstats`	Directory for holding NTP statistics.

CHAPTER 4

Solaris 9 Resource Manager Topics

The section contains the following chapters on resource management in the Solaris operating environment.

Chapter 5	Provides an overview of resource management and discusses why you would want to use the functionality on your system.
Chapter 6	Covers the projects and tasks facilities and describes how they are used to label and separate workloads.
Chapter 7	Describes the extended accounting functionality used to capture detailed resource consumption statistics for capacity planning or billing purposes.
Chapter 8	Discusses resource controls, which are used to place bounds on resource usage by applications running on your system.
Chapter 9	Describes the fair share scheduler, which uses shares to specify the amounts of CPU time allocated to processes running on your system.
Chapter 10	Describes resource pools, which are used to partition system resources and guarantee that a known amount of resources will always be available to a specified workload running on your system.
Chapter 11	Describes the resource management functionality available in the Solaris Management Console™ tool.

CHAPTER 5

Introduction to Solaris 9 Resource Manager

Resource management functionality enables you to control how applications use available system resources. You can:

- Allocate computing resources, such as processor time
- Monitor how these allocations are being used and adjust them as necessary
- Generate extended accounting information for analysis, billing, and capacity planning

Overview

Modern computing environments have to provide a flexible response to the varying workloads generated by different applications on a system. If resource management features are not used, the Solaris operating environment responds to workload demands by adapting to new application requests dynamically. This default response generally means that all activity on the system is given equal access to resources. Solaris resource management features enable you to treat workloads individually. You can:

- Restrict access to a given resource
- Offer resources to workloads on a preferential basis
- Isolate workloads from one another

The ability to minimize cross-workload performance compromises, combined with the facilities that monitor resource usage and utilization, are collectively referred to as *resource management*. Resource management is implemented through a collection of algorithms that handle the series of capability requests that an application presents in the course of its execution.

Resource management facilities permit you to modify the default behavior of the operating system with respect to different workloads. In this case, *behavior* primarily refers to the set of decisions made by operating system algorithms when the system is presented with one or more resource requests by an application. You can use resource management facilities to:

- Deny resources or prefer one application over another for a larger set of allocations than otherwise permitted
- Treat certain allocations collectively instead of through isolated mechanisms

The implementation of a system configuration using the resource management facilities can serve several purposes. You can:

- Prevent an application from consuming resources indiscriminately
- Change an application's priority based on external events
- Balance resource guarantees to a set of applications against the goal of maximizing system utilization

When planning a resource-managed configuration, key requirements include the following:

- Identifying the competing workloads on the system
- Distinguishing those workloads that are not in conflict from those with performance requirements that compromise the primary workloads

Once cooperating and conflicting workloads have been identified, you can create a resource configuration that presents the least compromise to the service goals of the business, within the limitations of the system's capabilities.

Effective resource management is enabled in the Solaris environment by offering control mechanisms (see "Resource Management Control Mechanisms" on page 62), notification mechanisms (see Chapter 8), and monitoring mechanisms (see Chapter 8). Many of these capabilities are provided through enhancements to existing mechanisms such as the `proc`(4) file system, processor sets, and scheduling classes. Other capabilities are specific to resource management. These capabilities are described in Chapter 6, Chapter 7, Chapter 8, and Chapter 9.

Resource Management Control Mechanisms

The three types of control mechanisms available in the Solaris operating environment are constraints, scheduling, and partitioning.

Constraints

Constraints allow the administrator or application developer to ensure that there are known bounds on the consumption of specific resources for a workload. With known bounds, modelling resource consumption scenarios becomes a simpler process. Bounds can also be used to control ill-behaved applications that would otherwise compromise system performance or availability through unregulated resource requests.

Constraints do present complications to the application. It is even possible to modify the relationship between the application and the system to the point that the application is no longer able to function. One approach that can mitigate this risk is to gradually narrow the constraints on applications with unknown resource behavior. The resource controls feature discussed in Chapter 8 provides a constraint mechanism. Newer applications can be written to be aware of their resource constraints, but not all application writers will choose to do this.

Scheduling

Scheduling refers to making a sequence of allocation decisions at given intervals. The decision made is based on a predictable algorithm. An application that does not need its current allocation leaves the resource available for another application's use. Scheduling-based resource management permits full utilization of an undercommitted configuration, while providing controlled allocations in a critically committed or overcommitted scenario. The underlying algorithm defines how the term "controlled" is interpreted. In some cases, the scheduling algorithm might guarantee that all applications have some access to the resource. The fair share scheduler described in Chapter 9 manages application access to CPU resources in a controlled way.

Partitioning

Partitioning is used to bind a workload to a subset of the system's available resources. This binding guarantees that a known amount of resources will always be available to the workload. The resource pools functionality described in Chapter 10 allows you to limit workloads to specific subsets of the machine. Configurations that use partitioning can avoid system-wide overcommitment. However, in avoiding this overcommitment, the possibility for achieving high utilizations is reduced because a reserved group of resources (such as processors) is not available for use when the workload bound to them is idle.

Resource Classifications

A resource is any aspect of the computing system that can be manipulated with the intent to change application behavior. Thus, a resource is a capability that an application implicitly or explicitly requests which, if denied or constrained, causes the execution of a robustly written application to proceed more slowly.

Classification of resources (as opposed to identification of resources) can be made along a number of axes, such as implicitly requested versus explicitly requested, time-based versus time-independent, and so forth.

Generally, scheduler-based resource management is applied to resources that the application can implicitly request. For example, to continue execution, an application implicitly requests additional CPU time, while to write data to a network socket, an application implicitly requests bandwidth. Additional interfaces can be presented so that bandwidth or CPU service levels can be explicitly negotiated.

Resources that are explicitly requested, such as a request for an additional thread, can be managed by constraint. Constraints can be placed on the aggregate total use of an implicitly requested resource.

Resource Management Configuration

Portions of the resource management configuration can be placed in a network name service. This feature allows the administrator to apply resource management constraints across a collection of machines, rather than on an exclusively per-machine basis. Related work can share a common identifier, and the aggregate usage of that work can be tabulated from accounting data.

Resource management configuration and workload-oriented identifiers are described more fully in Chapter 6. The extended accounting facility that combines these identifiers with application resource usage is described in Chapter 7.

When to Use Resource Management

Use resource management to ensure that your applications get the response times they require.

Resource management can also increase resource utilization. By categorizing and prioritizing usage, you can effectively use reserve capacity during off-peak periods, often eliminating the need for additional processing power. You can also ensure that CPU cycles are not wasted due to load variability.

Server Consolidation

Resource management is ideal for environments that consolidate a number of applications on a single server.

The cost and complexity of managing numerous machines encourages consolidating several applications on larger, more scalable servers. Instead of running each workload on a separate system, with full access to that system's resources, you can use resource management software to segregate workloads within the system. Resource management makes it possible to lower overall total cost of ownership by running and controlling several dissimilar applications on a single Solaris system.

If you are providing Internet and application services, you can use resource management to:

- Host multiple web servers on a single machine, controlling the resource consumption associated with each web site and protecting each site from the potential excesses of other sites
- Prevent a faulty common gateway interface (CGI) script from exhausting CPU resources
- Stop an incorrectly behaving application from leaking all available virtual memory, and ensure that one customer's applications will not be affected by another customer's applications running at the same site
- Obtain accounting information for billing purposes

Supporting a Large or Varied User Population

Use resource management features in any system that has a large and diverse user base, such as an educational institution. Where there is a mix of workloads, the software can be configured to favor specific projects.

For example, in large brokerage firms, traders intermittently require fast access to execute a query or to perform a calculation. Other system users, however, have more consistent workloads. If you allocate a proportionately larger amount of processing power to the traders' projects, the traders will have the responsiveness they need.

Resource management is also ideal for supporting thin client systems. These platforms provide stateless consoles with frame buffers and input devices, such as smart cards. The actual computation is done on a shared server, resulting in a timesharing type of environment. Use resource management features to isolate the users on the server so that excess load generated by one user does not monopolize hardware resources and significantly impact others using the system.

Setting Up Resource Management (Task Map)

The following task map gives a basic overview of the steps involved in setting up resource management on your system.

Task	Description	For Instructions
Identify the workloads on your system.	Review project entries in either the `/etc/project` database file or in the NIS map or LDAP directory service.	"`project` Database" on page 71
Prioritize the workloads on your system.	Determine which applications are critical. These workloads might require preferential access to resources.	Refer to your business service goals.
Monitor real-time activity on your system.	Use performance tools to view the current resource consumption of workloads running on your system. This will help you evaluate whether you must restrict access to a particular resource or isolate workloads from one another.	"Monitoring by System" on page 130, `cpustat`(1M), `iostat`(1M), `mpstat`(1M), `prstat`(1M), and `vmstat`(1M)
Make temporary modifications to the workloads running on your system.	To determine which values can be altered, refer to the resource controls available in the Solaris environment. You can update the values from the command line while the task or process is running.	"Available Resource Controls" on page 88, "Actions on Resource Control Values" on page 89, "Temporarily Updating Resource Control Values on a Running System" on page 92, `rctladm`(1M), and `prctl`(1)

Task	Description	For Instructions
Set resource control attributes for each project entry in the `project`database or name service project table.	Each project entry in the `/etc/project` database or the name service project table can contain one or more resource controls that constrain tasks and processes attached to that project. For each threshhold value placed on a resource control, you can associate one or more actions to be taken when that value is reached. You can set resource controls using the command line interface or the Solaris Management Console. If you are setting configuration parameters across a large number of systems, use the console for this task.	"`project` Database" on page 71, "Local `project` File Format" on page 72, "Available Resource Controls" on page 88, "Actions on Resource Control Values" on page 89, and Chapter 9
Create resource pools configurations.	Resource pools provide a way to partition system resources, such as processors, and maintain those partitions across reboots. You can add a `project.pool` attribute to each entry in the `/etc/project` database.	"Creating Pools Configurations" on page 115
Make the fair share scheduler (FSS) your default system scheduler.	Ensure that all user processes in either a single CPU system or a processor set belong to the same scheduling class.	"FSS Configuration Examples" on page 107 and `dispadmin`(1M)
Activate the extended accounting facility to monitor and record resource consumption on a task or process basis.	Use extended accounting to assess current resource controls and plan capacity requirements for future workloads. Aggregate usage on a system-wide basis can be tracked. To obtain complete usage statistics for related workloads that span more than one system, the project name can be shared across several machines.	"How to Activate Extended Accounting for Processes and Tasks" on page 84 and `acctadm`(1M)
(Optional) If you determine that additional adjustments to your configuration are required, you can continue to alter the values from the command line while the task or process is running.	Modifications can be applied on a temporary basis without restarting the project. Tune the values until you are satisfied with the performance. Then update the current values in the `/etc/project` database or the name service project table.	"Temporarily Updating Resource Control Values on a Running System" on page 92, `rctladm`(1M), and`prctl`(1)

Task	Description	For Instructions
(Optional) Capture extended accounting data.	Record resource consumption on a task or project basis. The files produced can be used for planning, chargeback, and billing purposes.	`wracct`(1M)

CHAPTER 6

Projects and Tasks

This chapter discusses the *project* and *task* facilities of Solaris resource management. Projects and tasks are used to label workloads and separate them from one another. The project provides a network-wide administrative identifier for related work. The task collects a group of processes into a manageable entity representing a workload component.

Overview

To optimize workload response, you must first be able to identify the workloads that are running on the system you are analyzing. This information can be difficult to obtain by using either a purely process-oriented or a user-oriented method alone. In the Solaris environment, you have two additional facilities to apply to separating and identifying workloads, the project and the task.

Running processes can be manipulated with standard Solaris commands, based on their project or task membership. The extended accounting facility can report on both process usage and task usage, and tag each record with the governing project identifier. This process allows offline workload analysis to be correlated with online monitoring. The project identifier itself can be shared across multiple machines through the `project` name service database. Thus, the resource consumption of related workloads that run on (or span) multiple machines can ultimately be analyzed across all of the machines.

Projects

The project identifier is an administrative identifier used to identify related work. The project identifier can be thought of as a workload tag equivalent to the user and group identifiers. A user or group can belong to one or more projects. These projects can be used to represent the workloads in which the user (or group of users) is allowed to participate. This membership can then be the basis of chargeback based on, for example, usage or initial resource allocations. Although a user must have a default project assigned, the processes that the user launches can be associated with any of the projects of which that user is a member.

Determining a User's Default Project

To log in to the system, a user must be assigned a default project.

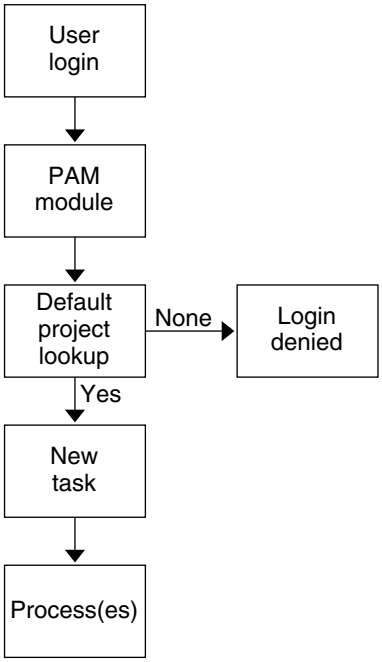


FIGURE 6–1 User Login Process

Because each process on the system possesses project membership, an algorithm to assign a default project to the login or other initial process is necessary. The algorithm

to determine a default project consists of four steps. If no default project is found, the user's login (or request to start a process) is denied.

The system follows these steps to determine a user's default project:

1. If the user has an entry with a `project` attribute defined in the `/etc/user_attr` extended user attributes database, then the value of the `project` attribute is the default project (see `user_attr`(4)).
2. If a project with the name `user.`*user-id* is present in the `project`(4) database, then that project is the default project.
3. If a project with the name `group.`*group-name* is present in the `project` database, where *group-name* is the name of the default group for the user (as specified in `passwd`(4)), then that project is the default project.
4. If the special project `default` is present in the `project` database, then that project is the default project.

This logic is provided by the `getdefaultproj()` library function (see `getprojent`(3PROJECT)).

Because information might be located in the `user_attr` database, as specified in the `/etc/nsswitch.conf` file, the search order does not strictly follow that of `project`(4). The `project`(4) and `user_attr`(4) databases can be stored in different sources, or these sources can be accessed in different orders.

`project` Database

You can store project data in a local file, in a Network Information Service (NIS) project map, or in a Lightweight Directory Access Protocol (LDAP) directory service. The `/etc/project` database or name service is used at login and by all requests for account management by the Pluggable Authentication Module (PAM) to bind a user to a default project.

Note – Updates to entries in the project database, whether to the `/etc/project` file or to a representation of the database in a network name service, are not applied to currently active projects. The updates are applied the next time `login`(1) or `newtask`(1) is used. You can update values on a temporary basis without restarting the project (see `prctl`(1)).

PAM Subsystem

When an operation involves changing or setting identity (such as logging in to the system, invoking an `rcp` or `rsh` command, using `ftp`, or using `su`), a set of

configurable modules is used to provide authentication, account management, credentials management, and session management.

The PAM system as a whole is documented in the man pages `pam`(3PAM), `pam.conf`(4), and `pam_unix`(5). The account management PAM module for projects is documented in the `pam_projects`(5) man page.

Name Service Configuration

Resource management supports the name service `project` database. To include `/etc/nsswitch.conf` support for the `project` database, use the following options:

```
project: files [nis] [ldap]
```

The location where the `project` database is stored is defined in `/etc/nsswitch.conf`. By default, `files` is listed first, but the sources can be listed in any order. If more than one source for project information is listed, the `nsswitch.conf` file directs the routine to start searching for the information in the first source listed, then search subsequent databases.

For more information on `/etc/nsswitch.conf`, see "The Name Service Switch" in *System Administration Guide: Naming and Directory Services* and `nsswitch.conf`(4).

Local `project` File Format

If you select `files` as your `project` database in `nsswitch.conf`, the login process searches the `/etc/project` file for project information (see `projects`(1) and `project`(4)). The `project` file contains a one-line entry for each project recognized by the system, of the form:

```
projname:projid:comment:user-list:group-list:attributes
```

The fields are defined as follows:

projname — The name of the project. It must be a string consisting of alphanumeric characters, the underline (_) character, and the hyphen (-). The name must begin with an alphabetic character. It cannot contain periods (.), colons (:), or newline characters.

projid — The project's unique numerical ID (PROJID) within the system. The maximum value of the `projid` field is UID_MAX.

comment — The project's description.

user-list — A comma-separated list of users allowed in the project.

group-list — A comma-separated list of groups of users allowed in the project.

attributes A semicolon-separated list of name-value pairs. *name* is an arbitrary string specifying the object-related attribute, and *value* is the optional value for that attribute.

```
name[=value]
```

In the name-value pair, names are restricted to letters, digits, underscores, and the period. The period is conventionally used as a separator between the categories and subcategories of the `rctl`. The first character of an attribute name must be a letter. The name is case-sensitive.

Values can be structured using commas and parentheses to establish precedence. The semicolon is used to separate name-value pairs and cannot be used in a value definition. The colon is used to separate project fields and cannot be used in a value definition.

Note – Routines that read this file halt when they encounter a malformed entry, so any project assignments specified after the entry are not made.

This example shows the default `/etc/project` file:

```
system:0:System:::
user.root:1:Super-User:::
noproject:2:No Project:::
default:3::::
group.staff:10::::
```

This example shows the default `/etc/project` file with project entries added at the end:

```
system:0:System:::
user.root:1:Super-User:::
noproject:2:No Project:::
default:3::::
group.staff:10::::
user.ml:2424:Lyle Personal:::
booksite:4113:Book Auction Project:ml,mp,jtd,kjh::
```

To add resource controls to the `/etc/project` file, see "Using Resource Controls" on page 93.

Name Service Configuration for NIS

If you are using NIS, you can specify in the `/etc/nsswitch.conf` file to search the NIS maps for projects:

```
project: nis files
```

The NIS map, either `project.byname` or `project.bynumber`, has the same form as the `/etc/project` file:

```
projname:projid:comment:user-list:group-list:attributes
```

For more information, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Directory Service Configuration for LDAP

If you are using LDAP, you can specify in the `/etc/nsswitch.conf` file to search the LDAP entries for projects.

```
project: ldap files
```

For more information, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Tasks

With each successful login into a project, a new *task* is created that contains the login process. The task is a process collective that represents a set of work over time. A task can also be viewed as a *workload component*.

Each process is a member of one task, and each task is associated with one project.

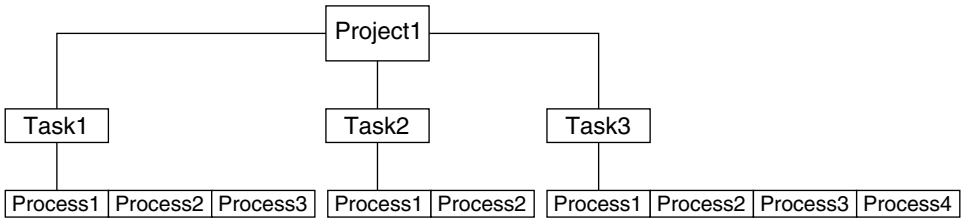


FIGURE 6–2 Project and Task Tree

All operations on sessions, such as signal delivery, are also supported on tasks. You can also bind tasks to processor sets and set their scheduling priorities and classes, which modifies all current and subsequent processes in the task.

Tasks are created at login (see `login`(1)), by `cron`(1M), or by using the `newtask`(1) command.

The extended accounting facility can provide accounting data that is aggregated at the task level.

Commands Used to Administer Projects and Tasks

Command	Description
`projects`(1)	Prints the project membership of a user.
`newtask`(1)	Executes the user's default shell or specified command, placing the execution command in a new task owned by the specified project.
`projadd`(1M)	Adds a new project entry to the `/etc/project` file. `projadd` creates a project entry only on the local system. `projadd` cannot change information supplied by the network name service.
`projmod`(1M)	Modifies a project's information on the local system. `projmod` cannot change information supplied by the network name service. However, the command does verify the uniqueness of the project name and project ID against the external name service.
`projdel`(1M)	Deletes a project from the local system. `projdel` cannot change information supplied by the network name service.

Command Options Used With Projects and Tasks

ps

Use `ps -o` to display task and project IDs. For example, to view the project ID, type:

```
# ps -o user,pid,uid,projid
USER PID   UID  PROJID
jtd  89430 124  4113
```

id

Use `id -p` to print the current project ID in addition to the user and group IDs. If the *user* operand is provided, the project associated with that user's normal login is printed:

```
#  id -p
uid=124(jtd) gid=10(staff) projid=4113(booksite)
```

pgrep and pkill

To match only processes with a project ID in the given list, type:

```
# pgrep -J projidlist
# pkill -J projidlist
```

To match only processes with a task ID in the given list, type:

```
# pgrep -T taskidlist
# pkill -T taskidlist
```

prstat

To display various statistics for processes and projects currently running on your system, type:

```
% prstat -J
      PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
 21634 jtd      5512K 4848K cpu0    44    0   0:00.00 0.3% prstat/1
   324 root       29M   75M sleep   59    0   0:08.27 0.2% Xsun/1
 15497 jtd        48M   41M sleep   49    0   0:08.26 0.1% adeptedit/1
   328 root     2856K 2600K sleep   58    0   0:00.00 0.0% mibiisa/11
  1979 jtd      1568K 1352K sleep   49    0   0:00.00 0.0% csh/1
  1977 jtd      7256K 5512K sleep   49    0   0:00.00 0.0% dtterm/1
   192 root     3680K 2856K sleep   58    0   0:00.36 0.0% automountd/5
  1845 jtd        24M   22M sleep   49    0   0:00.29 0.0% dtmail/11
  1009 jtd      9864K 8384K sleep   49    0   0:00.59 0.0% dtwm/8
   114 root     1640K  704K sleep   58    0   0:01.16 0.0% in.routed/1
   180 daemon   2704K 1944K sleep   58    0   0:00.00 0.0% statd/4
   145 root     2120K 1520K sleep   58    0   0:00.00 0.0% ypbind/1
   181 root     1864K 1336K sleep   51    0   0:00.00 0.0% lockd/1
   173 root     2584K 2136K sleep   58    0   0:00.00 0.0% inetd/1
   135 root     2960K 1424K sleep    0    0   0:00.00 0.0% keyserv/4
PROJID    NPROC  SIZE   RSS MEMORY      TIME  CPU PROJECT
    10       52  400M  271M    68%   0:11.45 0.4% booksite
     0       35  113M  129M    32%   0:10.46 0.2% system

Total: 87 processes, 205 lwps, load averages: 0.05, 0.02, 0.02
```

To display various statistics for tasks currently running on your system, type:

```
% prstat -T
```

Using cron and su With Projects and Tasks

cron

The cron command issues a settaskid to ensure that each cron, at, and batch job executes in a separate job, with the appropriate default project for the submitting user. Also, the at and batch commands capture the current project ID and ensure that the project ID is restored when running an at job.

su

To switch the user's default project (and thus create a new task) as part of simulating a login, type:

```
# su - user
```

To retain the project ID of the invoker, issue su without the - flag:

```
# su user
```

Project Administration Examples

▼ How to Define a Project and View the Current Project

This example shows how to use the projadd and projmod commands.

1. **Become superuser.**
2. **Look at the default** /etc/project **file on your system.**

```
# cat /etc/project
system:0::::
user.root:1::::
noproject:2::::
```

```
default:3::::
group.staff:10::::
```

3. **Add a project called** *booksite* **and assign it to a user named** *mark* **with project ID number** *4113***.**

```
# projadd -U mark -p 4113 booksite
```

4. **Look at the** `/etc/project` **file again to see the project addition.**

```
# cat /etc/project
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
booksite:4113::mark::
```

5. **Add a comment that describes the project in the comment field.**

```
# projmod -c 'Book Auction Project' booksite
```

6. **Look at the** `/etc/project` **file to view the changes.**

```
# cat /etc/project
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
booksite:4113:Book Auction Project:mark::
```

▼ How to Delete a Project From the `/etc/project` File

This example shows how to use the `projdel` command to delete a project.

1. **Become superuser.**

2. **Remove the project** *booksite* **by using the** `projdel` **command.**

```
# projdel booksite
```

3. **Display the** `/etc/project` **file.**

```
# cat /etc/project
system:0::::
user.root:1::::
noproject:2::::
default:3::::
group.staff:10::::
```

4. **Log in as user *mark* and type `projects` to view the projects assigned.**

```
# su - mark
# projects
default
```

▼ How to Obtain User and Project Membership Information

Use the `id` command with the `-p` flag to view the current project membership of the invoking process.

```
$ id -p
uid=100(mark) gid=1(other) projid=3(default)
```

▼ How to Create a New Task

1. **Become superuser.**
2. **Create a new task in the *booksite* project by using the `newtask` command with the `-v` (verbose) option to obtain the system task ID.**

```
# newtask -v -p booksite
16
```

3. **View the current project membership of the invoking process.**

```
# id -p
uid=100(mark) gid=1(other) projid=4113(booksite)
```

Note that the process is now a member of the new project.

CHAPTER 7

Extended Accounting

If workloads are labelled and separated using the project and task facilities described in Chapter 6, you can monitor resource consumption by workloads running on the system. You can use the *extended accounting* subsystem to capture a detailed set of resource consumption statistics on both processes and tasks, and label the usage records with the project for which the work was done.

To begin using extended accounting, see "How to Activate Extended Accounting for Processes and Tasks" on page 84.

Overview

Before you can apply resource management mechanisms, you must first be able to characterize the resource consumption demands that various workloads place on a system. The extended accounting facility in the Solaris operating environment provides a flexible way to record resource consumption on a task or process basis. In contrast to online monitoring tools, which permit measurement of system usage in real time, extended accounting allows you to examine historical usage and make assessments of capacity requirements for future workloads.

With extended accounting data available, you can develop or purchase software for resource chargeback, workload monitoring, or capacity planning.

How Extended Accounting Works

The extended accounting facility in the Solaris environment uses a versioned, extensible file format to contain accounting data. Files using this data format can be accessed or created using the API provided in the included library, `libexacct`. These files can then be analyzed on any platform with extended accounting enabled, and their data can be used for capacity planning and chargeback.

If extended accounting is active, statistics are gathered that can be examined by the `libexacct` API. `libexacct` allows examination of the `exacct` files either forward or backward. The API supports third-party files generated by `libexacct` as well as those files created by the kernel.

With extended accounting enabled, the task tracks the aggregate resource usage of its member processes. A task accounting record is written at task completion. For more information on tasks, see Chapter 6.

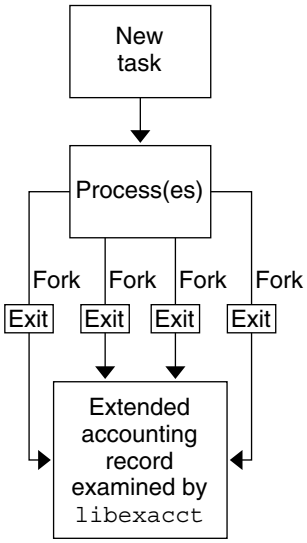


FIGURE 7–1 Task Tracking With Extended Accounting Activated

Extensible Format

The extended accounting format is substantially more extensible than the SunOS™ legacy system accounting software format (see "What is System Accounting?" in *System Administration Guide: Advanced Administration*). Extended accounting permits

accounting metrics to be added and removed from the system between releases, and even during system operation.

Note – Both extended accounting and legacy system accounting software can be active on your system at the same time.

`exacct` Records and Format

Routines that allow `exacct` records to be created serve two purposes.

- To enable third-party `exacct` files to be created
- To enable the creation of tagging records to be embedded in the kernel accounting file by using the `putacct` system call (see `getacct`(2)).

The format permits different forms of accounting records to be captured without requiring that every change be an explicit version change. Well-written applications that consume accounting data must ignore records they do not understand.

The `libexacct` library converts and produces files in the `exacct` format. This library is the *only* supported interface to `exacct` format files.

Extended Accounting Configuration

The `/etc/acctadm.conf` file contains the current extended accounting configuration. The file is edited through the `acctadm` interface, not by the user.

The directory `/var/adm/exacct` is the standard location for placing extended accounting data. You can use the `acctadm`(1M) command to specify a different location for the process and task accounting data files.

Commands Used With Extended Accounting

Command	Description
`acctadm`(1M)	Modifies various attributes of the extended accounting facility, stops and starts extended accounting, and is used to select accounting attributes to track for processes and tasks.
`wracct`(1M)	Writes extended accounting activity for active processes and tasks.
`lastcomm`(1)	Displays previously invoked commands. `lastcomm` can consume either standard accounting process data or extended accounting process data.

For information on commands associated with tasks and projects, see "Commands Used to Administer Projects and Tasks" on page 75.

Using Extended Accounting Functionality

▼ How to Activate Extended Accounting for Processes and Tasks

To activate the extended accounting facility for tasks and processes, use the `acctadm`(1M) command. The optional final parameter to `acctadm` indicates whether the command should act on the process or the system task accounting components of the extended account facility.

1. **Become superuser.**
2. **Activate extended accounting for processes.**

```
# acctadm -e extended -f /var/adm/exacct/proc process
```

3. **Activate extended accounting for tasks.**

```
# acctadm -e extended,mstate -f /var/adm/exacct/task task
```

▼ How to Activate Extended Accounting With a Startup Script

Activate extended accounting on an ongoing basis by linking the `/etc/init.d/acctadm` script into `/etc/rc2.d`.

```
# ln -s /etc/init.d/acctadm /etc/rc2.d/Snacctadm
# ln -s /etc/init.d/acctadm /etc/rc2.d/Knacctadm
```

The *n* variable is replaced by a number.

See "Extended Accounting Configuration" on page 83 for information on accounting configuration.

▼ How to Display Extended Accounting Status

Type `acctadm` without arguments to display the current status of the extended accounting facility.

```
# acctadm
                Task accounting: active
           Task accounting file: /var/adm/exacct/task
         Tracked task resources: extended,mstate
       Untracked task resources: host
             Process accounting: active
        Process accounting file: /var/adm/exacct/proc
      Tracked process resources: extended
    Untracked process resources: host,mstate
```

In the above example, system task accounting is active in extended mode and `mstate` mode. Process accounting is active in extended mode.

Note – In the context of extended accounting, microstate (`mstate`) refers to the extended data, associated with microstate process transitions, that are available in the process usage file (see `proc`(4)). This data provides much more detail about the activities of the process than basic or extended records.

▼ How to View Available Accounting Resources

Available resources can vary from system to system, and from platform to platform. Use the `-r` option to view the available accounting resources on the system.

```
# acctadm -r
process:
extended pid,uid,gid,cpu,time,command,tty,projid,taskid,ancpid,wait-status,flag
basic    pid,uid,gid,cpu,time,command,tty,flag
task:
extended taskid,projid,cpu,time,host,mstate,anctaskid
basic    taskid,projid,cpu,time
```

▼ How to Deactivate Process and Task Accounting

To deactivate process and task accounting, turn each of them off individually.

1. **Become superuser.**
2. **Turn off process accounting.**

   ```
   # acctadm -x process
   ```

3. **Turn off task accounting.**

   ```
   # acctadm -x task
   ```

4. **Verify that task accounting and process accounting have been turned off.**

   ```
   # acctadm
                     Task accounting: inactive
                Task accounting file: none
              Tracked task resources: extended,mstate
            Untracked task resources: host
                  Process accounting: inactive
             Process accounting file: none
           Tracked process resources: extended
         Untracked process resources: host,mstate
   ```

CHAPTER 8

Resource Controls

After you determine the resource consumption of workloads on your system using the functionality described in Chapter 7, you can place bounds on resource usage and prevent workloads from over-consuming resources. The resource controls facility, which extends the UNIX resource limit concept, is the constraint mechanism used for this purpose.

Overview

UNIX systems have traditionally provided a resource limit facility (*rlimit*). The rlimit facility allows administrators to set one or more numerical limits on the amount of resources a process can consume. These limits include per-process CPU time used, per-process core file size, and per-process maximum heap size. Heap size is the amount of memory allocated for the process data segment.

In the Solaris operating environment, the concept of a per-process resource limit has been extended to the task and project entities described in Chapter 6. These extended limits can now be observed on a system-wide basis. These enhancements are provided by the *resource controls* (*rctls*) facility.

The resource controls facility provides compatibility interfaces for the resource limits facility. Existing applications utilizing resource limits continue to run unchanged, and they can be observed in the same way as applications modified to take advantage of the resource controls facility.

Resource controls provide a mechanism for constraint on system resources. Processes, tasks, and projects can be prevented from consuming amounts of specified system resources. This mechanism leads to a more manageable system by preventing over-consumption of resources.

Constraint mechanisms can be used to support of capacity planning processes. An encountered constraint can provide information about application resource needs without necessarily denying the resource to the application.

Resource controls can also serve as a simple attribute mechanism for resource management facilities. For example, the number of CPU shares made available to a project in the fair share scheduler (FSS) scheduling class is defined by the `project.cpu-shares` resource control. Because the project is assigned a fixed number of shares by the control, the various actions associated with exceeding a control are not relevant. In this sense, the current value for the `project.cpu-shares` control is considered to be an attribute on the given project.

Administering Resource Controls

The resource controls facility is configured through the project database (see Chapter 6). The `rctladm`(1M) command allows you to make runtime interrogations of the resource controls facility. The `prctl`(1) command allows you to make runtime modifications to the resource controls facility.

Available Resource Controls

A list of the standard resource controls available in this release is shown in the following table.

The table describes the resource constrained by each control and identifies the default units used by the `project` database for that resource. The default units are of two types:

- Quantities represent a limited amount.
- Indexes represent a maximum valid identifier.

Thus, `project.cpu-shares` specifies the number of shares the project is entitled to, while `process.max-file-descriptor` specifies the highest file number that can be assigned to a process by the `open`(2) system call.

TABLE 8–1 Standard Resource Controls

Control Name	Description	Default Unit
`project.cpu-shares`	The number of CPU shares granted to this project for use with `FSS`(7)	Quantity (shares)

TABLE 8–1 Standard Resource Controls *(Continued)*

Control Name	Description	Default Unit
`task.max-cpu-time`	Maximum CPU time available to this task's processes	Time (milliseconds)
`task.max-lwps`	Maximum number of LWPs simultaneously available to this task's processes	Quantity (LWPs)
`process.max-cpu-time`	Maximum CPU time available to this process	Time (milliseconds)
`process.max-file-descriptor`	Maximum file descriptor index available to this process	Index (maximum file descriptor)
`process.max-file-size`	Maximum file offset available for writing by this process	Size (bytes)
`process.max-core-size`	Maximum size of a core file created by this process	Size (bytes)
`process.max-data-size`	Maximum heap memory available to this process	Size (bytes)
`process.max-stack-size`	Maximum stack memory segment available to this process	Size (bytes)
`process.max-address-space`	Maximum amount of address space, as summed over segment sizes, available to this process	Size (bytes)

Actions on Resource Control Values

For each threshold value placed on a resource control, you can associate one or more actions.

- You can choose to deny the resource requests for an amount greater than the threshold.
- You can choose to send a signal to the violating or observing process if the threshold value is reached.

Due to implementation restrictions, the global properties of each control can restrict the set of available actions that can be set on the threshold value. A list of available signal actions is presented in the following table. For additional information on signals, see signal(3HEAD).

TABLE 8–2 Signals Available to Resource Control Values

Signal	Notes
SIGABRT	
SIGHUP	
SIGTERM	
SIGKILL	
SIGSTOP	
SIGXRES	
SIGXFSZ	Available only to resource controls with the RCTL_GLOBAL_FILE_SIZE attribute set
SIGXCPU	Available only to resource controls with the RCTL_GLOBAL_CPUTIME attribute set

Resource Control Enforcement

More than one resource control can exist on a resource, one at each containment level in the process model. If resource controls are active on the same resource at different container levels, the smallest container's control is enforced first. Thus, action is taken

on process.max-cpu-time before task.max-cpu-time if both controls are encountered simultaneously.

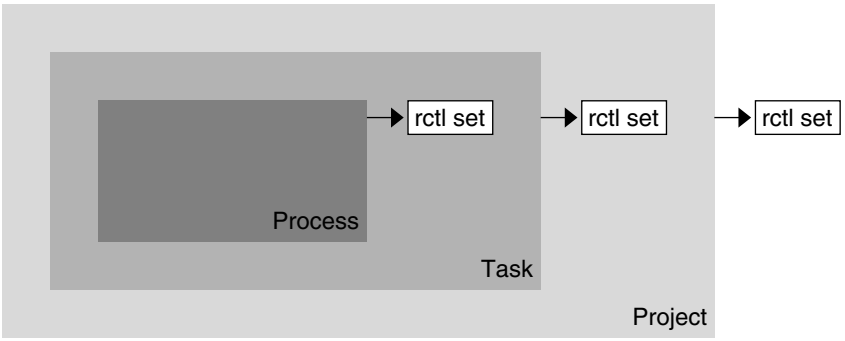


FIGURE 8–1 Process Collectives, Container Relationships, and Their Resource Control Sets

Global Monitoring of Resource Control Events

In cases where the resource consumption of processes is unknown, it can be helpful to use the global resource control actions available with rctladm(1M). When a syslog action has been established on a resource control, any occurrence of a threshold value being encountered by any entity managed by that resource control results in a system message being logged at the configured logging level.

Configuration

Each resource control listed in Table 8–1 can be assigned to a project at login or when newtask(1) or the other project-aware launchers at(1), batch (see at(1)), or cron(1M) are invoked. Each command initiated is launched in a separate task with the invoking user's default project.

Updates to entries in the project database, whether to the /etc/project file or to a representation of the database in a network name service, are not applied to currently active projects. The updates are applied when a project is restarted using login(1) or newtask.

Temporarily Updating Resource Control Values on a Running System

Although values changed in the `project` database only take effect when the project is restarted, you can use the `rctladm` and `prctl` commands to update resource controls on a running process.

Updating Logging Status

The `rctladm` command affects the global logging state of each resource control on a system-wide basis. This command can be used to view the global state and to set up the level of `syslog` logging when controls are exceeded.

Updating Resource Controls

You can view and temporarily alter resource control values and actions on a per-project basis by using `prctl`. A project, task, or process ID is given as input, and the command operates on the resource control at the level where it is defined.

Any modifications to values and actions take effect immediately. However, these modifications apply to the current session only. The changes are not recorded in the `project` database. If the system is restarted, the modifications are lost. Permanent changes to resource controls must be made in the `project` database.

All resource control settings that can be modified in the `project` database can also be modified with the `prctl` command. Both basic and privileged values can be added or deleted and their actions modified. By default, the basic type is assumed for all set operations, but superuser-equivalent invocations can also modify privileged resource controls. System resource controls cannot be altered.

Using Resource Controls

▼ How to Set the Maximum Number of LWPs for Each Task in a Project

Type this entry in the `/etc/project` database to set the maximum number of LWPs in each task in project *x-files* to *3*.

```
x-files:100::root::task.max-lwps=(privileged,3,deny)
```

When superuser creates a new task in project *x-files* by joining it with `newtask`, superuser will not be able to create more than three LWPs while running in this task, as shown in the following annotated sample session:

```
# newtask -p x-files csh

# prctl -n task.max-lwps $$
688: csh
task.max-lwps
                           3 privileged deny
                  2147483647 system     deny
# id -p
uid=0(root) gid=1(other) projid=100(x-files)

# ps -o project,taskid -p $$
 PROJECT TASKID
 x-files   236

# csh        /* creates second LWP */

# csh        /* creates third LWP */

# csh        /* cannot create more LWPs */
Vfork failed

#
```

▼ How to Set a `basic` Control

The `/etc/project` file can also contain settings for multiple resource controls per project. The following line in the file sets a `basic` control with no action on the maximum LWPs per task for project *x-files*. The line also sets a privileged `deny` control on the maximum LWPs per task. This control causes any LWP creation that exceeds

the maximum to fail, as shown in the previous example. Finally, the maximum file descriptors per process are limited at the `basic` level, which will force failure of any open call that exceeds the maximum.

```
x-files:101::root::task.max-lwps=(basic,10,none),(privileged,500,deny);\
    process.max-file-descriptor=(basic,128,deny)
```

▼ How to Use `prctl`

As superuser, type `prctl` to display the maximum file descriptor for the current running shell:

```
# prctl -n process.max-file-descriptor $$
8437:   sh
process.max-file-descriptor              [ lowerable deny ]
                           256 basic      deny
                         65536 privileged deny
                    2147483647 system     deny
```

Use the `prctl` command to temporarily add a new privileged value to deny the use of more than three LWPs per task for the *x-files* project. The result is identical to the result in "How to Set the Maximum Number of LWPs for Each Task in a Project" on page 93, as shown in the following annotated sample session:

```
# newtask -p x-files

# id -p
uid=0(root) gid=1(other) projid=101(x-files)

# prctl -n task.max-lwps -t privileged -v 3 -e deny -i project x-files

# prctl -n task.max-lwps -i project x-files
670:    sh
task.max-lwps
                        3 privileged deny
                2147483647 system     deny
```

You can also use `prctl -r` to change the lowest value of a resource control. Type:

```
# prctl -n process.max-file-descriptor -r -v 128 $$
```

▼ How to Use `rctladm`

You can use `rctladm` to enable the global `syslog` attribute of a resource control. When the control is exceeded, notification will be logged at the specified `syslog` level. Type:

```
# rctladm -e syslog process.max-file-descriptor
```

Capacity Warnings

A global action on a resource control enables you to receive notice of any entity tripping over a resource control value.

For example, assume you want to determine whether a web server possesses sufficient CPUs for its typical workload. You could do this by analyzing `sar`(1) data for idle CPU time and load average, or by examining extended accounting data to determine the number of simultaneous processes running for the web server process.

However, an easier approach is to place the web server in a task and then set a global action, using `syslog`, to notify you whenever a task exceeds a scheduled number of LWPs appropriate for the machine's capabilities.

▼ How to Determine Whether a Web Server Is Allocated Enough CPUs for Its Workload

1. **Use the `prctl` command to place a privileged (superuser-owned) resource control on the tasks that contain an `httpd` process. Limit each task's total number of LWPs to 40, and disable all local actions:**

   ```
   # prctl -n task.max-lwps -v 40 -t privileged -d all `pgrep httpd`
   ```

2. **Enable a system log global action on the `task.max-lwps` resource control:**

   ```
   # rctladm -e syslog task.max-lwps
   ```

3. **Observe whether the workload trips the resource control.**

   If it does, you will see `/var/adm/messages` such as:

   ```
   Jan  8 10:15:15 testmachine unix: [ID 859581 kern.notice]
   NOTICE: privileged rctl task.max-lwps exceeded by task 19
   ```

CHAPTER 9

Fair Share Scheduler

If an analysis of workload data indicates that CPU resources are being monopolized by a particular workload or group of workloads, but not violating resource constraints on CPU usage, it is possible to modify the allocation policy for CPU time on the system. The fair share scheduler scheduling class described in this chapter allows you to allocate CPU time based on shares, rather than on the priority scheme of the timesharing (TS) scheduling class.

Overview

A fundamental job of the operating system is to arbitrate which processes get access to the system's resources. The process scheduler (also called the dispatcher) is the portion of the kernel that controls allocation of the CPU to processes. The scheduler supports the concept of scheduling classes, where each class defines a scheduling policy used to schedule processes within the class. The default scheduler in the Solaris operating environment, the TS scheduler, tries to give every process relatively equal access to the available CPUs. However, in some cases you want to specify that certain processes be given more resources than others.

You can use the *fair share scheduler* (FSS) to control the allocation of available CPU resources among workloads based on their importance. This importance is expressed by the number of *shares* of CPU resources that you assign to each workload.

You give each project CPU shares to control the project's entitlement to CPU resources. The FSS guarantees the fair dispersion of CPU resources among projects based on allocated shares, independent of the number of processes attached to a project. Fairness is achieved by reducing a project's entitlement for heavy CPU usage and increasing its entitlement for light usage, with respect to other projects.

The FSS consists of a kernel scheduling class module and class-specific versions of the `dispadmin`(1M) and `priocntl`(1) commands. Project shares used by the FSS are specified through the `project.cpu-shares` property in the `project`(4) database.

CPU Share Definition

The term "share" is used to define a portion of the system's CPU resources allocated to a project. The larger the number of CPU shares assigned to a project, the more CPU resources it receives from the fair share scheduler, relative to other projects.

CPU shares are not equivalent to percentages of CPU resources. Shares are used to define the relative importance of workloads in relation to other workloads. When you assign CPU shares to a project, what matters is not how many shares the project has, but how many shares it has compared to other projects, and how many of those other projects will be competing with it for CPU resources.

Note – Processes in projects with zero shares always run at the lowest system priority (0). These processes only run when projects with non-zero shares are not using CPU resources.

CPU Shares and Process State

In the Solaris operating environment, a project workload usually consists of more than one process. From the fair share scheduler perspective, each project workload can be in either an *idle* state or an *active* state. A project is considered idle if none of its processes are using any CPU resources. This usually means that such processes are either *sleeping* (waiting for I/O completion) or stopped. A project is considered active if at least one of its processes is using CPU resources. The sum of shares of all active projects is used in calculating the portion of CPU resources to be assigned to projects.

The following formula shows how the FSS scheduler calculates per-project allocation of CPU resources.

$$\text{allocation}_{\text{project}_i} = \frac{\text{shares}_{\text{project}_i}}{\sum_{j=1...n} (\text{shares}_{\text{project}_j})}$$

j is the index among all active projects

FIGURE 9–1 FSS Scheduler Share Calculation

When more projects become active, each project's CPU allocation gets smaller, but the proportion does not change.

CPU Share Versus Utilization

In the absence of workload competition, share allocation is not the same as utilization. A project allocated 50 percent of the CPU resources might average only a 20 percent CPU use. Shares serve to cap CPU usage only when there is competition from other projects. When there is no competition, utilizations by some projects can be larger than their shares. Available CPU cycles are never wasted; they are distributed between projects. No matter how low a project's allocation, it is always given 100 percent of the processing power if it is running alone on the system and there are no competing projects.

Allocating a small share to a busy workload slows it down, but does not prevent it from completing its work as long as the system is not overloaded.

CPU Share Examples

Assume you have a system with two CPUs running two parallel CPU-bound workloads called *A* and *B*, respectively. Each workload is running as a separate project. The projects have been configured so that project *A* is assigned S_A shares, and project *B* is assigned S_B shares.

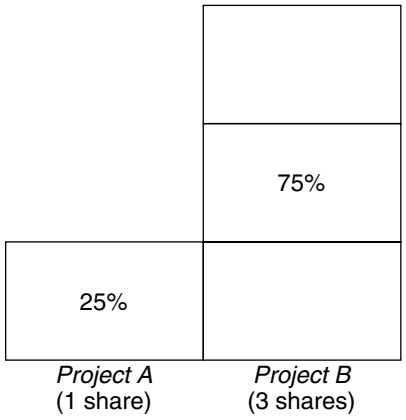
On average, under the traditional TS scheduler, each of these projects would be given the same amount of CPU resources, which would be 50 percent of the system's capacity to each project.

When run under the control of the FSS scheduler with S_A=S_B, these projects will also be given approximately the same amounts of CPU resources. However, if projects are given different numbers of shares, their CPU resource allocations will be different.

The next three examples illustrate how shares work in different configurations. These examples show that shares are only mathematically accurate for representing the usage if demand exceeds available resources.

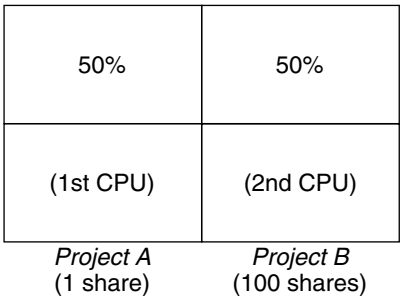
Example 1: Two CPU-Bound Processes in Each Project

If *A* and *B* have two CPU-bound processes each, and $S_A = 1$ and $S_B = 3$, then the total number of shares is 1 + 3 = 4. In this configuration, given sufficient CPU demand, projects *A* and *B* are allocated 25 percent and 75 percent of CPU resources, respectively.



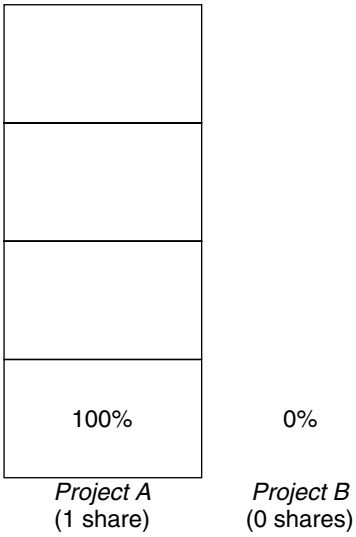
Example 2: No Competition Between Projects

If *A* and *B* have only *one* CPU-bound process each, and $S_A = 1$ and $S_B = 100$, then the total number of shares is 101. Each project cannot use more than one CPU because each project has only one running process. Because there is no competition between projects for CPU resources in this configuration, projects *A* and *B* are each allocated 50 percent of all CPU resources. In this configuration, CPU share values are irrelevant. The projects' allocations would be the same (50/50) even if both projects were assigned zero shares.



Example 3: One Project Unable to Run

If *A* and *B* have two CPU-bound processes each, and project *A* is given 1 share and project *B* is given 0 shares, then project *B* is not allocated any CPU resources and project *A* is allocated all CPU resources. Processes in *B* will always run at system priority 0, so they will never get a chance to run because processes in project *A* will always have higher priorities.



FSS Configuration

Projects and Users

Projects are the smallest workload containers in the FSS scheduler. Groups of users assigned to a project are treated as single controllable blocks. Note that you can create a project with its own number of shares for an individual user if desired.

Users can be members of multiple projects that have different numbers of shares assigned to them. By moving processes from one project to another, processes can be assigned varying amounts of CPU resources.

For more information on the `project`(4) database and name services, see "`project` Database" on page 71.

CPU Shares Configuration

The configuration of CPU shares is managed by the name service, as a property of the `project` database.

When a project is created through the `setproject`(3PROJECT) library function, the number of CPU shares defined as resource control `project.cpu-shares` in the `project` database is passed to the kernel. Projects that do not have the `project.cpu-shares` resource control defined are assigned one share each.

In the following example, this entry in the `/etc/project` file sets the number of shares for project *x-files* to *5*:

```
x-files:100::::project.cpu-shares=(privileged,5,none)
```

If the number of CPU shares allocated to a project is altered in the database when processes are already running, the number of shares for that project will not be modified at that point. The project must be restarted for the change to take effect.

If you want to temporarily change the number of shares assigned to a project without altering the project's attributes in the `project` database, use `prctl`(1). For example, to change the value of a project *x-files*'s `project.cpu-shares` resource control to *3* while processes associated with that project are running, type:

```
# prctl -r -n project.cpu-shares -v 3 -i project x-files
```

`-r` Replaces the current value for the named resource control.

`-n` *name*	Specifies the name of the resource control.
`-v` *val*	Specifies the value for the resource control.
`-i` *idtype*	Specifies the ID type of the next argument.
x-files	Specifies the object of the change. In this case, project *x-files* is the object.

Project `system` with project ID 0, which includes all system daemons started by the boot-time initialization scripts, can be viewed as a project with an unlimited number of shares. This means that it is always scheduled first no matter how many shares have been given to other projects.

As stated earlier, processes that belong to projects with zero shares are always given zero system priority. Projects with one or more shares are running with priorities one and above. Thus, projects with zero shares are only scheduled when there are available CPU resources that are not requested by a non-zero share project.

The maximum number of shares that can be assigned to one project is 65535.

FSS and Processor Sets

The FSS can be used in conjunction with processor sets to provide more fine-grained controls over allocations of CPU resources among projects running on each processor set than would be available with processor sets alone. The FSS scheduler treats processor sets as entirely independent partitions, with each processor set controlled independently with respect to CPU allocations.

The CPU allocations of projects running in one processor set are not affected by the CPU shares or activity of projects running in another processor set because they are not competing for the same resources. Projects only compete with each other if they are running within the same processor set.

The number of shares allocated to a project is system-wide. No matter which processor set it is running on, each portion of a project is given the same amount of shares.

When processor sets are used, project CPU allocations are calculated for active projects running within each processor set, as shown in the following figure.

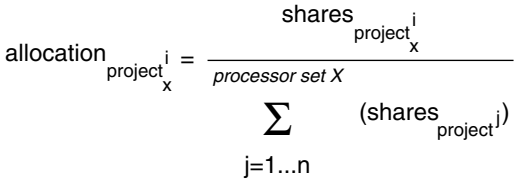


FIGURE 9–2 FSS Scheduler Share Calculation With Processor Sets

Project partitions running on different processor sets might have different CPU allocations. The CPU allocation for each project partition in a processor set depends only on the allocations of other projects running on the same processor set.

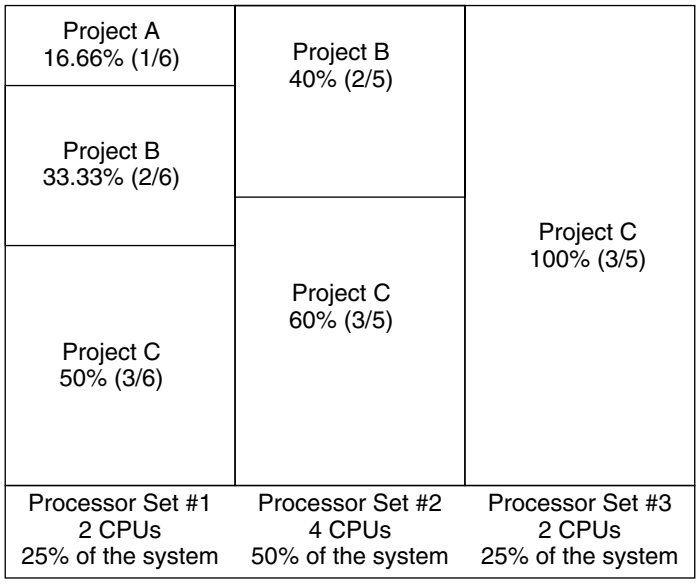
The performance and availability of applications running within the boundaries of their processor sets are not affected by the introduction of new processor sets or changes made to the share allocations of projects running on other processor sets.

Empty processor sets (sets without processors in them) or processor sets without processes bound to them do not have any impact on the FSS scheduler behavior.

FSS and Processor Sets Examples

Assume that a server with eight CPUs is running several CPU-bound applications in projects *A*, *B*, and *C*. Project *A* is allocated one share, project *B* is allocated two shares, and project *C* is allocated three shares.

Project *A* is running only on processor set 1. Project *B* is running on processor sets 1 and 2. Assume that each project has enough processes to utilize all available CPU power, so there is always competition for CPU resources on each processor set.



The total system-wide project CPU allocations on such a system are:

Project A	$4\% = (1/6 \text{ X } 2/8)_{pset1}$
Project B	$28\% = (2/6 \text{ X } 2/8)_{pset1} + (2/5 * 4/8)_{pset2}$
Project C	$67\% = (3/6 \text{ X } 2/8)_{pset1} + (3/5 \text{ X } 4/8)_{pset2} + (3/3 \text{ X } 2/8)_{pset3}$

Although these percentages do not match the corresponding amounts of CPU shares given to projects, within each processor set the per-project CPU allocation ratios are proportional to their respective shares.

On the same system *without* processor sets, the distribution of CPU resources would be different:

Project A	16.66% = (1/6)
Project B	33.33% = (2/6)
Project C	50% = (3/6)

Combining FSS With Other Scheduling Classes

By default, the FSS scheduling class uses the same range of priorities (0 to 59) as the timesharing (TS), interactive (IA), and fixed priority (FX) scheduling classes. The FSS class schedules individual LWPs, not whole processes. Thus, a mix of processes in the FSS, TS, IA, and FX classes could result in unexpected scheduling behavior.

With the use of processor sets, you can mix TS, IA, and FX with FSS in one system as long as all the processes running on each processor set are in *one* scheduling class, so they do not compete for the same CPUs. If these classes are mixed in a processor set, FSS scheduling behavior will be most affected if the processor set also has CPU-intensive processes in the TS, IA, or FX scheduling class bound to the same processors.

The Solaris operating environment also offers a real-time (RT) scheduler to users with superuser privileges. By default, the RT scheduling class uses system priorities in a different range (usually from 100 to 119) than FSS. Because RT and FSS are using disjoint ranges of priorities, FSS can coexist with the RT scheduling class within the same processor set. However, the FSS scheduling class will not have any control over processes running in the RT class.

For example, on a four-processor system, a single-threaded RT process can consume one entire processor if the process is CPU-bound. If the system also runs FSS, regular user processes will be competing for the three remaining CPUs that are not being used by the RT process. Note that the RT process might not use the CPU continuously. When it is idle, FSS will utilize all four processors.

To find out which scheduling classes processor sets are running in and make sure that each processor set is configured to run either TS, IA, FX, or FSS processes, type the following command:

```
$ ps -ef -o pset,class | grep -v CLS | sort | uniq
1 FSS
1 SYS
2 TS
2 RT
3 FX
```

To set the default scheduler for the system, see “FSS Configuration Examples” on page 107 and `dispadmin`(1M). To move running processes into a different scheduling class, see “FSS Configuration Examples” on page 107 and `priocntl`(1).

Monitoring the FSS

You can use `prstat`(1M) to monitor CPU usage by active projects.

You can use the extended accounting data for tasks to obtain per-project statistics on the amount of CPU resources consumed over longer periods of time. See Chapter 7 for more information.

▼ How to Monitor System CPU Usage by Projects

To monitor the CPU usage of projects running on the system, type:

```
% prstat -J
```

▼ How to Monitor CPU Usage by Projects in Processor Sets

To monitor the CPU usage of projects on a list of processor sets, where *pset-list* is a list of processor set IDs separated by commas, type:

```
% prstat -J -C pset-list
```

FSS Configuration Examples

Like other scheduling classes in the Solaris environment, commands to set the scheduler class, configure the scheduler's tunable parameters, and configure the properties of individual processes can be used with FSS.

▼ How to Set the Scheduler Class

Use the `dispadmin` command to set FSS as the default scheduler for the system. Type:

```
# dispadmin -d FSS
```

This change will take effect on the next reboot. After reboot, every process on the system will be running in the FSS scheduling class.

▼ How to Manually Move Processes Into the FSS Class

If you do not want to change the default scheduling class and reboot, you can manually move processes from the TS scheduling class into the FSS scheduling class.

1. **Become superuser.**
2. **Move the `init` process (pid 1) into the FSS scheduling class. Type:**

   ```
   # priocntl -s -c FSS -i pid 1
   ```

3. **Move all processes from the TS scheduling class into the FSS scheduling class. Type:**

   ```
   # priocntl -s -c FSS -i class TS
   ```

   All processes will again run in the TS scheduling class after reboot.

▼ How to Move a Project's Processes Into the FSS Class

You can manually move processes in a given project from their current scheduling class to the FSS scheduling class.

1. **Become superuser.**
2. **Move processes running in project ID *10* to the FSS scheduling class. Type:**

   ```
   # priocntl -s -c FSS -i projid 10
   ```

   The project's processes will again run in the TS scheduling class after reboot.

▼ How to Tune Scheduler Parameters

You can use the `dispadmin` command to examine and tune the FSS scheduler's time quantum value. *Time quantum* is the amount of time that a thread is allowed to run before it must relinquish the processor. To display the current time quantum for the FSS scheduler, type:

```
$ dispadmin -c FSS -g
#
# Fair Share Scheduler Configuration
#
RES=1000
#
# Time Quantum
#
QUANTUM=110
```

When you use the `-g` option, you can also use the `-r` option to specify the resolution used for printing time quantum values. If no resolution is specified, time quantum values are displayed in milliseconds by default. Type:

```
$ dispadmin -c FSS -g -r 100
#
# Fair Share Scheduler Configuration
#
RES=100
#
# Time Quantum
#
QUANTUM=11
```

To set scheduling parameters for the FSS scheduling class, use `dispadmin -s`. The values in *file* must be in the format output by the `-g` option. These values overwrite the current values in the kernel. Type:

```
$ dispadmin -c FSS -s file
```

References

For more information on how to use the FSS scheduler, see `priocntl`(1), `ps`(1), `dispadmin`(1M), and `FSS`(7).

CHAPTER 10

Resource Pools

This chapter discusses resource pools, which are used for partitioning machine resources. Resource pools enable you to separate workloads so that workload consumption of certain resources does not overlap. This resource reservation helps to achieve predictable performance on systems with mixed workloads.

Overview

Resource pools provide a persistent configuration mechanism for processor set configuration and scheduling class assignment. By grouping multiple partitions together, pools provide a handle to associate with labeled workloads. Each project entry in the `/etc/project` database can have a pool associated with it. New work begun on a project is bound to the appropriate pool.

The pools mechanism is primarily for use on large machines of more than four CPUs. However, small machines can still benefit from this functionality. On small machines, pools can share noncritical resource partitions, and be separated only on the basis of critical resources.

When to Use Pools

Resource pools are versatile mechanisms that can be applied to many administrative scenarios, as described in the following sections.

Batch Compute Server

Use pools functionality to split a server into two pools.

One pool is used for login sessions and interactive work by timesharing users. The other pool is used for jobs submitted through the batch system.

Application or Database Server

Partition the resources for interactive applications according to the applications' requirements.

Bringing Up Applications in Phases

Set user expectations.

You might initially deploy a machine running only a fraction of the services that the machine is ultimately expected to deliver. User difficulties can occur if reservation-based resource management mechanisms are not established when the machine comes online.

For example, the fair share scheduler optimizes CPU utilization. The response times for a machine running only one application can be misleadingly fast when compared to the response times users will see with multiple applications loaded. By using separate pools for each application, you can ensure that a ceiling on the number of CPUs available to each application is in place before all applications are deployed.

Complex Timesharing Server

Partition a server that supports large user populations.

Server partitioning provides an isolation mechanism that leads to a more predictable per-user response. An example of such an operation is a server that provides support for a group of Sun Ray™ desktops.

By dividing users into groups that bind to separate pools, and using the fair share scheduling (FSS) facility, you can tune CPU allocations to favor sets of users that have priority. This assignment can be based on user role, accounting chargeback, and so forth.

Workloads That Change Seasonally

Use resource pools to adjust to changing demand.

If your site experiences predictable shifts in workload demand over long periods of time, such as a monthly, quarterly, or annual cycles, you can alternate between multiple pools configurations by invoking `pooladm` from a `cron`(1M) job.

Real-Time Applications

Create a real-time pool using the RT scheduler and designated processor resources.

Administering Pools

The commands shown in the following table provide the primary administrative interface to the pools facility.

Command	Description
`pooladm`(1M)	Activates a given configuration or deactivates the current configuration. If run without options, `pooladm` prints out the current running pools configuration.
`poolbind`(1M)	Enables the manual binding of projects, tasks, and processes to a pool.
`poolcrg`(1M)	Creates and modifies pools configuration files. If run with the `info` subcommand argument to the `-c` option, `poolcfg` displays the current configuration.

A library API is provided by `libpool`(3POOL). The library can be used by programs to manipulate pool configurations.

Pools Framework

The resource pools framework stores its view of the machine in the `/var/run/pool.state` file. This file represents the pools framework's view of the machine. The file also contains information about configured pools and the organization of partitionable resources. Each pool contains a reference to a processor set and a reference to a scheduling class.

Implementing Pools on a System

Pools can be implemented on a system using one of these methods.

1. When the Solaris software boots, an `init` script checks to see whether `/etc/pooladm.conf` exists. If this file is found, then `pooladm` is invoked to make this configuration the active pools configuration. The system creates `/var/run/pool.state` to reflect the organization requested in `/etc/pooladm.conf`, and the machine's resources are partitioned accordingly.
2. When the Solaris environment is up and running, a pools configuration can either be activated (if it is not already present) or modified by using the `pooladm` command. By default, `pooladm` operates on `/etc/pooladm.conf`. However, you can optionally specify an alternate location and file name, and use this file to update the pools configuration.

See "Configuration Example" on page 120 for more information.

Dynamic Reconfiguration Operations and Resource Pools

Dynamic reconfiguration (DR) enables you to reconfigure hardware while the system is running. Because DR affects available resource amounts, the pools facility must be included in these operations.

When a DR operation occurs, the `/var/run/pool.state` file is automatically updated.

If the DR operation would take the pools facility out of its valid configuration, it requires an administrative override. To initiate this action, the pools framework participates in DR through the RCM framework. The `SUNW_pool_rcm.so` module in `/usr/lib/rcm/modules` requires that DR transactions which would compromise the minimums on any resource be blocked, unless the force operation option is specified.

Creating Pools Configurations

The configuration file contains a description of the pools to be created on the system. The file describes the entities and resource types that can be manipulated.

Type	Description
`pset`	A processor set resource.
`sched`	A scheduling class resource.
`pool`	Named collection of resource associations.
`system`	The machine-level entity.

See `poolcfg`(1M) for more information on elements that be manipulated.

There are two ways to create a structured `/etc/pooladm.conf` file.

- You can use `poolcfg` to discover the resources on the current system and place the results in a configuration file.

  This simplifies file construction. All active resources and components on the system that are capable of being manipulated by the pools facility will be recorded. This includes existing processor set configurations and entries for all loaded scheduling classes. You can then modify the configuration to rename the processor sets or to create additional pools if necessary.
- You can use `poolcfg` to create a new pools configuration.

  Use this method when you develop configurations for other machines or when you create configurations that you want to apply to the current machine at a later time.

Use `poolcfg`or `libpool` to edit the pools configuration file. The file should not be edited by hand.

▼ How to Create a Configuration by Discovery

Use the `discover` subcommand argument to the `-c` option of `/usr/sbin/poolcfg` to create the pools configuration file.

1. **Become superuser.**
2. **Type:**

```
# poolcfg -c discover
```

The resulting file, `/etc/pooladm.conf`, will contain any existing processor sets, plus entries for all loaded scheduling classes.

▼ How to Create a New Configuration

Use the `create` subcommand argument to the `-c` option of `/usr/sbin/poolcfg` to create a simple configuration file for a system named `tester`. Note that you must quote subcommand arguments that contain white space.

1. **Become superuser.**
2. **Type:**

```
# poolcfg -c 'create system tester'
```

3. **View the contents of the configuration file in readable form.**

```
# poolcfg -c info
system tester
        int system.version 1
        boolean system.bind-default true
        string system.comment
```

▼ How to Modify a Configuration

To enhance your simple configuration, create a processor set named *batch* and a pool named *batch*. Then join them with an association. Note that you must quote subcommand arguments that contain white space.

1. **Become superuser.**
2. **Create processor set** `batch`**.**

```
# poolcfg -c 'create pset batch (uint pset.min = 2; uint pset.max = 10)'
```

3. **Create pool** `batch`**.**

```
# poolcfg -c 'create pool batch'
```

4. **Join with an association.**

```
# poolcfg -c 'associate pool batch (pset batch)' /tmp/foo
```

5. **Display the edited configuration.**

```
# poolcfg -c info
system tester
        int system.version 1
        boolean system.bind-default true
        string system.comment

        pool batch
                boolean pool.default false
                boolean pool.active true
                int pool.importance 5
                string pool.comment
                pset batch

        pset batch
                int pset.sys_id -2
                string pset.units population
                boolean pset.default true
                uint pset.max 10
                uint pset.min 2
                string pset.comment
                boolean pset.escapable false
                uint pset.load 0
                uint pset.size 0
```

▼ How to Use Command Files With `poolcfg`

`poolcfg -f` can take input from a text file that contains `poolcfg` subcommand arguments to the `-c` option. This technique is appropriate when you want a set of operations to be performed atomically. When processing multiple commands, the configuration is only updated if all of them succeed. For large or complex configurations, this technique can be more useful than per-subcommand invocations.

1. **Create the input file.**

```
$ cat > poolcmds.txt
create system system
create pset batch (int pset.man = 2; int pset.max = 10)
create pool batch
associate pool batch (pset batch)
```

2. **Become superuser.**

3. **Type:**

```
# /usr/sbin/poolcfg -f poolcmds.txt
```

Activating and Deactivating Pools Configurations

Use `pooladm`(1M) to make a given pool configuration active or to remove an active pools configuration.

▼ How to Activate a Pools Configuration

To activate the configuration in the default static configuration file, `/etc/pooladm.conf`, invoke `pooladm` with the `-c` option, "commit configuration."

1. **Become superuser.**
2. **Type:**

```
# /usr/sbin/pooladm -c
```

▼ How to Deactivate a Pools Configuration

To remove the running configuration and all associated resources, such as processor sets, use the `-x` option for "remove configuration."

1. **Become superuser.**
2. **Type:**

```
# /usr/sbin/pooladm -x
```

The `-x` option to `pooladm` removes the dynamic configuration file `/var/adm/pool.state` as well as all resource configurations associated with the dynamic configuration. Thus, the `-x` option provides a mechanism for recovering from a poorly designed pools configuration. All processes will be sharing all of the resources on the machine.

Note – Mixing scheduling classes within one processor set can lead to unpredictable results. If you use `pooladm -x` to recover from a bad configuration, you should then use `priocntl`(1) to move running processes into a different scheduling class.

Binding to a Pool

There are two ways to bind a running process to a pool.

- You can use the `poolbind`(1M) command to bind a specific process to a named resource pool.
- You can use the `project.pool` attribute in the `project`(4) database to identify the pool binding for a new login session or a task launched through `newtask`(1).

▼ How to Bind the Current Shell to a Pool

The following procedure manually binds the current shell to a pool named *ohare*.

1. **Become superuser.**
2. **Type:**

```
# poolbind -p ohare $$
```

▼ How to Bind Processes to a Pool

To bind tasks or projects to a pool, use `poolbind` with the `-i` option. The following example binds all processes in the *airmiles* project to the *laguardia* pool.

1. **Become superuser.**
2. **Type:**

```
# poolbind -i project -p laguardia airmiles
```

▼ How to Use `project` Attributes to Bind New Processes to a Pool

To automatically bind new processes in a project to a pool, add the `project.pool` attribute to each entry in the `project` database.

For example, assume you have a configuration with two pools named `studio` and `backstage`. The `/etc/project` file has the following contents.

```
user.paul:1024::::project.pool=studio
user.george:1024::::project.pool=studio
user.ringo:1024::::project.pool=backstage
passes:1027::paul::project.pool=backstage
```

With this configuration, processes started by user `paul` are bound by default to the `studio` pool.

▼ How to Use `project` Attributes to Bind a Process to a Different Pool

Using the above configuration, user `paul` can modify the pool binding for processes he starts. He can use `newtask` to bind work to the `backstage` pool as well, by launching in the `passes` project.

1. **Launch a process in the `passes` project.**

   ```
   $ newtask -l -p passes
   ```

2. **Verify the pool binding for the process.**

   ```
   $ poolbind -q $$
   process id 6384 : pool 'backstage'
   ```

Configuration Example

The configuration used in this section was generated on a 20–CPU Sun Enterprise™ E6500.

To create the configuration file, the following commands were passed to `poolcfg` in a file, by using the `-f` option.

```
create pset pset_tiny_1 (uint pset.min=1;uint pset.max=1;
string pset.comment="A tiny processor partition.")
```

```
create pset pset_small_1 (uint pset.min=2;uint pset.max=4;
string pset.comment="A small processor partition.")
create pset pset_small_2 (uint pset.min=2;uint pset.max=4;
string pset.comment="A small processor partition.")
create pset pset_large_1 (uint pset.min=4;uint pset.max=8;
string pset.comment="A large processor partition.")
create pset pset_large_2 (uint pset.min=4;uint pset.max=8;
string pset.comment="A large processor partition.")
create pset pset_large_3 (uint pset.min=4;uint pset.max=12;
string pset.comment="A large processor partition.")
create pool pool_db (string pool.comment="This pool is
used by the database driving the main user applications.
The database performs best when provided guaranteed access
to a low number of processors in association with the
timeshare scheduler")
associate pool pool_db (pset pset_small_1; sched TS)
create pool pool_app (string pool.comment="This pool is used
by the main application accessed by our users. The
application is graphical and is supporting a large number
of users. The optimum configuration requires
about 6 processors and use of the interactive scheduler.")
associate pool pool_app (pset pset_large_1; sched IA)
create pool pool_batch (string pool.comment="This pool is
used by the overnight batch programs which manipulate the
data entered during the working day by our users. Since the
users will not be active when this sequence of programs is
executing, we can use the same resource pool as used by
pool_app, however the programs are not interactive and so
we use the timeshare scheduler for better performance")
associate pool pool_batch (pset pset_large_1; sched TS)
create pool pool_sunray (string pool.comment="This pool is
used by sunray users. It is intended for use by users logging
into the sunray project which is configured (via
/etc/project) to be automatically associated with this pool.")
associate pool pool_sunray (pset pset_large_2; sched FSS)
create pool pool_rt (string pool.comment="This pool is used
by real-time users. There is a small real-time application
run by some users and we would like to create a reliable
real-time environment for them to use.")
associate pool pool_rt (pset pset_tiny_1; sched RT)
create pool pool_web (string pool.comment="This pool is
used by web applications, such as the web server. This
system supports a large web-site and we would like to
support as many users as possible without impacting our
internal applications.")
associate pool pool_web (pset pset_large_3; sched TS)
```

The configuration file illustrates how the resources and pools for a host machine named `scale20` are represented in XML .

```
<?xml version="1.0"?>
<!DOCTYPE system PUBLIC "-//Sun Microsystems Inc//DTD Resource Management All//EN"
"file:///usr/share/lib/xml/dtd/rm_pool.dtd.1">
<!--
```

```
Configuration for pools facility. Do NOT edit this file by hand - use poolcfg(1) or
libpool(3POOL) instead.
-->
<system name="scale20" ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_22be7da4"
comment="Discovered by libpool" bind-default="true">
  <res_comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2307443c" type="pset"
  name="pset_default" default="true" max="4294967295" min="1" sys_id="-1"
  units="population">
    <property name="pset.escapable" type="boolean">false</property>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_232577a7" type="cpu" sys_id="0"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_232a4213" type="cpu" sys_id="1"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_232fa972" type="cpu" sys_id="4"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_23337639" type="cpu" sys_id="5"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_233762bf" type="cpu" sys_id="8"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_233b0122" type="cpu" sys_id="9"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_233ea303" type="cpu" sys_id="10"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2342528f" type="cpu" sys_id="11"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2345f386" type="cpu" sys_id="12"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_23498c15" type="cpu" sys_id="13"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_234d3883" type="cpu" sys_id="14"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2350d440" type="cpu" sys_id="15"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_23545a19" type="cpu" sys_id="16"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2357fc5a" type="cpu" sys_id="17"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_235bfc60" type="cpu" sys_id="20"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_235f948e" type="cpu" sys_id="21"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_23637823" type="cpu" sys_id="24"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_236733d8" type="cpu" sys_id="25"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_236b0ae3" type="cpu" sys_id="28"/>
    <comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_236ea5a6" type="cpu" sys_id="29"/>
  </res_comp>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" type="sched" name="TS"
  default="false" units="none" sys_id="1"/>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_24579cc2" type="sched" name="FSS"
  default="true" units="none" sys_id="2"/>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_24d4074d" type="sched" name="RT"
  default="false" units="none" sys_id="3"/>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2555ac85" type="sched" name="IA"
  default="false" units="none" sys_id="4"/>
  <pool ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_257cf551" name="pool_default"
  res="id_809b56e8_0005a1d5_00000001_3bc18181_2307443c
  id_809b56e8_0005a1d5_00000001_3bc18181_24579cc2" importance="5"
  default="true" active="true"/>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_00158dd2" type="pset"
  name="pset_tiny_1" default="false" max="1" min="1" sys_id="-2" units="population"
  comment="A tiny processor partition.">
    <property name="pset.escapable" type="boolean">false</property>
  </res_comp>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_00bf820a" type="pset"
  name="pset_small_1" default="false" max="4" min="2" sys_id="-2" units="population"
  comment="A small processor partition.">
    <property name="pset.escapable" type="boolean">false</property>
  </res_comp>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_0172b4e6" type="pset"
  name="pset_small_2" default="false" max="4" min="2" sys_id="-2" units="population"
```

```
comment="A small processor partition.">
  <property name="pset.escapable" type="boolean">false</property>
</res_comp>
<res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_0231e4bf" type="pset"
name="pset_large_1" default="false" max="8" min="4" sys_id="-2" units="population"
comment="A large processor partition.">
  <property name="pset.escapable" type="boolean">false</property>
</res_comp>
<res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_02faa576" type="pset"
name="pset_large_2" default="false" max="8" min="4" sys_id="-2" units="population"
comment="A large processor partition.">
  <property name="pset.escapable" type="boolean">false</property>
</res_comp>
<res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_03d1afe6" type="pset"
name="pset_large_3" default="false" max="12" min="4" sys_id="-2" units="population"
comment="A large processor partition.">
  <property name="pset.escapable" type="boolean">false</property>
</res_comp>
<pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_04172a48" name="pool_db"
res="id_809b56e8_0005a1d6_00000001_3bc18185_00bf820a
id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" importance="5"
default="false" active="true" comment="This pool is used by the database
driving the main user applications. The database performs best when provided
guaranteed access to a low number of processors in association
with the timeshare scheduler"/>
<pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_07d69316" name="pool_app"
res="id_809b56e8_0005a1d6_00000001_3bc18185_0231e4bf
id_809b56e8_0005a1d5_00000001_3bc18181_2555ac85" importance="5" default="false"
active="true" comment="This pool is used by the main application accessed by our
users. The application is graphical and is supporting a large number of users.
The optimum configuration requires about 6 processors and use of the interactive
scheduler."/>
<pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_0b994bb8" name="pool_batch"
res="id_809b56e8_0005a1d6_00000001_3bc18185_0231e4bf
id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" importance="5" default="false"
active="true" comment="This pool is used by the overnight batch programs
which manipulate the data entered during the working day by our users. Since
the users will not be active when this sequence of programs is executing, we
can use the same resource pool as used by pool_app, however the programs are
not interactive and so we use the timeshare scheduler for better performance"/>
<pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18186_0f6d8045" name="pool_sunray"
res="id_809b56e8_0005a1d6_00000001_3bc18185_02faa576
id_809b56e8_0005a1d5_00000001_3bc18181_24579cc2" importance="5" default="false"
active="true" comment="This pool is used by sunray users. It is intended for use
by users logging into the sunray project which is configured (via /etc/project)
to be automatically associated with this pool."/>
<pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18186_1355cf33" name="pool_rt"
res="id_809b56e8_0005a1d6_00000001_3bc18185_00158dd2
id_809b56e8_0005a1d5_00000001_3bc18181_24d4074d" importance="5" default="false"
active="true" comment="This pool is used by real-time users. There is a small
real-time application run by some users and we would like to create a reliable
real-time environment for them to use."/>
<pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18186_17526c1b" name="pool_web"
res="id_809b56e8_0005a1d6_00000001_3bc18185_03d1afe6
```

```
  id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" importance="5" default="false"
  active="true" comment="This pool is used by web applications, such as the web server.
  This system supports a large web-site and we would like to support as many users as
  possible without impacting our internal applications."/>
</system>
```

The `/var/run/pool.state` file from the machine that was used to create the above configuration file is shown below.

```
<?xml version="1.0"?>
<!DOCTYPE system PUBLIC "-//Sun Microsystems Inc//DTD Resource Management All//EN"
"file:///usr/share/lib/xml/dtd/rm_pool.dtd.1">
<!--
Configuration for pools facility. Do NOT edit this file by hand - use poolcfg(1) or
libpool(3POOL) instead.
-->
<system name="scale20" ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_22be7da4"
comment="Discovered by libpool" bind-default="true" version="1">
  <res_comp ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2307443c" type="pset"
  name="pset_default" default="true" max="4294967295" min="1" sys_id="-1"
  units="population" comment="">
    <property name="pset.escapable" type="boolean">true</property>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a56acc7" type="cpu" sys_id="28"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a5a37fa" type="cpu" sys_id="29"/>
  </res_comp>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" type="sched" name="TS"
  default="false" units="none" sys_id="1" comment=""/>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_24579cc2" type="sched" name="FSS"
  default="true" units="none" sys_id="2" comment=""/>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_24d4074d" type="sched" name="RT"
  default="false" units="none" sys_id="3" comment=""/>
  <res_agg ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_2555ac85" type="sched" name="IA"
  default="false" units="none" sys_id="4" comment=""/>
  <pool ref_id="id_809b56e8_0005a1d5_00000001_3bc18181_257cf551" name="pool_default"
  res="id_809b56e8_0005a1d5_00000001_3bc18181_2307443c
  id_809b56e8_0005a1d5_00000001_3bc18181_24579cc2" importance="5" default="true"
  active="true" comment=""/>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_00158dd2" type="pset"
  name="pset_tiny_1" default="false" max="1" min="1" sys_id="1" units="population"
  comment="A tiny processor partition.">
    <property name="pset.escapable" type="boolean">false</property>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a52e512" type="cpu" sys_id="25"/>
  </res_comp>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_00bf820a" type="pset"
  name="pset_small_1" default="false" max="4" min="2" sys_id="2" units="population"
  comment="A small processor partition.">
    <property name="pset.escapable" type="boolean">false</property>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a4098ef" type="cpu" sys_id="16"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a441114" type="cpu" sys_id="17"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a480471" type="cpu" sys_id="20"/>
  </res_comp>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_0172b4e6" type="pset"
  name="pset_small_2" default="false" max="4" min="2" sys_id="3" units="population"
  comment="A small processor partition.">
```

```
    <property name="pset.escapable" type="boolean">false</property>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a4b975c" type="cpu" sys_id="21"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a4f5dd6" type="cpu" sys_id="24"/>
  </res_comp>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_0231e4bf" type="pset"
  name="pset_large_1" default="false" max="8" min="4" sys_id="4" units="population"
  comment="A large processor partition.">
    <property name="pset.escapable" type="boolean">false</property>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a13d4fd" type="cpu" sys_id="0"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a186771" type="cpu" sys_id="1"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a1c4122" type="cpu" sys_id="4"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a1fc476" type="cpu" sys_id="5"/>
  </res_comp>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_02faa576" type="pset"
  name="pset_large_2" default="false" max="8" min="4" sys_id="5" units="population"
  comment="A large processor partition.">
    <property name="pset.escapable" type="boolean">false</property>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a239fb2" type="cpu" sys_id="8"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a282ee7" type="cpu" sys_id="9"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a2ba951" type="cpu" sys_id="10"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a2f2ab1" type="cpu" sys_id="11"/>
  </res_comp>
  <res_comp ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_03d1afe6" type="pset"
  name="pset_large_3" default="false" max="12" min="4" sys_id="6" units="population"
  comment="A large processor partition.">
    <property name="pset.escapable" type="boolean">false</property>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a32ac21" type="cpu" sys_id="12"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a362118" type="cpu" sys_id="13"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a39a053" type="cpu" sys_id="14"/>
    <comp ref_id="id_809b56e8_0005a1e6_00000001_3bc181df_0a3d151a" type="cpu" sys_id="15"/>
  </res_comp>
  <pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_04172a48" name="pool_db"
  res="id_809b56e8_0005a1d6_00000001_3bc18185_00bf820a
  id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" importance="5" default="false"
  active="true" comment="This pool is used by the database
  driving the main user applications. The database performs best
when provided guaranteed access to a low number of
  processors in association with the timeshare scheduler"/>
  <pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_07d69316" name="pool_app"
  res="id_809b56e8_0005a1d6_00000001_3bc18185_0231e4bf
  id_809b56e8_0005a1d5_00000001_3bc18181_2555ac85" importance="5" default="false"
  active="true" comment="This pool is used by the main application accessed by our users.
  The application is graphical and is supporting a large number of users. The optimum
  configuration requires about 6 processors and use of the interactive scheduler."/>
  <pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18185_0b994bb8" name="pool_batch"
  res="id_809b56e8_0005a1d6_00000001_3bc18185_0231e4bf
  id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" importance="5" default="false"
  active="true" comment="This pool is used by the overnight batch programs which
  manipulate the data entered during the working day by our users. Since the users will not
  be active when this sequence of programs is executing, we can use the same resource
  pool as used by pool_app, however the programs are not interactive and so we
use the timeshare scheduler for better performance"/>
  <pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18186_0f6d8045" name="pool_sunray"
  res="id_809b56e8_0005a1d6_00000001_3bc18185_02faa576
```

```
id_809b56e8_0005a1d5_00000001_3bc18181_24579cc2" importance="5" default="false"
active="true" comment="This pool is used by sunray users. It is intended for use by users
logging into the sunray project which is configured (via /etc/project) to be automatically
associated with this pool."/>
  <pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18186_1355cf33" name="pool_rt"
  res="id_809b56e8_0005a1d6_00000001_3bc18185_00158dd2
  id_809b56e8_0005a1d5_00000001_3bc18181_24d4074d" importance="5" default="false"
  active="true" comment="This pool is used by real-time users. There is a small real-time
  application run by some users and we would like to create a reliable real-time
environment for them to use."/>
  <pool ref_id="id_809b56e8_0005a1d6_00000001_3bc18186_17526c1b" name="pool_web"
  res="id_809b56e8_0005a1d6_00000001_3bc18185_03d1afe6
  id_809b56e8_0005a1d5_00000001_3bc18181_23e065d5" importance="5" default="false"
  active="true" comment="This pool is used by web applications, such as the web server.
  This system supports a large web-site and we would like to support as many users as
  possible without impacting our internal applications."/>
</system>
```

CHAPTER 11

Resource Control Functionality in the Solaris Management Console Tool

This chapter describes the resource control and performance monitoring features in the Solaris Management Console™ tool.

You can use the console to monitor system performance and to enter resource control values for projects, tasks, and processes. The console provides a convenient, secure alternative to the command line interface (CLI) for managing hundreds of configuration parameters spread across a large number of systems. The console's graphical interface supports all experience levels.

Using the Console (Task Map)

Task	Description	For Instructions
Use the console.	Start the Solaris Management Console in a local environment or in a name service or directory service environment. Note that the performance tool is not available in a name service environment.	"Starting the Solaris Management Console" in *System Administration Guide: Basic Administration* and "Using the Solaris Mangement Console Tools in a Name Service Environment (Task Map)" in *System Administration Guide: Basic Administration*
Monitor system performance.	Access the Performance tool under System Status.	"How to Access the Performance Tool (Task)" on page 129
Add resource controls to projects.	Access the Resource Controls tab under System Configuration.	"How to Access the Resource Controls Tab" on page 132

Overview

Resource management functionality is a component of the Solaris Management Console. The console is a container for GUI-based administrative tools that are stored in collections called toolboxes. For information on the console and how to use it, see "Working With the Solaris Management Console Tools (Tasks)" in *System Administration Guide: Basic Administration*.

When using the console and its tools, the main source of documentation is the online help system in the console itself. For a description of the documentation available in the online help, see "Solaris Management Console Interface (Overview)" in *System Administration Guide: Basic Administration*.

Management Scope

The term *management scope* refers to the name service environment that you choose to use with the selected management tool. The management scope choices for the resource control and performance tools are the`/etc/project` local file, or NIS.

The management scope you select during a console session should correspond to the primary name service identified in the `/etc/nsswitch.conf` file.

Performance Tool

The Performance tool is used to monitor resource utilization. Resource utilization can be summarized for the system, viewed by project, or viewed for an individual user.

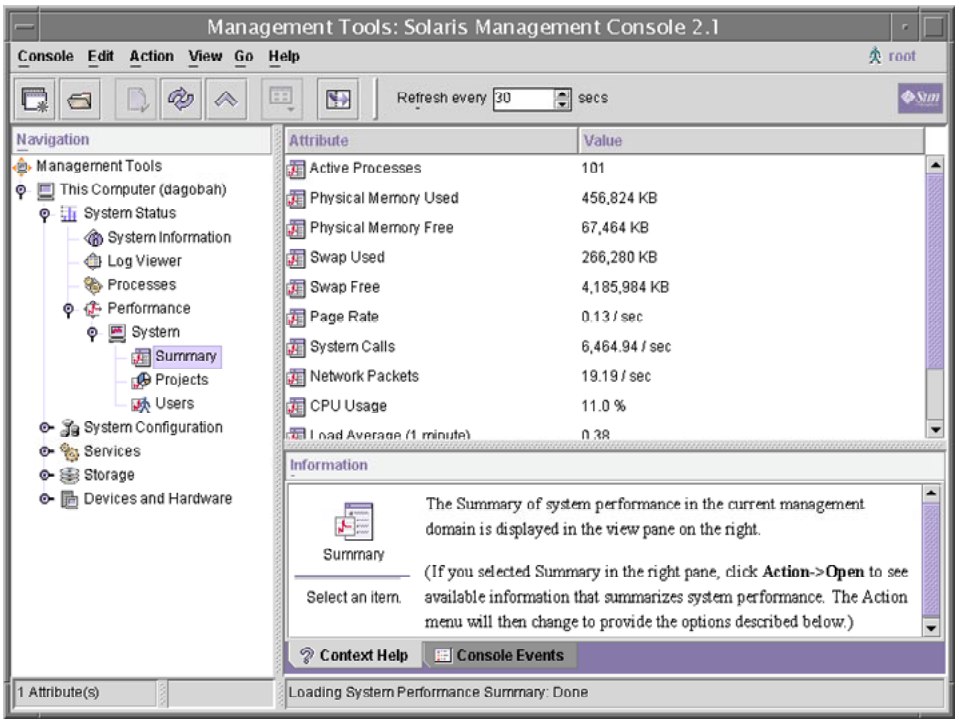


FIGURE 11–1 Performance Tool in the Solaris Management Console

▼ How to Access the Performance Tool (Task)

The Performance tool is located under System Status in the Navigation pane. To access the Performance tool:

1. **Click the System Status control entity in the Navigation pane.**
   The control entity is used to expand menu items in the Navigation pane.
2. **Click the Performance control entity.**
3. **Click the System control entity.**
4. **Double-click Summary, Projects, or Users.**
   Your choice will depend on the usage you want to monitor.

Monitoring by System

Values are shown for the following attributes:

Attribute	Description
Active Processes	Number of processes active on the system
Physical Memory Used	Amount of system memory in use
Physical Memory Free	Amount of system memory available
Swap Used	Amount of system swap space in use
Swap Free	Amount of free system swap space
Page Rate	Rate of system paging activity
System Calls	Number of system calls over the last time interval
Network Packets	Number of network packets transmitted per second
CPU Usage	Percentage of CPU currently in use
Load Average	Number of processes in the system run queue averaged over the last 1, 5, and 15 minutes

Monitoring by Project or User Name

Values are shown for the following attributes:

Attribute	Short Name	Description
Input Blocks	inblk	
Blocks Written	oublk	
Chars Read/Written	ioch	
Data Page Fault Sleep Time	dftime	
Involuntary Context Switches	ictx	
System Mode Time	stime	
Major Page Faults	majfl	
Messages Received	mrcv	

Attribute	Short Name	Description
Messages Sent	msend	
Minor Page Faults	minf	
Num Processes	nprocs	
Num LWPs	count	
Other Sleep Time	slptime	
CPU Time	pctcpu	
Memory Used	pctmem	
Heap Size	brksize	
Resident Set Size	rsssize	
Process Image Size	size	
Signals Received	sigs	
Stopped Time	stoptime	
Swap Operations	swaps	
System Calls Made	sysc	
System Page Fault Sleep Time	kftime	
System Trap Time	ttime	
Text Page Fault Sleep Time	tftime	
User Lock Wait Sleep Time	ltime	
User Mode Time	utime	
User and System Mode Time	time	
Voluntary Context Switches	vctx	
Wait CPU Time	wtime	

Resource Controls Tab

Resource controls allow you to associate a project with a set of resource constraints. These constraints will determine the allowable resource usage of tasks and processes running in the context of the project.

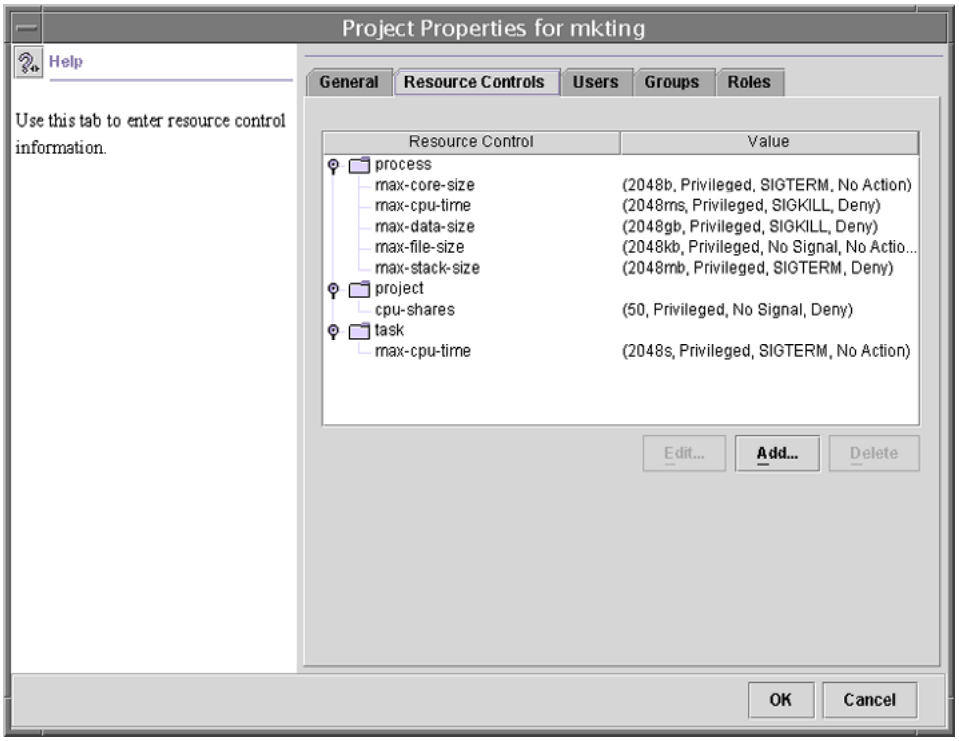


FIGURE 11–2 Resource Controls Tab in the Solaris Management Console

▼ How to Access the Resource Controls Tab

The Resource Controls tab is located under System Configuration in the Navigation pane. To access Resource Controls:

1. **Click the System Configuration control entity in the Navigation pane.**
2. **Double-click Projects.**
3. **Click on a project in the console main window to select it.**

4. **Select Properties from the Action menu.**

5. **Click the Resource Controls tab.**
   View, add, edit, or delete resource control values for processes, projects, and tasks.

Resource Controls You Can Set

To view the list of available resource controls, see About Resource Controls in the console or "Available Resource Controls" on page 88.

Setting Values

You can view, add, edit, or delete resource control values for processes, projects, and tasks. These operations are performed through dialog boxes in the console.

Resource controls and values are viewed in tables in the console. The Resource Control column lists the resource controls that can be set. The Value column displays the properties associated with each resource control. In the table, these values are enclosed in parentheses, and they appear as plain text separated by commas. The values in parentheses comprise an "action clause." Each action clause is composed of a threshold, a privilege level, one signal, and one local action associated with the particular threshold. Each resource control can have multiple action clauses, which are also separated by commas.

Note – Values altered in the `project` database through the console are not applied to currently active projects. The revised values take effect when the project is restarted.

References

For information on projects and tasks, see Chapter 6. For information on resource controls, see Chapter 8. For information on the fair share scheduler (FSS), see Chapter 9.

CHAPTER 12

Accessing Remote File Systems Topics

Chapter 13	Provides overview information for the NFS service
Chapter 14	Provides step-by-step instructions for setting up and troubleshooting the NFS service
Chapter 15	Provides background information on the NFS service

CHAPTER 13

Solaris NFS Environment

This chapter provides an overview of the NFS environment. It includes a short introduction to networking, a description of the NFS service, and a discussion of the concepts necessary to understand the NFS environment.

NFS Servers and Clients

The terms *client* and *server* are used to describe the roles that a computer plays when sharing file systems. If a file system resides on a computer's disk and that computer makes the file system available to other computers on the network, that computer acts as a server. The computers that are accessing that file system are said to be clients. The NFS service enables any given computer to access any other computer's file systems and, at the same time, to provide access to its own file systems. A computer can play the role of client, server, or both at any given time on a network.

Clients access files on the server by mounting the server's shared file systems. When a client mounts a remote file system, it does not make a copy of the file system; rather, the mounting process uses a series of remote procedure calls that enable the client to access the file system transparently on the server's disk. The mount looks like a local mount and users type commands as if the file systems were local. See "Mounting File Systems" on page 150 for information about tasks that mount file systems.

After a file system has been shared on a server through an NFS operation, it can be accessed from a client. You can mount an NFS file system automatically with autofs.

See “Automatic File-System Sharing” on page 146 and “Autofs Administration Task Overview” on page 161 for tasks involving the `share` command and autofs.

NFS File Systems

The objects that can be shared with the NFS service include any whole or partial directory tree or a file hierarchy—including a single file. A computer cannot share a file hierarchy that overlaps one that is already shared. Peripheral devices such as modems and printers cannot be shared.

In most UNIX system environments, a file hierarchy that can be shared corresponds to a file system or to a portion of a file system; however, NFS support works across operating systems, and the concept of a file system might be meaningless in other, non-UNIX environments. Therefore, the term *file system* used throughout this guide refers to a file or file hierarchy that can be shared and mounted over the NFS environment.

About the NFS Environment

The NFS service enables computers of different architectures running different operating systems to share file systems across a network. NFS support has been implemented on many platforms ranging from the MS-DOS to the VMS operating systems.

The NFS environment can be implemented on different operating systems because it defines an abstract model of a file system, rather than an architectural specification. Each operating system applies the NFS model to its file system semantics. This means that file system operations like reading and writing function as though they are accessing a local file.

The benefits of the NFS service are that it:

- Allows multiple computers to use the same files, so everyone on the network can access the same data
- Reduces storage costs by having computers share applications instead of needing local disk space for each user application
- Provides data consistency and reliability because all users can read the same set of files

- Makes mounting of file systems transparent to users
- Makes accessing remote files transparent to users
- Supports heterogeneous environments
- Reduces system administration overhead

The NFS service makes the physical location of the file system irrelevant to the user. You can use the NFS implementation to enable users to see all the relevant files regardless of location. Instead of placing copies of commonly used files on every system, the NFS service enables you to place one copy on one computer's disk and have all other systems access it across the network. Under NFS operation, remote file systems are almost indistinguishable from local ones.

NFS Version 2

Version 2 was the first version of the NFS protocol in wide use. It continues to be available on a large variety of platforms. All Solaris releases support version 2 of the NFS protocol, but Solaris releases prior to Solaris 2.5 support version 2 only.

NFS Version 3

An implementation of NFS version 3 protocol was a new feature of the Solaris 2.5 release. Several changes have been made to improve interoperability and performance. For optimal use, the version 3 protocol must be running on both the NFS servers and clients.

This version allows for safe asynchronous writes on the server, which improves performance by allowing the server to cache client write requests in memory. The client does not need to wait for the server to commit the changes to disk, so the response time is faster. Also, the server can batch the requests, which improves the response time on the server.

All Solaris NFS version 3 operations return the file attributes, which are stored in the local cache. Because the cache is updated more often, the need to do a separate operation to update this data arises less often. Therefore, the number of RPC calls to the server is reduced, improving performance.

The process for verifying file access permissions has been improved. In particular, version 2 would generate a message reporting a "write error" or a "read error" if users tried to copy a remote file to which they do not have permissions. In version 3, the permissions are checked before the file is opened, so the error is reported as an "open error."

The NFS version 3 implementation removes the 8-Kbyte transfer size limit. Clients and servers negotiate whatever transfer size they support, rather than be restricted by the 8-Kbyte limit that was imposed in version 2. The Solaris 2.5 implementation defaults to a 32-Kbyte transfer size.

NFS ACL Support

Access control list (ACL) support was added in the Solaris 2.5 release. ACLs provide a finer-grained mechanism to set file access permissions than is available through standard UNIX file permissions. NFS ACL support provides a method of changing and viewing ACL entries from a Solaris NFS client to a Solaris NFS server. See "Using Access Control Lists (ACLs)" in *System Administration Guide: Security Services* for more information about ACLs.

NFS Over TCP

The default transport protocol for the NFS protocol was changed to the Transport Control Protocol (TCP) in the Solaris 2.5 release, which helps performance on slow networks and wide area networks. TCP provides congestion control and error recovery. NFS over TCP works with version 2 and version 3. Prior to 2.5, the default NFS protocol was User Datagram Protocol (UDP).

Network Lock Manager

The Solaris 2.5 release also included an improved version of the network lock manager, which provided UNIX record locking and PC file sharing for NFS files. The locking mechanism is now more reliable for NFS files, so commands which use locking are less likely to hang.

NFS Large File Support

The Solaris 2.6 implementation of the NFS version 3 protocol was changed to correctly manipulate files larger than 2 Gbytes. The NFS version 2 protocol and the Solaris 2.5 implementation of the version 3 protocol cannot handle files larger than 2 Gbytes.

NFS Client Failover

Dynamic failover of read-only file systems was added in the Solaris 2.6 release. It provides a high level of availability for read-only resources that are already replicated, such as man pages, other documentation, and shared binaries. Failover can occur anytime after the file system is mounted. Manual mounts can now list multiple replicas, much like the automounter allowed in previous releases. The automounter has not changed, except that failover need not wait until the file system is remounted. See "How to Use Client-Side Failover" on page 153 and "Client-Side Failover" on page 216 for more information.

Kerberos Support for the NFS Environment

Support for Kerberos V4 clients was included in the Solaris 2.0 release. In release 2.6, the `mount` and `share` commands were altered to support NFS mounts using Kerberos V5 authentication. Also, the `share` command was changed to allow for multiple authentication flavors to different clients. See "RPCSEC_GSS Security Flavor" on page 141 for more information about changes which involve security flavors. See "Configuring SEAM NFS Servers" in *System Administration Guide: Security Services* for information about Kerberos V5 authentication.

WebNFS Support

The Solaris 2.6 release also included the ability to make a file system on the Internet accessible through firewalls, using an extension to the NFS protocol. One of the advantages to using the WebNFS™ protocol for Internet access is its reliability: the service is built as an extension of the NFS version 3 and version 2 protocol. Also, an NFS server provides greater throughput under a heavy load than HyperText Transfer Protocol (HTTP) access to a Web server. This can decrease the amount of time required to retrieve a file. In addition, the WebNFS implementation provides the ability to share these files without the administrative overhead of an anonymous `ftp` site. See "Security Negotiation for the WebNFS Service" on page 142 for a description of more changes related to WebNFS. See "WebNFS Administration Tasks" on page 159 for more task information.

RPCSEC_GSS Security Flavor

A security flavor, called RPCSEC_GSS, is supported in the Solaris 7 release. This flavor uses the standard GSS-API interfaces to provide authentication, integrity and privacy, as well as allowing for support of multiple security mechanisms. See "Kerberos

Support for the NFS Environment" on page 141 for more information about support of Kerberos V5 authentication. See *GSS-API Programming Guide* for more information about GSS-API.

Solaris 7 Extensions for NFS Mounting

Included in the Solaris 7 release are extensions to the `mount` and `automountd` command that allow for the mount request to use the public file handle instead of the MOUNT protocol. This is the same access method that the WebNFS service uses. By circumventing the MOUNT protocol, the mount can occur through a firewall. In addition, because fewer transactions need to occur between the server and client, the mount should occur faster.

The extensions also allow for NFS URLs to be used instead of the standard path name. Also, you can use the `-public` option with the `mount` command and the automounter maps to force the use of the public file handle. See "WebNFS Support" on page 141 for more information about changes to the WebNFS service.

Security Negotiation for the WebNFS Service

A new protocol has been added to enable a WebNFS client to negotiate a security mechanism with an NFS server. This provides the ability to use secure transactions when using the WebNFS service.

NFS Server Logging

NFS server logging allows an NFS server to provide a record of file operations performed on its file systems. The record includes information to track what is accessed, when it is accessed, and who accessed it. You can specify the location of the logs that contain this information through a set of configuration options. You can also use these options to select the operations that should be logged. This feature is particularly useful for sites that make anonymous FTP archives available to NFS and WebNFS clients. See "How to Enable NFS Server Logging" on page 149 for more information.

About Autofs

File systems shared through the NFS service can be mounted using automatic mounting. Autofs, a client-side service, is a file system structure that provides automatic mounting. The autofs file system is initialized by `automount`, which is run automatically when a system is booted. The automount daemon, `automountd`, runs continuously, mounting and unmounting remote directories on an as-needed basis.

Whenever a user on a client computer running `automountd` tries to access a remote file or directory, the daemon mounts the 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 certain period of time, it is automatically unmounted.

Mounting need not be done at boot time, and the user no longer has to know the superuser password to mount a directory; users need not use the `mount` and `umount` commands. The autofs service mounts and unmounts file systems as required without any intervention on the part of the user.

Mounting some file hierarchies with `automountd` does not exclude the possibility of mounting others with `mount`. A diskless computer *must* mount `/` (root), `/usr`, and `/usr/kvm` through the `mount` command and the `/etc/vfstab` file.

"Autofs Administration Task Overview" on page 161 and "How Autofs Works" on page 229 give more specific information about the autofs service.

Autofs Features

Autofs works with file systems specified in the local name space. This information can be maintained in NIS, NIS+, or local files.

A fully multithreaded version of `automountd` was included in the Solaris 2.6 release. This enhancement makes autofs more reliable and allows for concurrent servicing of multiple mounts, which prevents the service from hanging if a server is unavailable.

The new `automountd` also provides better on-demand mounting. Previous releases would mount an entire set of file systems if they were hierarchically related. Now only the top file system is mounted. Other file systems related to this mount point are mounted when needed.

The autofs service supports browsability of indirect maps. This support allows a user to see what directories could be mounted, without having to actually mount each one of the file systems. A `-nobrowse` option has been added to the autofs maps, so that

large file systems, such as `/net` and `/home`, are not automatically browsable. Also, you can turn off autofs browsability on each client by using the `-n` option with `automount`.

CHAPTER 14

Remote File-System Administration (Tasks)

This chapter provides information on how to perform such NFS administration tasks as setting up NFS services, adding new file systems to share, mounting file systems, using the Secure NFS system, or using the WebNFS functionality. The last part of the chapter includes troubleshooting procedures and a list of many of the NFS error messages and their meanings.

Your responsibilities as an NFS administrator depend on your site's requirements and the role of your computer on the network. You might be responsible for all the computers on your local network, in which case you might be responsible for determining these configuration items:

- Which computers, if any, should be dedicated servers
- Which computers should act as both servers and clients
- Which computers should be clients only

Maintaining a server after it has been set up involves the following tasks:

- Sharing and unsharing file systems as necessary
- Modifying administrative files to update the lists of file systems your computer shares or mounts automatically
- Checking the status of the network
- Diagnosing and fixing NFS related problems as they arise
- Setting up maps for autofs

Remember, a computer can be both a server and a client—sharing local file systems with remote computers and mounting remote file systems.

Automatic File-System Sharing

Servers provide access to their file systems by sharing them over the NFS environment. You specify which file systems are to be shared with the `share` command or the `/etc/dfs/dfstab` file.

Entries in the `/etc/dfs/dfstab` file are shared automatically whenever you start NFS server operation. You should set up automatic sharing if you need to share the same set of file systems on a regular basis. For example, if your computer is a server that supports home directories, you need to make the home directories available at all times. Most file-system sharing should be done automatically; the only time that manual sharing should occur is during testing or troubleshooting.

The `dfstab` file lists all the file systems that your server shares with its clients and controls which clients can mount a file system. If you want to modify `dfstab` to add or delete a file system or to modify the way sharing is done, edit the file with any supported text editor (such as `vi`). The next time the computer enters run level 3, the system reads the updated `dfstab` to determine which file systems should be shared automatically.

Each line in the `dfstab` file consists of a `share` command—the same command you type at the command-line prompt to share the file system. The `share` command is located in `/usr/sbin`.

TABLE 14–1 File-System Sharing Task Map

Task	Description	For Instructions, Go to ...
Establish automatic file-system sharing	Steps to configure a server so that file systems are automatically shared when the server is rebooted.	"How to Set Up Automatic File-System Sharing" on page 147
Enable WebNFS	Steps to configure a server so that users can access files using WebNFS	"How to Enable WebNFS Access" on page 147
Enabling NFS server logging	Steps to configure a server so that NFS logging is run on selected file systems	"How to Enable NFS Server Logging" on page 149

▼ How to Set Up Automatic File-System Sharing

1. **Become superuser.**
2. **Add entries for each file system to be shared.**

   Edit `/etc/dfs/dfstab` and add one entry to the file for each file system that you want to be automatically shared. Each entry must be on a line by itself in the file and uses this syntax:

   ```
   share [-F nfs] [-o specific-options] [-d description] pathname
   ```

   See the `share_nfs`(1M) man page for a complete list of options.
3. **Check if the NFS service is running on the server.**

   If this is the first `share` command or set of `share` commands that you have initiated, it is likely that the NFS daemons are not running. Check to see if one of the NFS daemons is running by using the following command.

   ```
   # pgrep nfsd
   318
   ```

   `318` is the process ID for `nfsd` in this example. If a ID is not displayed, it means that the service is not running. The second deamon to check for is `mountd`.
4. **(Optional) Start the NFS service.**

   If the previous step does not report a process ID for `nfsd`, start the NFS service by using the following command.

   ```
   # /etc/init.d/nfs.server start
   ```

   This ensures that NFS service is now running on the servers and will restart automatically when the server is at run level 3 during boot.

Where to Go From Here

The next step is to set up your autofs maps so that clients can access the file systems you have shared on the server. See “Autofs Administration Task Overview” on page 161.

▼ How to Enable WebNFS Access

Starting with the 2.6 release, by default all file systems that are available for NFS mounting are automatically available for WebNFS access. The only time that this procedure needs to be followed is on servers that do not already allow NFS mounting, or if one of the following conditions apply.

- to reset the public file handle to shorten NFS URLs using the `-public` option

- to force the loading of a specific html file using the `-index` option

See "Planning for WebNFS Access" on page 160 for a list of issues that you should consider before starting the WebNFS service.

1. **Become superuser.**

2. **Add entries for each file system to be shared using the WebNFS service.**

   Edit `/etc/dfs/dfstab` and add one entry to the file for each file system. The `-public` and `-index` tag shown in the following example are optional.

   ```
   share -F nfs -o ro,public,index=index.html /export/ftp
   ```

   See the `share_nfs`(1M) man page for a complete list of options.

3. **Check if the NFS service is running on the server.**

   If this is the first `share` command or set of `share` commands that you have initiated, it is likely that the NFS daemons are not running. Check to see if one of the NFS daemons is running by using the following command.

   ```
   # pgrep nfsd
   318
   ```

   `318` is the process ID for `nfsd` in this example. If a ID is not displayed, it means that the service is not running. The second deamon to check for is `mountd`.

4. **(Optional) Start the NFS service.**

   If the previous step does not report a process ID for `nfsd`, start the NFS service by using the following command.

   ```
   # /etc/init.d/nfs.server start
   ```

   This ensures that NFS service is now running on the servers and will restart automatically when the server is at run level 3 during boot.

5. **(Optional) Share the file system.**

   After the entry is in `/etc/dfs/dfstab`, the file system can be shared by either rebooting the system or by using the `shareall` command. If the NFS service was started earlier, this command does not need to be run because the script runs the command.

   ```
   # shareall
   ```

6. **Verify that the information is correct.**

   Run the `share` command to check that the correct options are listed:

   ```
   # share
   -        /export/share/man   ro   ""
   -        /usr/src     rw=eng   ""
   -        /export/ftp    ro,public,index=index.html  ""
   ```

▼ How to Enable NFS Server Logging

1. **Become superuser.**

2. **Optional: Change file system configuration settings.**

   In `/etc/nfs/nfslog.conf`, you can either edit the default settings for all file systems by changing the data associated with the `global` tag or you can add a new tag for this file system. If these changes are not needed you do not need to change this file. The format of `/etc/nfs/nfslog.conf` is described in `nfslog.conf`(4).

3. **Add entries for each file system to be shared using NFS server logging.**

   Edit `/etc/dfs/dfstab` and add one entry to the file for the file system that you want to have NFS server logging enabled on. The tag used with the `log=`*tag* option must be entered in `/etc/nfs/nfslog.conf`. This example uses the default settings in the `global` tag.

   ```
   share -F nfs -o ro,log=global /export/ftp
   ```

   See the `share_nfs`(1M) man page for a complete list of options.

4. **Check if the NFS service is running on the server.**

   If this is the first `share` command or set of `share` commands that you have initiated, it is likely that the NFS daemons are not running. Check to see if one of the NFS daemons is running by using the following command.

   ```
   # pgrep nfsd
   318
   ```

   `318` is the process ID for `nfsd` in this example. If a ID is not displayed, it means that the service is not running. The second deamon to check for is `mountd`.

5. **(Optional) Start the NFS service.**

   If the previous step does not report a process ID for `nfsd`, start the NFS service by using the following command.

   ```
   # /etc/init.d/nfs.server start
   ```

   This ensures that NFS service is now running on the servers and will restart automatically when the server is at run level 3 during boot.

6. **(Optional) Share the file system.**

   After the entry is in `/etc/dfs/dfstab`, the file system can be shared by either rebooting the system or by using the `shareall` command. If the NFS services was started earlier, this command does not need to be run because the script runs the command.

   ```
   # shareall
   ```

7. **Verify that the information is correct.**

   Run the `share` command to check that the correct options are listed:

```
# share
-        /export/share/man   ro   ""
-        /usr/src     rw=eng   ""
-        /export/ftp    ro,log=global  ""
```

8. **Start the NFS log daemon, `nfslogd`, if it is not running already.**

   Restarting the NFS daemons using the `nfs.server` script will start the daemon if the `nfslog.conf` file exists. Otherwise the command needs to be run once by hand to create the files so that the command will automatically restart when the server is rebooted.

   ```
   # /usr/lib/nfs/nfslogd
   ```

Mounting File Systems

You can mount file systems in several ways. They can be mounted automatically when the system is booted, on demand from the command line, or through the automounter. The automounter provides many advantages to mounting at boot time or mounting from the command line, but many situations require a combination of all three. In addition to these three ways of mounting a file system, there are several ways of enabling or disabling processes depending on the options you use when mounting the file system. See the following table for a complete list of the tasks associated with file system mounting.

TABLE 14–2 Mounting File Systems Task Map

Task	Description	For Instructions, Go to ...
Mount a file system at boot time	Steps so that a file system is mounted whenever a system is rebooted.	"How to Mount a File System at Boot Time" on page 151
Mount a file system using a command	Steps to mount a file system when a system is running. This procedure is useful when testing.	"How to Mount a File System From the Command Line" on page 152
Mount with the automounter	Steps to access a file system on demand without using the command line.	"Mounting With the Automounter" on page 152
Disallowing large files	Steps to prevent large files from being created on a file system.	"How to Disable Large Files on an NFS Server" on page 152
Using client-side failover	Steps to enable the automatic switchover to a working file system if a server fails.	"How to Use Client-Side Failover" on page 153
Disabling mount access for a client	Steps to disable the ability of one client to access a remote file system.	"How to Disable Mount Access for One Client" on page 154

TABLE 14–2 Mounting File Systems Task Map *(Continued)*

Task	Description	For Instructions, Go to ...
Providing access to a file system through a firewall	Steps to allow access to a file system through a firewall by using the WebNFS protocol.	"How to Mount an NFS File System Through a Firewall" on page 154
Mounting a file system using a NFS URL	Steps to allow access to a file system using an NFS URL. This process allows for file-system access without using the MOUNT protocol.	"How to Mount an NFS File System Using an NFS URL" on page 155

▼ How to Mount a File System at Boot Time

If you want to mount file systems at boot time instead of using autofs maps, follow this procedure. Although you must follow this procedure for all local file systems, it is not recommended for remote file systems because it must be completed on every client.

1. **Become superuser.**
2. **Add an entry for the file system to** `/etc/vfstab`**.**

   Entries in the `/etc/vfstab` file have the following syntax:

   ```
   special  fsckdev  mountp  fstype  fsckpass  mount-at-boot  mntopts
   ```

   See the `vfstab`(4) man page for more information.

Caution – NFS servers should not have NFS `vfstab` entries because of a potential deadlock. The NFS service is started after the entries in `/etc/vfstab` are checked, so that if two servers that are mounting file systems from each other fail at the same time, each system could hang as the systems reboot.

Example of a `vfstab` entry

You want a client computer to mount the `/var/mail` directory from the server `wasp`. You would like the file system to be mounted as `/var/mail` on the client and you want the client to have read-write access. Add the following entry to the client's `vfstab` file.

```
wasp:/var/mail - /var/mail nfs - yes rw
```

▼ How to Mount a File System From the Command Line

Mounting a file system from the command line is often done to test a new mount point or to allow for temporary access to a file system that is not available through the automounter.

1. **Become superuser.**
2. **Mount the file system.**

   Type the following command:

   ```
   # mount -F nfs -o ro bee:/export/share/local /mnt
   ```

   In this case, the `/export/share/local` file system from the server `bee` is mounted on read-only `/mnt` on the local system. Mounting from the command line allows for temporary viewing of the file system. You can unmount the file system with `umount` or by rebooting the local host.

Caution – Starting with the 2.6 release, all versions of the `mount` command will not warn about invalid options. The command silently ignores any options that cannot be interpreted. Make sure you verify all of the options that were used, to prevent unexpected behavior.

Mounting With the Automounter

"Autofs Administration Task Overview" on page 161 includes the specific instructions for establishing and supporting mounts with the automounter. Without any changes to the generic system, clients should be able to access remote file systems through the `/net` mount point. To mount the `/export/share/local` file system from the previous example, all you need to do is type:

```
% cd /net/bee/export/share/local
```

Because the automounter allows all users to mount file systems, root access is not required. It also provides for automatic unmounting of file systems, so there is no need to unmount file systems after you are finished.

▼ How to Disable Large Files on an NFS Server

For servers that are supporting clients that cannot handle a file over 2 GBytes, it might be necessary to disable the ability to create large files.

Note – Previous versions of the Solaris operating environment cannot use large files. Check that clients of the NFS server are running at least the 2.6 release if the clients need to access large files.

1. **Become superuser.**
2. **Check that no large files exist on the file system.**

   Here is an example of a command that you can run to locate large files:

   ```
   # cd /export/home1
   # find . -xdev -size +2000000 -exec ls -l {} \;
   ```

   If large files are on the file system, you must remove or move them to another file system.
3. **Unmount the file system.**

   ```
   # umount /export/home1
   ```
4. **Reset the file system state if the file system has been mounted using** `-largefiles`**.**

   `fsck` resets the file system state if no large files exist on the file system:

   ```
   # fsck /export/home1
   ```
5. **Mount the file system using** `nolargefiles`**.**

   ```
   # mount -F ufs -o nolargefiles /export/home1
   ```

   You can do this from the command line, but to make the option more permanent, add an entry like the following into `/etc/vfstab`:

```
/dev/dsk/c0t3d0s1 /dev/rdsk/c0t3d0s1 /export/home1  ufs  2  yes  nolargefiles
```

▼ How to Use Client-Side Failover

1. **Become superuser.**
2. **On the NFS client, mount the file system using the** `ro` **option.**

   You can do this from the command line, through the automounter, or by adding an entry to `/etc/vfstab` that looks like:

   ```
   bee,wasp:/export/share/local  -  /usr/local  nfs  -  no  -o ro
   ```

   This syntax has been allowed by the automounter in earlier releases, but the failover was not available while file systems were mounted, only when a server was being selected.

Note – Servers that are running different versions of the NFS protocol cannot be mixed using a command line or in a `vfstab` entry. Mixing servers supporting NFS V2 and V3 protocols can only be done with autofs, in which case the best subset of version 2 or version 3 servers is used.

▼ How to Disable Mount Access for One Client

1. **Become superuser.**
2. **Add an entry in `/etc/dfs/dfstab`.**

   The first example allows mount access to all clients in the `eng` netgroup except the host named `rose`. The second example allows mount access to all clients in the `eng.sun.com` DNS domain except for `rose`.

   ```
   share -F nfs -o ro=-rose:eng /export/share/man
   share -F nfs -o ro=-rose:.eng.sun.com /export/share/man
   ```

   For additional information on access lists, see "Setting Access Lists With the `share` Command" on page 204.
3. **Share the file system.**

   The NFS server does not use changes to `/etc/dfs/dfstab` until the file systems are shared again or until the server is rebooted.

   ```
   # shareall
   ```

▼ How to Mount an NFS File System Through a Firewall

1. **Become superuser.**
2. **Manually mount the file system, using a command like:**

   ```
   # mount -F nfs -o public bee:/export/share/local /mnt
   ```

   In this example the file system `/export/share/local` is mounted on the local client using the public file handle. An NFS URL can be used instead of the standard path name. If the public file handle is not supported by the server `bee`, the mount operation will fail.

Note – This procedure requires that the file system on the NFS server be shared using the `-public` option and any firewalls between the client and the server allow TCP connections on port `2049`. Starting with the 2.6 release, all file systems that are shared allow for public file handle access, so the `-public` option is applied by default.

▼ How to Mount an NFS File System Using an NFS URL

1. **Become superuser.**
2. **Manually mount the file system, using a command such as:**

   ```
   # mount -F nfs nfs://bee:3000/export/share/local /mnt
   ```

   In this example, the `/export/share/local` file system is being mounted from the server `bee` using NFS port number `3000`. The port number is not required and by default uses the standard NFS port number of `2049`. You can include the public option with an NFS URL, if you want. Without the public option, the MOUNT protocol is used if the public file handle is not supported by the server. The public option will force the use of the public file handle, and the mount will fail if the public file handle is not supported.

Setting Up NFS Services

This section discusses some of the tasks necessary to initialize or use NFS services.

TABLE 14–3 NFS Services Task Map

Task	Description	For Instructions, Go To ...
Start the NFS server	Steps to start the NFS service, if it has not been started automatically.	"How to Start the NFS Services" on page 156
Stop the NFS server	Steps to stop the NFS service. Normally the service should not need to be stopped.	"How to Stop the NFS Services" on page 156
Start the automounter	Steps to start the automounter. This procedure is required when some of the automounter maps are changed.	"How to Start the Automounter" on page 156

TABLE 14–3 NFS Services Task Map *(Continued)*

Task	Description	For Instructions, Go To ...
Stop the automounter	Steps to stop the automounter. This procedure is required when some of the automounter maps are changed.	"How to Stop the Automounter" on page 156

▼ How to Start the NFS Services

1. **Become superuser.**
2. **Enable the NFS service daemons.**
   Type the following command:

   ```
   # /etc/init.d/nfs.server start
   ```

   This starts the daemons if there is an entry in `/etc/dfs/dfstab`.

▼ How to Stop the NFS Services

1. **Become superuser.**
2. **Disable the NFS service daemons.**
   Type the following command:

   ```
   # /etc/init.d/nfs.server stop
   ```

▼ How to Start the Automounter

1. **Become superuser.**
2. **Enable the autofs daemon.**
   Type the following command:

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

   This starts the daemon.

▼ How to Stop the Automounter

1. **Become superuser.**

2. **Disable the autofs daemon.**

   Type the following command:

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

Administering the Secure NFS System

To use the Secure NFS system, all the computers you are responsible for must have a domain name. A domain is an administrative entity, typically consisting of several computers, that is part of a larger network. If you are running NIS+, you should also establish the NIS+ name service for the domain. See *System Administration Guide: Naming and Directory Services (FNS and NIS+)*.

You can configure the Secure NFS environment to use Diffie-Hellman authentication."Using Authentication Services (Tasks)" in *System Administration Guide: Security Services* discusses this authentication service.

Kerberos V5 authentication is also supported by the NFS service. "Introduction to SEAM" in *System Administration Guide: Security Services* discusses the Kerberos service.

▼ How to Set Up a Secure NFS Environment With DH Authentication

1. **Assign your domain a domain name, and make the domain name known to each computer in the domain.**

   See the *System Administration Guide: Naming and Directory Services (FNS and NIS+)* if you are using NIS+ as your name service.

2. **Establish public keys and secret keys for your clients' users using the `newkey` or `nisaddcred` command, and have each user establish his or her own secure RPC password using the `chkey` command.**

   Note – For information about these commands, see the `newkey`(1M), the `nisaddcred`(1M), and the `chkey`(1) man pages.

   When public and secret keys have been generated, the public and encrypted secret keys are stored in the `publickey` database.

3. **Verify that the name service is responding. If you are running NIS+, type the following:**

```
# nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

   If you are running NIS, verify that the `ypbind` daemon is running.

4. **Verify that the `keyserv` daemon (the key server) is running.**

   Type the following command.

```
# ps -ef | grep keyserv
root    100     1  16    Apr 11 ?      0:00 /usr/sbin/keyserv
root      2215  2211   5  09:57:28 pts/0  0:00 grep keyserv
```

   If the daemon isn't running, start the key server by typing the following:

```
# /usr/sbin/keyserv
```

5. **Decrypt and store the secret key.**

   Usually, the login password is identical to the network password. In this case, `keylogin` is not required. If the passwords are different, the users have to log in, and then do a `keylogin`. You still need to use the `keylogin -r` command as root to store the decrypted secret key in `/etc/.rootkey`.

Note – You only need to run `keylogin -r` if the root secret key changes or `/etc/.rootkey` is lost.

---

6. **Update mount options for the file system.**

   Edit the `/etc/dfs/dfstab` file and add the `sec=dh` option to the appropriate entries (for Diffie-Hellman authentication).

```
share -F nfs -o sec=dh /export/home
```

7. **Update the automounter maps for the file system.**

   Edit the `auto_master` data to include `sec=dh` as a mount option in the appropriate entries (for Diffie-Hellman authentication):

```
/home    auto_home    -nosuid,sec=dh
```

Note – With 2.5 and earlier Solaris releases, if a client does not mount as secure a file system that is shared as secure, users have access as user `nobody`, rather than as themselves. With Version 2 on later releases, the NFS server refuses access if the security modes do not match, unless `-sec=none` is included on the `share` command line. With version 3, the mode is inherited from the NFS server, so clients do not need to specify `sec=krb4` or `sec=dh` . The users have access to the files as themselves.

When you reinstall, move, or upgrade a computer, remember to save `/etc/.rootkey` if you do not establish new keys or change them for `root`. If you do delete `/etc/.rootkey`, you can always type:

```
# keylogin -r
```

WebNFS Administration Tasks

This section provides instructions for administering the WebNFS system. This is a list of some related tasks.

TABLE 14–4 WebNFS Administration Task Map

Task	Description	For Instructions, Go To ...
Plan for WebNFS	Issues to consider before enabling the WebNFS service.	"Planning for WebNFS Access" on page 160
Enable WebNFS	Steps to enable mounting of an NFS file system using the WebNFS protocol.	"How to Enable WebNFS Access" on page 147
Enabling WebNFS through a firewall	Steps to allow access to files through a firewall by using the WebNFS protocol.	"Enabling WebNFS Access Through a Firewall" on page 161
Browsing using an NFS URL	Instructions for using an NFS URL within a web browser.	"Browsing Using an NFS URL" on page 161
Using a public file handle with autofs	Steps to force use of the public file handle when mounting a file system with the automounter.	"How to Use a Public File Handle With Autofs" on page 173
Using an NFS URL with autofs	Steps to add an NFS URL to the automounter maps.	"How to Use NFS URLs With Autofs" on page 174
Providing access to a file system through a firewall	Steps to allow access to a file system through a firewall using the WebNFS protocol.	"How to Mount an NFS File System Through a Firewall" on page 154

TABLE 14–4 WebNFS Administration Task Map *(Continued)*

Task	Description	For Instructions, Go To ...
Mounting a file system using an NFS URL	Steps to allow access to a file system using an NFS URL. This process allows for file system access without using the MOUNT protocol.	"How to Mount an NFS File System Using an NFS URL" on page 155

Planning for WebNFS Access

To use the WebNFS functionality, you first need an application capable of running and loading an NFS URL (for example, `nfs://server/path`). The next step is to choose the file system that will be exported for WebNFS access. If the application is web browsing, often the document root for the web server is used. Several factors need to be considered when choosing a file system to export for WebNFS access.

1. Each server has one public file handle that by default is associated with the server's root file system. The path in an NFS URL is evaluated relative to the directory with which the public file handle is associated. If the path leads to a file or directory within an exported file system, the server provides access. You can use the `-public` option of the `share` command to associate the public file handle with a specific exported directory. Using this option allows URLs to be relative to the shared file system rather than to the servers' root file system. The root file system does not allow web access unless the root file system is shared.
2. The WebNFS environment allows users who already have mount privileges to access files through a browser regardless of whether the file system is exported using the `-public` option. Because users already have access to these files through the NFS setup, this should not create any additional security risk. You only need to share a file system using the `-public` option if users who cannot mount the file system need to use WebNFS access.
3. File systems that are already open to the public make good candidates for using the `-public` option, like the top directory in an ftp archive or the main URL directory for a web site.
4. You can use the `-index` option with the `share` command to force the loading of an HTML file instead of listing the directory when an NFS URL is accessed.

   After a file system is chosen, review the files and set access permissions to restrict viewing of files or directories as needed. Establish the permissions as appropriate for any NFS file system that is being shared. For many sites, `755` permissions for directories and `644` permissions for files provides the correct level of access.

   Additional factors need to be considered if both NFS and HTTP URLs are to be used to access one web site. These are described in "WebNFS Limitations With Web Browser Use" on page 219.

▼ Browsing Using an NFS URL

Browsers capable of supporting WebNFS access should provide access using an NFS URL that looks something like:

`nfs://`*server*`<:`*port*`>/`*path*

server	Name of the file server
port	Port number to use (the default value is `2049`)
path	Path to file, which can be relative to the public file handle or to the root file system

---

Note – In most browsers, the URL service type (for example, `nfs` or `http`) is remembered from one transaction to the next, unless a URL that includes a different service type is loaded. After using an NFS URLs, if a reference to an HTTP URL is loaded, subsequent pages are loaded using the HTTP protocol instead of the NFS protocol.

▼ Enabling WebNFS Access Through a Firewall

You can enable WebNFS access for clients that are not part of the local subnet by configuring the firewall to allow a TCP connection on port `2049`. Just allowing access for `httpd` does not allow NFS URLs to be used.

Autofs Administration Task Overview

This section describes some of the most common tasks you might encounter in your own environment. Recommended procedures are included for each scenario to help you configure autofs to best meet your clients' needs.

Note – Use the Solstice System Management Tools or see the *System Administration Guide: Naming and Directory Services (FNS and NIS+)* to perform the tasks discussed in this section.

Autofs Administration Task Map

The following table lists a description and a pointer to many of the tasks that are related to autofs.

TABLE 14–5 Autofs Administration Task Map

Task	Description	For Instructions, Go To ...
Start autofs	Start the automount service without having to reboot the system	"How to Start the Automounter" on page 156
Stop autofs	Stop the automount service without disabling other network services	"How to Stop the Automounter" on page 156
Access file systems using autofs	Access file systems using the automount service	"Mounting With the Automounter" on page 152
Modifying the autofs maps	Steps to modify the master map, which should be used to list other maps	"How to Modify the Master Map" on page 165
	Steps to modify an indirect map, which should be used for most maps	"How to Modify Indirect Maps" on page 165
	Steps to modify a direct map, which should be used when a direct association between a mount point on a client and a server is required	"How to Modify Direct Maps" on page 165
Modifying the autofs maps to access non NFS file systems	Steps to set up an autofs map with an entry for a CD-ROM application	"How to Access CD-ROM Applications With Autofs" on page 167
	Steps to set up an autofs map with an entry for a PC-DOS diskette	"How to Access PC-DOS Data Diskettes With Autofs" on page 167
	Steps to use autofs to access a CacheFS file system	"How to Access NFS File Systems Using CacheFS" on page 167
Using /home	Example of how to set up a common `/home` map	"Setting Up a Common View of `/home`" on page 168

TABLE 14–5 Autofs Administration Task Map *(Continued)*

Task	Description	For Instructions, Go To ...
	Steps to set up a `/home` map that refers to multiple file systems	"How to Set Up `/home` With Multiple Home Directory File Systems" on page 169
Using a new autofs mount point	Steps to set up a project-related autofs map	"How to Consolidate Project-Related Files Under `/ws`" on page 170
	Steps to set up an autofs map that supports different client architectures	"How to Set Up Different Architectures to Access a Shared Name Space" on page 171
	Steps to set up an autofs map that supports different operating systems	"How to Support Incompatible Client Operating System Versions" on page 172
Replicating file systems with autofs	Provide access to file systems that failover	"How to Replicate Shared Files Across Several Servers" on page 173
Using security restrictions with autofs	Provide access to file systems while restricting remote `root` access to the files	"How to Apply Security Restrictions" on page 173
Using a public file handle with autofs	Force use of the public file handle when mounting a file system	"How to Use a Public File Handle With Autofs" on page 173
Using an NFS URL with autofs	Add an NFS URL so that the automounter can use it	"How to Use NFS URLs With Autofs" on page 174
Disable autofs browsability	Steps to disable browability so that autofs mount points are not automatically populated on a single client	"How to Completely Disable Autofs Browsability on a Single NFS Client" on page 174
	Steps to disable browability so that autofs mount points are not automatically populated on all clients	"How to Disable Autofs Browsability for All Clients" on page 175
	Steps to disable browability so that a specific autofs mount point is not automatically populated on a client	"How to Disable Autofs Browsability on an NFS Client" on page 175

Administrative Tasks Involving Maps

The following tables describe several of the factors you need to be aware of when administering autofs maps. Which type of map and which name service you choose changes the mechanism which you need to use to make changes to the autofs maps.

The following table describes the types of maps and their uses.

TABLE 14–6 Types of autofs Maps and Their Uses

Type of Map	Use
Master	Associates a directory with a map
Direct	Directs autofs to specific file systems
Indirect	Directs autofs to reference-oriented file systems

The following table describes how to make changes to your autofs environment based on your name service.

TABLE 14–7 Map Maintenance

Name Service	Method
Local files	Text editor
NIS	`make` files
NIS+	`nistbladm`

The next table tells you when to run the `automount` command, depending on the modification you have made to the type of map. For example, if you have made an addition or a deletion to a direct map, you need to run the `automount` command on the local system to allow the change take effect; however, if you've modified an existing entry, you do not need to run the `automount` command for the change to take effect.

TABLE 14–8 When to Run the `automount` Command

Type of Map	Restart `automount`?	
	Addition or Deletion	Modification
`auto_master`	`Y`	`Y`
`direct`	`Y`	`N`
`indirect`	`N`	`N`

Modifying the Maps

The following procedures require that you use NIS+ as your name service.

▼ How to Modify the Master Map

1. **Using the `nistbladm` command, make the changes you want to the master map.**
   See the *System Administration Guide: Naming and Directory Services (FNS and NIS+)*.
2. **For each client, become superuser.**
3. **For each client, run the `automount` command to ensure the changes you made take effect.**
4. **Notify your users of the changes.**
   Notification is required so that the users can also run the automount command as superuser on their own computers.

   The `automount` command gathers information from the master map whenever it is run.

▼ How to Modify Indirect Maps

- **Using the `nistbladm` command, make the changes you want to the indirect map.**
  See the *System Administration Guide: Naming and Directory Services (FNS and NIS+)*.

  The change takes effect the next time the map is used, which is the next time a mount is done.

▼ How to Modify Direct Maps

1. **Using the `nistbladm` command, add or delete the changes you want to the direct map.**
   See the *System Administration Guide: Naming and Directory Services (FNS and NIS+)*.
2. **If you added or deleted a mount-point entry in step 1, run the `automount` command.**
3. **Notify your users of the changes.**
   Notification is required so that the users can also run the `automount` command as superuser on their own computers.

Note – If you only modify or change the contents of an existing direct map entry, you do not need to run the `automount` command.

For example, suppose you modify the `auto_direct` map so that the `/usr/src` directory is now mounted from a different server. If `/usr/src` is not mounted at this time, the new entry takes effect immediately when you try to access `/usr/src`. If `/usr/src` is mounted now, you can wait until the auto-unmounting takes place, then access it.

Note – Because of the additional steps, and because they do not take up as much space in the mount table as direct maps, use indirect maps whenever possible. They are easier to construct, and less demanding on the computers' file systems.

Avoiding Mount-Point Conflicts

If you have a local disk partition mounted on `/src` and you also want to use the autofs service to mount other source directories, you might encounter a problem. If you specify the mount point `/src`, the service hides the local partition whenever you try to reach it.

You need to mount the partition somewhere else; for example, on `/export/src`. You would then need an entry in `/etc/vfstab` like:

```
/dev/dsk/d0t3d0s5 /dev/rdsk/c0t3d0s5 /export/src ufs 3 yes -
```

and this entry in `auto_src`:

```
terra        terra:/export/src
```

where `terra` is the name of the computer.

Accessing Non NFS File Systems

Autofs can also mount files other than NFS files. Autofs mounts files on removable media, such as diskettes or CD-ROM. Normally, you would mount files on removable media using the Volume Manager. The following examples show how this mounting could be done through autofs. The Volume Manager and autofs do not work together, so these entries would not be used without first deactivating the Volume Manager.

Instead of mounting a file system from a server, you put the media in the drive and reference it from the map. If you want to access non NFS file systems and you are using autofs, see the following procedures.

How to Access CD-ROM Applications With Autofs

Note – Use this procedure if you are *not* using Volume Manager.

1. **Become superuser.**
2. **Update the autofs map.**
   Add an entry for the CD-ROM file system, which should look like:

   ```
   hsfs      -fstype=hsfs,ro      :/dev/sr0
   ```

   The CD-ROM device you want to mount must appear as a name following a colon.

▼ How to Access PC-DOS Data Diskettes With Autofs

Note – Use this procedure if you are *not* using Volume Manager.

1. **Become superuser.**
2. **Update the autofs map.**
   Add an entry for the diskette file system such as:

   ```
   pcfs      -fstype=pcfs      :/dev/diskette
   ```

Accessing NFS File Systems Using CacheFS

The cache file system (CacheFS) is a generic nonvolatile caching mechanism that improves the performance of certain file systems by utilizing a small, fast, local disk.

You can improve the performance of the NFS environment by using CacheFS to cache data from an NFS file system on a local disk.

▼ How to Access NFS File Systems Using CacheFS

1. **Become superuser.**
2. **Run the `cfsadmin` command to create a cache directory on the local disk.**

   ```
   # cfsadmin -c /var/cache
   ```

3. **Add the cachefs entry to the appropriate automounter map.**

   For example, adding this entry to the master map caches all home directories:

   ```
   /home auto_home -fstype=cachefs,cachedir=/var/cache,backfstype=nfs
   ```

   Adding this entry to the `auto_home` map only caches the home directory for the user named `rich`:

```
rich -fstype=cachefs,cachedir=/var/cache,backfstype=nfs dragon:/export/home1/rich
```

Note – Options that are included in maps that are searched later override options set in maps that are searched earlier. The last options found are the ones that are used. In the previous example, a specific entry added to the `auto_home` map only needs to include the options listed in the master maps if some of the options needed to be changed.

Customizing the Automounter

You can set up the automounter maps in several ways. The following tasks give detailed instructions on how to customize the automounter maps to provide an easy-to-use directory structure.

▼ Setting Up a Common View of `/home`

The ideal is for all network users to be able to locate their own, or anyone else's home directory under `/home`. This view should be common across all computers, whether client or server.

Every Solaris installation comes with a master map: `/etc/auto_master`.

```
# Master map for autofs
#
+auto_master
/net      -hosts      -nosuid,nobrowse
/home     auto_home   -nobrowse
/xfn      -xfn
```

A map for `auto_home` is also installed under `/etc`.

```
# Home directory map for autofs
#
+auto_home
```

Except for a reference to an external `auto_home` map, this map is empty. If the directories under `/home` are to be common to all computers, do not modify this `/etc/auto_home` map. All home directory entries should appear in the name service files, either NIS or NIS+.

Note – Users should not be permitted to run `setuid` executables from their home directories; without this restriction, any user could have superuser privileges on any computer.

▼ How to Set Up `/home` With Multiple Home Directory File Systems

1. **Become superuser.**
2. **Install home directory partitions under `/export/home`.**

   If there are several partitions, install them under separate directories, for example, `/export/home1`, `/export/home2`, and so on.
3. **Use the Solstice System Management Tools to create and maintain the `auto_home` map.**

   Whenever you create a new user account, type the location of the user's home directory in the `auto_home` map. Map entries can be simple, for example:

   ```
   rusty        dragon:/export/home1/&
   gwenda       dragon:/export/home1/&
   charles      sundog:/export/home2/&
   rich         dragon:/export/home3/&
   ```

   Notice the use of the `&` (ampersand) to substitute the map key. This is an abbreviation for the second occurrence of `rusty` in the following example.

   ```
   rusty         dragon:/export/home1/rusty
   ```

   With the `auto_home` map in place, users can refer to any home directory (including their own) with the path `/home/`*user*, where *user* is their login name and the key in the map. This common view of all home directories is valuable when logging in to another user's computer. Autofs mounts your home directory for you. Similarly, if you run a remote windowing system client on another computer, the client program has the same view of the `/home` directory as you do on the computer providing the windowing system display.

   This common view also extends to the server. Using the previous example, if `rusty` logs in to the server `dragon`, autofs there provides direct access to the local disk by loopback-mounting `/export/home1/rusty` onto `/home/rusty`.

   Users do not need to be aware of the real location of their home directories. If `rusty` needs more disk space and needs to have his home directory relocated to another server, you need only change `rusty`'s entry in the `auto_home` map to reflect the new

location. Everyone else can continue to use the `/home/rusty` path.

▼ How to Consolidate Project-Related Files Under `/ws`

Assume you are the administrator of a large software development project. You want to make all project-related files available under a directory called `/ws`. This directory is to be common across all workstations at the site.

1. **Add an entry for the `/ws` directory to the site `auto_master` map, either NIS or NIS+.**

   ```
   /ws     auto_ws     -nosuid
   ```

   The `auto_ws` map determines the contents of the `/ws` directory.

2. **Add the `-nosuid` option as a precaution.**

   This option prevents users from running setuid programs that might exist in any workspaces.

3. **Add entries to the `auto_ws` map.**

   The `auto_ws` map is organized so that each entry describes a subproject. Your first attempt yields a map that looks like the following:

   ```
   compiler   alpha:/export/ws/&
   windows    alpha:/export/ws/&
   files      bravo:/export/ws/&
   drivers    alpha:/export/ws/&
   man        bravo:/export/ws/&
   tools      delta:/export/ws/&
   ```

   The ampersand (`&`) at the end of each entry is an abbreviation for the entry key. For instance, the first entry is equivalent to:

   ```
   compiler        alpha:/export/ws/compiler
   ```

   This first attempt provides a map that looks simple, but it turns out to be inadequate. The project organizer decides that the documentation in the `man` entry should be provided as a subdirectory under each subproject. Also, each subproject requires subdirectories to describe several versions of the software. You must assign each of these subdirectories to an entire disk partition on the server.

   Modify the entries in the map as follows:

   ```
   compiler \
       /vers1.0    alpha:/export/ws/&/vers1.0 \
       /vers2.0    bravo:/export/ws/&/vers2.0 \
       /man        bravo:/export/ws/&/man
   windows \
       /vers1.0    alpha:/export/ws/&/vers1.0 \
       /man        bravo:/export/ws/&/man
   ```

```
files \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /vers2.0    bravo:/export/ws/&/vers2.0 \
    /vers3.0    bravo:/export/ws/&/vers3.0 \
    /man        bravo:/export/ws/&/man
drivers \
    /vers1.0    alpha:/export/ws/&/vers1.0 \
    /man        bravo:/export/ws/&/man
tools \
    /           delta:/export/ws/&
```

Although the map now appears to be much larger, it still contains only the five entries. Each entry is larger because it contains multiple mounts. For instance, a reference to `/ws/compiler` requires three mounts for the `vers1.0`, `vers2.0`, and `man` directories. The backslash at the end of each line tells autofs that the entry is continued onto the next line. In effect, the entry is one long line, though line breaks and some indenting have been used to make it more readable. The `tools` directory contains software development tools for all subprojects, so it is not subject to the same subdirectory structure. The `tools` directory continues to be a single mount.

This arrangement provides the administrator with much flexibility. Software projects are notorious for consuming substantial amounts of disk space. Through the life of the project you might be required to relocate and expand various disk partitions. As long as these changes are reflected in the `auto_ws` map, the users do not need to be notified, as the directory hierarchy under `/ws` is not changed.

Because the servers `alpha` and `bravo` view the same autofs map, any users who log in to these computers can find the `/ws` name space as expected. These users are provided with direct access to local files through loopback mounts instead of NFS mounts.

▼ How to Set Up Different Architectures to Access a Shared Name Space

You need to assemble a shared name space for local executables, and applications, such as spreadsheet tools and word-processing packages. The clients of this name space use several different workstation architectures that require different executable formats. Also, some workstations are running different releases of the operating system.

1. **Create the `auto_local` map with the `nistbladm` command.**

   See the *System Administration Guide: Naming and Directory Services (FNS and NIS+)*.

2. **Choose a single, site-specific name for the shared name space so that files and directories that belong to this space are easily identifiable.**

   For example, if you choose `/usr/local` as the name, the path `/usr/local/bin` is obviously a part of this name space.

3. **For ease of user community recognition, create an autofs indirect map and mount it at `/usr/local`. Set up the following entry in the NIS+ (or NIS) `auto_master` map:**

   ```
   /usr/local     auto_local     -ro
   ```

   Notice that the `-ro` mount option implies that clients will not be able to write to any files or directories.

4. **Export the appropriate directory on the server.**

5. **Include a `bin` entry in the auto_local map.**

   Your directory structure looks like this:

   ```
   bin      aa:/export/local/bin
   ```

   To satisfy the need to serve clients of different architectures, references to the `bin` directory need to be directed to different directories on the server, depending on the clients' architecture type.

6. **To serve clients of different architectures, change the entry by adding the autofs `CPU` variable.**

   ```
   bin      aa:/export/local/bin/$CPU
   ```

   - For SPARC clients – Place executables in `/export/local/bin/sparc`
   - For IA clients – Place executables in `/export/local/bin/i386`

▼ How to Support Incompatible Client Operating System Versions

1. **Combine the architecture type with a variable that determines the operating system type of the client.**

   The autofs `OSREL` variable can be combined with the `CPU` variable to form a name that determines both CPU type and OS release.

2. **Create the following map entry.**

   ```
   bin      aa:/export/local/bin/$CPU$OSREL
   ```

   For clients running version 5.6 of the operating system, export the following file systems:

   - For SPARC clients – Export `/export/local/bin/sparc5.6`
   - For IA clients – Place executables in `/export/local/bin/i3865.6`

▼ How to Replicate Shared Files Across Several Servers

The best way to share replicated file systems that are read-only is to use failover. See "Client-Side Failover" on page 216 for a discussion of failover.

1. **Become superuser.**
2. **Modify the entry in the autofs maps.**

   Create the list of all replica servers as a comma-separated list, such as:

   ```
   bin     aa,bb,cc,dd:/export/local/bin/$CPU
   ```

   Autofs chooses the nearest server. If a server has several network interfaces, list each interface. Autofs chooses the nearest interface to the client, avoiding unnecessary routing of NFS traffic.

▼ How to Apply Security Restrictions

1. **Become superuser.**
2. **Create the following entry in the name service `auto_master` file, either NIS or NIS+:**

   ```
   /home     auto_home     -nosuid
   ```

   The `nosuid` option prevents users from creating files with the `setuid` or `setgid` bit set.

   This entry overrides the entry for `/home` in a generic local `/etc/auto_master` file (see the previous example) because the `+auto_master` reference to the external name service map occurs before the `/home` entry in the file. If the entries in the `auto_home` map include mount options, the `nosuid` option is overwritten, so either no options should be used in the `auto_home` map or the `nosuid` option must be included with each entry.

Note – Do not mount the home directory disk partitions on or under `/home` on the server.

▼ How to Use a Public File Handle With Autofs

1. **Become superuser.**
2. **Create an entry in the autofs map like:**

   ```
   /usr/local     -ro,public     bee:/export/share/local
   ```

The `public` option forces the public handle to be used. If the NFS server does not support a public file handle, the mount will fail.

▼ How to Use NFS URLs With Autofs

1. **Become superuser.**
2. **Create an autofs entry like:**

```
/usr/local      -ro     nfs://bee/export/share/local
```

The service tries to use the public file handle on the NFS server, but if the server does not support a public file handle, the MOUNT protocol is used.

Disabling Autofs Browsability

Starting with the Solaris 2.6 release, the default version of `/etc/auto_master` that is installed has the `-nobrowse` option added to the entries for `/home` and `/net`. In addition, the upgrade procedure adds the `-nobrowse` option to the `/home` and `/net` entries in `/etc/auto_master` if these entries have not been modified. However, it might be necessary to make these changes manually or to turn off browsability for site-specific autofs mount points after the installation.

You can turn off the browsability feature in several ways. Disable it using a command-line option to the `automountd` daemon, which completely disables autofs browsability for the client. Or disable it for each map entry on all clients using the autofs maps in either a NIS or NIS+ name space, or for each map entry on each client, using local autofs maps if no network-wide name space is being used.

▼ How to Completely Disable Autofs Browsability on a Single NFS Client

1. **Become superuser.**
2. **Add the `-n` option to the startup script.**

   As `root`, edit the `/etc/init.d/autofs` script and add the `-n` option to the line that starts the `automountd` daemon:

```
/usr/lib/autofs/automountd -n \
    < /dev/null > /dev/console 2>&1    # start daemon
```

3. **Restart the autofs service.**

```
# /etc/init.d/autofs stop
# /etc/init.d/autofs start
```

▼ How to Disable Autofs Browsability for All Clients

To disable browsability for all clients, you must employ a name service such as NIS or NIS+. Otherwise, you need to manually edit the automounter maps on each client. In this example, the browsability of the `/home` directory is disabled. You must follow this procedure for each indirect autofs node that needs to be disabled.

1. **Add the `-nobrowse` option to the `/home` entry in the name service `auto_master` file.**

```
/home     auto_home     -nobrowse
```

2. **On all clients: run the `automount` command.**

   The new behavior takes effect after running the `automount` command on the client systems or after a reboot.

```
# /usr/sbin/automount
```

▼ How to Disable Autofs Browsability on an NFS Client

In this example, browsability of the `/net` directory is disabled. The same procedure can be used for `/home` or any other autofs mount points.

1. **Check the `automount` entry in `/etc/nsswitch.conf`.**

   For local file entries to take precedence, the entry in the name service switch file should list `files` before the name service. For example:

```
automount:  files nisplus
```

   This is the default configuration in a standard Solaris installation.

2. **Check the position of the `+auto_master` entry in `/etc/auto_master`.**

   For additions to the local files to take precedence over the entries in the name space, the `+auto_master` entry must be moved below `/net`:

```
# Master map for automounter
#
/net    -hosts     -nosuid
/home   auto_home
/xfn    -xfn
+auto_master
```

A standard configuration places the `+auto_master` entry at the top of the file. This prevents any local changes from being used.

3. **Add the `-nobrowse` option to the `/net` entry in the `/etc/auto_master` file.**

```
/net     -hosts     -nosuid,nobrowse
```

4. **On all clients: run the `automount` command.**

   The new behavior takes effect after running the `automount` command on the client systems or after a reboot.

```
# /usr/sbin/automount
```

Strategies for NFS Troubleshooting

When tracking down an NFS problem, keep in mind the main points of possible failure: the server, the client, and the network. The strategy outlined in this section tries to isolate each individual component to find the one that is not working. In all cases, the `mountd` and `nfsd` daemons must be running on the server for remote mounts to succeed.

Note – The `mountd` and `nfsd` daemons start automatically at boot time only if NFS share entries are in the `/etc/dfs/dfstab` file. Therefore, `mountd` and `nfsd` must be started manually when setting up sharing for the first time.

The `-intr` option is set by default for all mounts. If a program hangs with a "server not responding" message, you can kill it with the keyboard interrupt Control-c.

When the network or server has problems, programs that access hard-mounted remote files fail differently than those that access soft-mounted remote files. Hard-mounted remote file systems cause the client's kernel to retry the requests until the server responds again. Soft-mounted remote file systems cause the client's system calls to return an error after trying for awhile. Because these errors can result in unexpected application errors and data corruption, avoid soft-mounting.

When a file system is hard mounted, a program that tries to access it hangs if the server fails to respond. In this case, the NFS system displays the following message on the console:

```
NFS server hostname not responding still trying
```

When the server finally responds, the following message appears on the console:

```
NFS server hostname ok
```

A program accessing a soft-mounted file system whose server is not responding generates the following message:

`NFS` *operation* `failed for server` *hostname*`: error # (`*error_message*`)`

Note – Because of possible errors, do not soft-mount file systems with read-write data or file systems from which executables are run. Writable data could be corrupted if the application ignores the errors. Mounted executables might not load properly and can fail.

NFS Troubleshooting Procedures

To determine where the NFS service has failed, you need to follow several procedures to isolate the failure. Check for the following items:

- Can the client reach the server?
- Can the client contact the NFS services on the server?
- Are the NFS services running on the server?

In the process of checking these items, it might become apparent that other portions of the network are not functioning, such as the name service or the physical network hardware. The *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)* contains debugging procedures for several name services. Also, during the process it might become obvious that the problem isn't at the client end (for instance, if you get at least one trouble call from every subnet in your work area). In this case, it is much more timely to assume that the problem is the server or the network hardware near the server, and start the debugging process at the server, not at the client.

▼ How to Check Connectivity on an NFS Client

1. **Check that the NFS server is reachable from the client. On the client, type the following command.**

   ```
   % /usr/sbin/ping bee
   bee is alive
   ```

   If the command reports that the server is alive, remotely check the NFS server (see "How to Check the NFS Server Remotely" on page 178).

2. **If the server is not reachable from the client, make sure that the local name service is running.**

   For NIS+ clients type the following:

```
% /usr/lib/nis/nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

3. **If the name service is running, make sure that the client has received the correct host information by typing the following:**

```
% /usr/bin/getent hosts bee
129.144.83.117    bee.eng.acme.com
```

4. **If the host information is correct, but the server is not reachable from the client, run the `ping` command from another client.**

   If the command run from a second client fails, see "How to Verify the NFS Service on the Server" on page 179.

5. **If the server is reachable from the second client, use `ping` to check connectivity of the first client to other systems on the local net.**

   If this fails, check the networking software configuration on the client (`/etc/netmasks`, `/etc/nsswitch.conf`, and so forth).

6. **If the software is correct, check the networking hardware.**

   Try moving the client onto a second net drop.

▼ How to Check the NFS Server Remotely

1. **Check that the NFS services have started on the NFS server by typing the following command:**

```
% rpcinfo -s bee|egrep 'nfs|mountd'
 100003  3,2    tcp,udp                          nfs     superuser
 100005  3,2,1  ticots,ticotsord,tcp,ticlts,udp  mountd  superuser
```

   If the daemons have not been started, see "How to Restart NFS Services" on page 180.

2. **Check that the server's `nfsd` processes are responding.**

   On the client, type the following command to test the UDP NFS connections from the server.

```
% /usr/bin/rpcinfo -u bee nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
```

   If the server is running, it prints a list of program and version numbers. Using the `-t` option tests the TCP connection. If this fails, skip to "How to Verify the NFS Service on the Server" on page 179.

3. **Check that the server's `mountd` is responding, by typing the following command.**

```
% /usr/bin/rpcinfo -u bee mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
```

If the server is running, it prints a list of program and version numbers associated with the UDP protocol. Using the `-t` option tests the TCP connection. If either attempt fails, skip to "How to Verify the NFS Service on the Server" on page 179.

4. **Check the local autofs service if it is being used:**

```
% cd /net/wasp
```

Choose a `/net` or `/home` mount point that you know should work properly. If this doesn't work, then as root on the client, type the following to restart the autofs service:

```
# /etc/init.d/autofs stop
# /etc/init.d/autofs start
```

5. **Verify that file system is shared as expected on the server.**

```
% /usr/sbin/showmount -e bee
/usr/src                                    eng
/export/share/man                       (everyone)
```

Check the entry on the server and the local mount entry for errors. Also check the name space. In this instance, if the first client is not in the `eng` netgroup, that client would not be able to mount the `/usr/src` file system.

Check all entries that include mounting information in all of the local files. The list includes `/etc/vfstab` and all the `/etc/auto_*` files.

▼ How to Verify the NFS Service on the Server

1. **Become superuser.**

2. **Check that the server can reach the clients.**

```
# ping lilac
lilac is alive
```

3. **If the client is not reachable from the server, make sure that the local name service is running. For NIS+ clients type the following:**

```
% /usr/lib/nis/nisping -u
Last updates for directory eng.acme.com. :
Master server is eng-master.acme.com.
        Last update occurred at Mon Jun  5 11:16:10 1995

Replica server is eng1-replica-58.acme.com.
        Last Update seen was Mon Jun  5 11:16:10 1995
```

4. **If the name service is running, check the networking software configuration on the server (`/etc/netmasks`, `/etc/nsswitch.conf`, and so forth).**

5. **Type the following command to check whether the `nfsd` daemon is running.**

```
# rpcinfo -u localhost nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
# ps -ef | grep nfsd
root    232      1  0  Apr 07     ?     0:01 /usr/lib/nfs/nfsd -a 16
root   3127   2462  1  09:32:57  pts/3  0:00 grep nfsd
```

If the server is running, it prints a list of program and version numbers associated with the UDP protocol. Also use the `-t` option with `rpcinfo` to check the TCP connection. If these commands fail, restart the NFS service (see "How to Restart NFS Services" on page 180).

6. **Type the following command to check whether the `mountd` daemon is running.**

```
# /usr/bin/rpcinfo -u localhost mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
# ps -ef | grep mountd
root    145      1 0 Apr 07  ?     21:57 /usr/lib/autofs/automountd
root    234      1 0 Apr 07  ?     0:04  /usr/lib/nfs/mountd
root   3084 2462 1 09:30:20 pts/3  0:00  grep mountd
```

If the server is running, it prints a list of program and version numbers associated with the UDP protocol. Also use the `-t` option with `rpcinfo` to check the TCP connection. If these commands fail, restart the NFS service (see "How to Restart NFS Services" on page 180).

7. **Type the following command to check whether the `rpcbind` daemon is running.**

```
# /usr/bin/rpcinfo -u localhost rpcbind
program 100000 version 1 ready and waiting
program 100000 version 2 ready and waiting
program 100000 version 3 ready and waiting
```

If the server is running, it prints a list of program and version numbers associated with the UDP protocol. If `rpcbind` seems to be hung, either reboot the server or follow the steps in "How to Warm-Start `rpcbind`" on page 181.

▼ How to Restart NFS Services

1. **Become superuser.**

2. **To enable daemons without rebooting, type the following commands.**

```
# /etc/init.d/nfs.server stop
# /etc/init.d/nfs.server start
```

This stops the daemons and restarts them, if there is an entry in /etc/dfs/dfstab.

▼ How to Warm-Start rpcbind

If the NFS server cannot be rebooted because of work in progress, it is possible to restart rpcbind without having to restart all of the services that use RPC by completing a warm start as described in this procedure.

1. **Become superuser.**
2. **Determine the PID for rpcbind.**

   Run ps to get the PID (which is the value in the second column).

   ```
   # ps -ef |grep rpcbind
       root   115     1  0   May 31 ?        0:14 /usr/sbin/rpcbind
       root 13000  6944  0 11:11:15 pts/3    0:00 grep rpcbind
   ```

3. **Send a SIGTERM signal to the rpcbind process.**

   In this example, term is the signal that is to be sent and 115 is the PID for the program (see the kill(1) man page). This causes rpcbind to create a list of the current registered services in /tmp/portmap.file and /tmp/rpcbind.file.

   ```
   # kill -s term 115
   ```

   Note – If you do not kill the rpcbind process with the -s term option, you cannot complete a warm start of rpcbind and must reboot the server to restore service.

4. **Restart rpcbind.**

   Do a warm restart of the command so that the files created by the kill command are consulted, and the process resumes without requiring that all of the RPC services be restarted (see the rpcbind(1M) man page).

   ```
   # /usr/sbin/rpcbind -w
   ```

▼ Identifying Which Host Is Providing NFS File Service

Run the nfsstat command with the -m option to gather current NFS information. The name of the current server is printed after "currserver=".

```
% nfsstat -m
/usr/local from bee,wasp:/export/share/local
 Flags: vers=3,proto=tcp,sec=sys,hard,intr,llock,link,synlink,
        acl,rsize=32768,wsize=32678,retrans=5
```

```
Failover: noresponse=0, failover=0, remap=0, currserver=bee
```

▼ How to Verify Options Used With the mount Command

In the Solaris 2.6 release and in any versions of the `mount` command that were patched after the 2.6 release, no warning is issued for invalid options. The following procedure helps determine whether the options that were supplied either on the command line or through `/etc/vfstab` were valid.

For this example, assume that the following command has been run:

```
# mount -F nfs -o ro,vers=2 bee:/export/share/local /mnt
```

1. **Verify the options, by running the following command.**

```
% nfsstat -m
/mnt from bee:/export/share/local
Flags:  vers=2,proto=tcp,sec=sys,hard,intr,dynamic,acl,rsize=8192,wsize=8192,
        retrans=5
```

The file system from `bee` has been mounted with the protocol version set to `2`. Unfortunately, the `nfsstat` command does not display information about all of the options, but using the `nfsstat` command is the most accurate way to verify the options.

2. **Check the entry in `/etc/mnttab`.**

The `mount` command does not allow invalid options to be added to the mount table, so verifying that the options listed in the file match those listed on the command line is a way to check those options not reported by the `nfsstat` command.

```
# grep bee /etc/mnttab
bee:/export/share/local /mnt nfs    ro,vers=2,dev=2b0005e 859934818
```

Troubleshooting Autofs

Occasionally, you might encounter problems with autofs. This section should make the problem-solving process easier. It is divided into two subsections.

This section presents a list of the error messages that autofs generates. The list is divided into two parts:

- Error messages generated by the verbose (`-v`) option of `automount`
- Error messages that might appear at any time

Each error message is followed by a description and probable cause of the message.

When troubleshooting, start the autofs programs with the verbose (`-v`) option; otherwise, you might experience problems without knowing why.

The following paragraphs are labeled with the error message you are likely to see if autofs fails, and a description of the possible problem.

Error Messages Generated by `automount -v`

`bad key` *key* `in direct map` *mapname*
: While scanning a direct map, autofs has found an entry key without a prefixed `/`. Keys in direct maps must be full path names.

`bad key` *key* `in indirect map` *mapname*
: While scanning an indirect map, autofs has found an entry key containing a `/`. Indirect map keys must be simple names—not path names.

`can't mount` *server*`:`*pathname: reason*
: The mount daemon on the server refuses to provide a file handle for *server:pathname*. Check the export table on server.

`couldn't create mount point` *mountpoint*`:` *reason*
: Autofs was unable to create a mount point required for a mount. This most frequently occurs when attempting to hierarchically mount all of a server's exported file systems. A required mount point can exist only in a file system that cannot be mounted (it cannot be exported) and it cannot be created because the exported parent file system is exported read-only.

`leading space in map entry` *entry* `text in` *mapname*
: Autofs has discovered an entry in an automount map that contains leading spaces. This is usually an indication of an improperly continued map entry, for example:

```
fake
/blat          frobz:/usr/frotz
```

  In this example, the warning is generated when autofs encounters the second line because the first line should be terminated with a backslash (`\`).

mapname`: Not found`
: The required map cannot be located. This message is produced only when the `-v` option is used. Check the spelling and path name of the map name.

`remount` *server*`:`*pathname* `on` *mountpoint*`: server not responding`
: Autofs has failed to remount a file system it previously unmounted.

`WARNING: mountpoint already mounted on`
Autofs is attempting to mount over an existing mount point. This means an internal error occurred in autofs (an anomaly).

Miscellaneous Error Messages

`dir mountpoint must start with '/'`
Automounter mount point must be given as full path name. Check the spelling and path name of the mount point.

`hierarchical mountpoints: pathname1 and pathname2`
Autofs does not allow its mount points to have a hierarchical relationship. An autofs mount point must not be contained within another automounted file system.

`host server not responding`
Autofs attempted to contact *server*, but received no response.

hostname`: exports: rpc_err`
Error getting export list from *hostname*. This indicates a server or network problem.

`map mapname, key key: bad`
The map entry is malformed, and autofs cannot interpret it. Recheck the entry; perhaps the entry has characters that need escaping.

mapname`: nis_err`
Error in looking up an entry in a NIS map. This can indicate NIS problems.

`mount of server:pathname on mountpoint:reason`
Autofs failed to do a mount. This can indicate a server or network problem.

mountpoint`: Not a directory`
Autofs cannot mount itself on *mountpoint* because it is not a directory. Check the spelling and path name of the mount point.

`nfscast: cannot send packet: reason`
Autofs cannot send a query packet to a server in a list of replicated file system locations.

`nfscast: cannot receive reply: reason`
Autofs cannot receive replies from any of the servers in a list of replicated file system locations.

`nfscast: select: reason`
All these error messages indicate problems attempting to `ping` servers for a replicated file system. This can indicate a network problem.

`pathconf: no info for server:pathname`
Autofs failed to get `pathconf` information for path name (see the `fpathconf`(2) man page).

`pathconf: server: server not responding`
Autofs is unable to contact the mount daemon on *server* that provides the information to `pathconf()`.

Other Errors With Autofs

If the `/etc/auto*` files have the execute bit set, the automounter tries to execute the maps, which creates messages like:

```
/etc/auto_home: +auto_home: not found
```

In this case, the `auto_home` file has incorrect permissions. Each entry in the file will generate an error message much like this one. The permissions to the file should be reset by typing the following command:

```
# chmod 644 /etc/auto_home
```

NFS Error Messages

This section shows an error message followed by a description of the conditions that should create the error and at least one way of fixing the problem.

`Bad argument specified with index option - must be a file`
You must include a file name with the `-index` option. You cannot use directory names.

`Cannot establish NFS service over /dev/tcp: transport setup problem`
This message is often created when the services information in the name space has not been updated. It can also be reported for UDP. To fix this problem, you must update the services data in the name space. For NIS+ the entries should be:

```
nfsd nfsd tcp 2049 NFS server daemon
nfsd nfsd udp 2049 NFS server daemon
```

For NIS and `/etc/services`, the entries should be:

```
nfsd    2049/tcp    nfs    # NFS server daemon
nfsd    2049/udp    nfs    # NFS server daemon
```

`Cannot use index option without public option`
Include the `public` option with the index option in the share command. You must define the public file handle for the `-index` option to work.

Note – The Solaris 2.5.1 release required that the public file handle be set using the share command. A change in the Solaris 2.6 release sets the public file handle to be `/` by default. This error message is no longer relevant.

`Could not use public filehandle in request to` *server*
This message is displayed if the public option is specified but the NFS server does not support the public file handle. In this case, the mount will fail. To remedy this situation, either try the mount request without using the public file handle or reconfigure the NFS server to support the public file handle.

`NOTICE: NFS3: failing over from` *host1* `to` *host2*
This message is displayed on the console when a failover occurs. It is an advisory message only.

filename`: File too large`
An NFS version 2 client is trying to access a file that is over 2 Gbytes.

`mount: ... server not responding:RPC_PMAP_FAILURE - RPC_TIMED_OUT`
The server sharing the file system you are trying to mount is down or unreachable, at the wrong run level, or its `rpcbind` is dead or hung.

`mount: ... server not responding: RPC_PROG_NOT_REGISTERED`
Mount registered with `rpcbind`, but the NFS mount daemon `mountd` is not registered.

`mount: ... No such file or directory`
Either the remote directory or the local directory does not exist. Check the spelling of the directory names. Run `ls` on both directories.

`mount: ...: Permission denied`
Your computer name might not be in the list of clients or netgroup allowed access to the file system you want to mount. Use `showmount -e` to verify the access list.

`nfs mount: ignoring invalid option "-`*option*`"`
The *-option* flag is not valid. Refer to the `mount_nfs`(1M) man page to verify the required syntax.

Note – This error message is not displayed when running any version of the `mount` command included in a Solaris release from 2.6 to the current release or in earlier versions that have been patched.

`nfs mount: NFS can't support "nolargefiles"`
An NFS client has attempted to mount a file system from an NFS server using the `-nolargefiles` option. This option is not supported for NFS file system types.

`nfs mount: NFS V2 can't support "largefiles"`
The NFS version 2 protocol cannot handle large files. You must use version 3 if access to large files is required.

`NFS server` *hostname* `not responding still trying`
If programs hang while doing file-related work, your NFS server might be dead. This message indicates that NFS server *hostname* is down or that a problem has occurred with the server or the network. If failover is being used, *hostname* is a list of servers. Start with "How to Check Connectivity on an NFS Client" on page 177.

`NFS fsstat failed for server` *hostname*`: RPC: Authentication error`
This error can be caused by many situations. One of the most difficult to debug is when this occurs because a user is in too many groups. Currently, a user can be in as many as 16 groups but no more if they are accessing files through NFS mounts. If a user must have the functionality of being in more than 16 groups and if at least the Solaris 2.5 release is running on the NFS server and the NFS clients, then use ACLs to provide the needed access privileges.

`port` *number* `in nfs URL not the same as port` *number* `in port option`
The port number included in the NFS URL must match the port number included with the `-port` option to mount. If the port numbers do not match, the mount will fail. Either change the command to make the port numbers the same or do not specify the port number that is incorrect. Usually, you do not need to specify the port number both in the NFS URL and with the `-port` option.

`replicas must have the same version`
For NFS failover to function properly, the NFS servers that are replicas must support the same version of the NFS protocol. Mixing version 2 and version 3 servers is not allowed.

`replicated mounts must be read-only`
NFS failover does not work on file systems that are mounted read-write. Mounting the file system read-write increases the likelihood that a file will change. NFS failover depends on the file systems being identical.

`replicated mounts must not be soft`
: Replicated mounts require that you wait for a timeout before failover occurs. The `soft` option requires that the mount fail immediately when a timeout starts, so you cannot include the `-soft` option with a replicated mount.

`share_nfs: Cannot share more than one filesystem with 'public' option`
: Check that the `/etc/dfs/dfstab` file has only one file system selected to be shared with the `-public` option. Only one public file handle can be established per server, so only one file system per server can be shared with this option.

`WARNING: No network locking on` *hostname*`:`*path*`: contact admin to install server change`
: An NFS client has unsuccessfully attempted to establish a connection with the network lock manager on an NFS server. Rather than fail the mount, this warning is generated to warn you that locking will not work.

CHAPTER 15

Accessing Remote File Systems Reference

This chapter provides an introduction to the NFS commands. This chapter also provides information about all of the pieces of the NFS environment and how these pieces work together.

NFS Files

You need several files to support NFS activities on any computer. Many of these files are ASCII, but some of them are data files. Table 15–1 lists these files and their functions.

TABLE 15–1 NFS Files

File Name	Function
`/etc/default/fs`	Lists the default file system type for local file systems.
`/etc/dfs/dfstab`	Lists the local resources to be shared.
`/etc/dfs/fstypes`	Lists the default file-system types for remote file systems.

TABLE 15–1 NFS Files *(Continued)*

File Name	Function
`/etc/default/nfslogd`	Lists configuration information for the NFS server logging daemon, `nfslogd`.
`/etc/dfs/sharetab`	Lists the resources (local and remote) that are shared (see the `sharetab`(4) man page); do not edit this file.
`/etc/mnttab`	Lists file systems that are currently mounted, including automounted directories (see the `mnttab`(4) man page); do not edit this file.
`/etc/netconfig`	Lists the transport protocols; do not edit this file.
`/etc/nfs/nfslog.conf`	Lists general configuration information for NFS server logging.
`/etc/nfs/nfslogtab`	Lists information for log post-processing by `nfslogd`; do not edit this file.
`/etc/nfssec.conf`	Lists NFS security services; do not edit this file.
`/etc/rmtab`	Lists file systems remotely mounted by NFS clients (see the `rmtab`(4) man page); do not edit this file.
`/etc/vfstab`	Defines file systems to be mounted locally (see the `vfstab`(4) man page).

The first entry in `/etc/dfs/fstypes` is often used as the default file-system type for remote file systems. This entry defines the NFS file-system type as the default.

Only one entry is in `/etc/default/fs`: the default file-system type for local disks. You can determine the file-system types that are supported on a client or server by checking the files in `/kernel/fs`.

/etc/default/nfslogd

This file defines some of the parameters used when using NFS server logging. The following parameters can be defined.

CYCLE_FREQUENCY
: Determines the number of hours that must pass before the log files are cycled. The default value is 24 hours. This option is used to prevent the log files from growing too large.

IDLE_TIME
: Sets the number of seconds `nfslogd` should sleep before checking for more information in the buffer file. It also determines how often the configuration file is checked. This parameter, along with MIN_PROCESSING_SIZE, determines how

often the buffer file is processed. The default value is 300 seconds. Increasing this number can improve performance by reducing the number of checks.

MAPPING_UPDATE_INTERVAL
Specifies the number of seconds between updates of the records in the file-handle-to-path mapping tables. The default value is 86400 seconds or one day. This parameter helps keep the file-handle-to-path mapping tables up-to-date without having to continually update the tables.

MAX_LOGS_PRESERVE
Determines the number of log files to be saved. The default value is 10.

MIN_PROCESSING_SIZE
Sets the minimum number of bytes that the buffer file must reach before processing and writing to the log file. This parameter, along with IDLE_TIME, determines how often the buffer file is processed. The default value for is 524288 bytes. Increasing this number can improve performance by reducing the number of times the buffer file is processed.

PRUNE_TIMEOUT
Selects the number of hours that must pass before a file-handle-to-path mapping record times out and can be pruned. The default value is 168 hours or 7 days.

UMASK
Specifies the permissions for the log files that are created by `nfslogd`. The default value is 0137.

/etc/nfs/nfslog.conf

This file defines the path, file names, and type of logging to be used by `nfslogd`. Each definition is associated with a *tag*. Starting NFS server logging requires that you identify the *tag* for each file system. The global tag defines the default values. The following parameters can be used with each tag as needed.

defaultdir=*path*
Specifies the default directory path for the logging files.

log=*path/filename*
Sets the path and file name for the log files.

fhtable=*path/filename*
Selects the path and file name for the file-handle-to-path database files.

buffer=*path/filename*
Determines the path and file name for the buffer files.

logformat=*basic* | *extended*
Selects the format to be used when creating user-readable log files. The basic format produces a log file similar to some `ftpd` daemons. The extended format gives a more detailed view.

For the parameters that can specify both the path and the file name, if the path is not specified, the path defined by `defaultdir` is used. Also, you can override `defaultdir` by using an absolute path.

To make identifying the files easier, place the files in separate directories. Here is an example of the changes needed.

```
% 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

publicftp log=logs/nfslog fhtable=fh/fhtables buffer=buffers/workbuffer
```

In this example, any file system shared with `log=publicftp` would use the following values: the default directory would be `/var/nfs`, log files would be stored in `/var/nfs/logs/nfslog*`, file-handle-to-path database tables would be stored in `/var/nfs/fh/fhtables`, and buffer files would be stored in `/var/nfs/buffers/workbuffer`.

NFS Daemons

To support NFS activities, several daemons are started when a system goes into run level 3 or multiuser mode. The `mountd` and `nfsd` daemons are run on systems that are NFS servers. The automatic startup of the server daemons depends on the existence of entries labeled with the NFS file-system type in `/etc/dfs/sharetab`. The `lockd` and `statd` daemons are run on NFS clients and servers to support NFS file locking.

automountd

This daemon handles the mount and unmount requests from the autofs service. The syntax of the command is:

`automountd [ -Tnv ] [ -D` *name=value* `]`

where `-T` enables tracing, `-n` disables browsing on all autofs nodes, `-v` selects to log all status messages to the console, and `-D` *name=value* substitutes *value* for the automount map variable indicated by *name*. The default value for the automount map is `/etc/auto_master`. Use the `-T` option for troubleshooting.

lockd

This daemon supports record-locking operations on NFS files. It manages RPC connections between the client and the server for the Network Lock Manager (NLM) protocol. The daemon is normally started without any options. You can use three options with this command (see the `lockd`(1M) man page).

The `-g` *graceperiod* option selects the number of seconds that the clients have to reclaim locks after a server reboot. During this time, the NFS server only processes reclaims of old locks. All other requests for service must wait until the grace period is over. This option affects the NFS server-side response, so it can be changed only on an NFS server. The default value for *graceperiod* is 45 seconds. Reducing this value means that NFS clients can resume operation more quickly after a server reboot, but a reduction increases the chances that a client might not be able to recover all its locks.

The `-t` *timeout* option selects the number of seconds to wait before retransmitting a lock request to the remote server. This option affects the NFS client-side service. The default value for *timeout* is 15 seconds. Decreasing the *timeout* value can improve response time for NFS clients on a noisy network, but it can cause additional server load by increasing the frequency of lock requests.

The *nthreads* option specifies the maximum number of concurrent threads that the server handles per connection. Base the value for *nthreads* on the load expected on the NFS server. The default value is 20. Because each NFS client using TCP uses a single connection with the NFS server, each TCP client is granted the ability to use up to 20 concurrent threads on the server. All NFS clients using UDP share a single connection with the NFS server. Under these conditions it might be necessary to increase the number of threads available for the UDP connection. A minimum calculation would be to allow two threads for each UDP client, but this is specific to the workload on the client, so two threads per client might not be sufficient. The disadvantage to using more threads is that when the threads are used, more memory is used on the NFS server, but if the threads are never used, increasing *nthreads* will have no effect.

mountd

This is a remote procedure call (RPC) server that handles file-system mount requests from remote systems and provides access control. It checks `/etc/dfs/sharetab` to determine which file systems are available for remote mounting and which systems

are allowed to do the remote mounting. You can use two options with this command (see the `mountd`(1M) man page): `-v` and `-r` .

The `-v` option runs the command in verbose mode. Each time an NFS server determines the access a client should get, a message is printed on the console. The information generated can be useful when trying to determine why a client cannot access a file system.

The `-r` option rejects all future mount requests from clients. This does not affect clients that already have a file system mounted.

nfsd

This daemon handles other client file-system requests. You can use several options with this command. See the `nfsd`(1M) man page for a complete listing.

The `-l` option sets the connection queue length for the NFS/TCP over connection-oriented transports. The default value is 32 entries.

The `-c` #*_conn* option selects the maximum number of connections per connection-oriented transport. The default value for #*_conn* is unlimited.

The *nservers* option is the maximum number of concurrent requests that a server can handle. The default value for *nservers* is 1, but the startup scripts select 16.

Unlike older versions of this daemon, `nfsd` does not spawn multiple copies to handle concurrent requests. Checking the process table with `ps` only shows one copy of the daemon running.

nfslogd

This daemon provides operational logging. NFS operations against a server are logged based on the configuration options defined in `/etc/default/nfslogd`. When NFS server logging is enabled, records of all RPC operations on a selected file system are written into a buffer file by the kernel. Then `nfslogd` post-processes these requests. The name service switch is used to help map UIDs to logins and IP addresses to host names. The number is recorded if no match can be found through the identified name services.

Mapping of file handles to path names is also handled by `nfslogd`. The daemon keeps track of these mappings in a file-handle-to-path mapping table. One mapping table exists for each tag identified in `/etc/nfs/nfslogd`. After post-processing, the records are written out to ASCII log files.

statd

This daemon works with `lockd` to provide crash and recovery functions for the lock manager. It tracks the clients that hold locks on an NFS server. If a server crashes, on rebooting `statd` on the server contacts `statd` on the client. The client `statd` can then attempt to reclaim any locks on the server. The client `statd` also informs the server `statd` when a client has crashed, so that the client's locks on the server can be cleared. There are no options to select with this daemon. For more information see the `statd`(1M) man page.

In the Solaris 7 release, the way that `statd` keeps track of the clients has been improved. In all earlier Solaris releases, `statd` created files in `/var/statmon/sm` for each client using the client's unqualified host name. This caused problems if you had two clients in different domains that shared a hos tname, or if there were clients that were not resident in the same domain as the NFS server. Because the unqualified host name only lists the host name, without any domain or IP-address information, the older version of `statd` had no way to differentiate between these types of clients. To fix this problem, the Solaris 7 `statd` creates a symbolic link in `/var/statmon/sm` to the unqualified host name using the IP address of the client. The new link will look like:

```
# ls -l /var/statmon/sm
lrwxrwxrwx   1 root          11 Apr 29 16:32 ipv4.192.9.200.1 -> myhost
--w-------   1 root          11 Apr 29 16:32 myhost
```

In this example, the client host name is `myhost` and the clients' IP address is `192.9.200.1`. If another host with the name `myhost` were mounting a file system, there would be two symbolic links to the host name.

NFS Commands

These commands must be run as root to be fully effective, but requests for information can be made by all users:

- "unshare" on page 206
- "unshareall" on page 207

automount

This command installs autofs mount points and associates the information in the automaster files with each mount point. The syntax of the command is:

`automount [ -t` *duration* `] [ -v ]`

where `-t` *duration* sets the time, in seconds, that a file system is to remain mounted, and `-v` selects the verbose mode. Running this command in the verbose mode allows for easier troubleshooting.

If not specifically set, the value for duration is set to 5 minutes. In most circumstances this is a good value; however, on systems that have many automounted file systems, you might need to increase the duration value. In particular, if a server has many users active, checking the automounted file systems every 5 minutes can be inefficient. Checking the autofs file systems every 1800 seconds (or 30 minutes) could be more optimal. By not unmounting the file systems every 5 minutes, it is possible that `/etc/mnttab`, which is checked by `df`, can become large. The output from `df` can be filtered by using the `-F` option (see the `df`(1M) man page) or by using `egrep` to help fix this problem.

Another factor to consider is that adjusting the duration also changes how quickly changes to the automounter maps will be reflected. Changes will not be seen until the file system is unmounted. Refer to "Modifying the Maps" on page 164 for instructions on how to modify automounter maps.

clear_locks

This command enables you to remove all file, record, and share locks for an NFS client. You must be `root` to run this command. From an NFS server you can clear the locks for a specific client and from an NFS client you can clear locks for that client on a specific server. The following example would clear the locks for the NFS client named `tulip` on the current system.

```
# clear_locks tulip
```

Using the `-s` option enables you to specify which NFS host to clear the locks from. It must be run from the NFS client, which created the locks. In this case, the locks from the client would be removed from the NFS server named `bee`.

```
# clear_locks -s bee
```

Caution – This command should only be run when a client crashes and cannot clear its locks. To avoid data corruption problems, do not clear locks for an active client.

mount

With this command, you can attach a named file system, either local or remote, to a specified mount point. For more information, see the `mount`(1M) man page. Used without arguments, `mount` displays a list of file systems that are currently mounted on your computer.

Many types of file systems are included in the standard Solaris installation. Each file-system type has a specific man page that lists the options to `mount` that are appropriate for that file-system type. The man page for NFS file systems is `mount_nfs`(1M); for UFS file systems it is `mount_ufs`(1M); and so forth.

The Solaris 7 release includes the ability to select a path name to mount from an NFS server using an NFS URL instead of the standard `server:/pathname` syntax. See "How to Mount an NFS File System Using an NFS URL" on page 155 for further information.

Caution – The version of the `mount` command included in any Solaris release from 2.6 to the current release, will not warn about options that are not valid. The command silently ignores any options that cannot be interpreted. Make sure to verify all of the options that were used to prevent unexpected behavior.

mount Options for NFS File Systems

The subsequent text lists some of the options that can follow the `-o` flag when mounting an NFS file system.

bg | fg
: These options can be used to select the retry behavior if a mount fails. The `-bg` option causes the mount attempts to be run in the background. The `-fg` option causes the mount attempt to be run in the foreground. The default is `-fg` , which is the best selection for file systems that must be available. It prevents further processing until the mount is complete. `-bg` is a good selection for file systems that are not critical, because the client can do other processing while waiting for the mount request to complete.

forcedirectio
: This option improves performance of sequential reads on large files. Data is copied directly to a user buffer and no caching is done in the kernel on the client. This option is off by default.

largefiles
: This option makes it possible to access files larger than 2 Gbytes on a server running the Solaris 2.6 release. Whether a large file can be accessed can only be controlled on the server, so this option is silently ignored on NFS version 3 mounts. Starting with release 2.6, by default, all UFS file systems are mounted with `-largefiles`. For mounts using the NFS version 2 protocol, the `-largefiles` option causes the mount to fail with an error.

nolargefiles
: This option for UFS mounts guarantees that there are and will be no large files on the file system (see the `mount_ufs`(1M) man page). Because the existence of large files can only be controlled on the NFS server, there is no option for `-nolargefiles` using NFS mounts. Attempts to NFS mount a file system using this option are rejected with an error.

public
: This option forces the use of the public file handle when contacting the NFS server. If the public file handle is supported by the server, the mounting operation is faster because the MOUNT protocol is not used. Also, because the MOUNT protocol is not used, the public option allows mounting to occur through a firewall.

rw | ro
: The `-rw` and `-ro` options indicate whether a file system is to be mounted read-write or read-only. The default is read-write, which is the appropriate option for remote home directories, mail-spooling directories, or other file systems that need to be changed by users. The read-only option is appropriate for directories that should not be changed by users; for example, shared copies of the man pages should not be writable by users.

sec=*mode*
: You can use this option to specify the authentication mechanism to be used during the mount transaction. The value for *mode* can be one of the values shown in Table 15–2. The modes are also defined in `/etc/nfssec.conf`.

TABLE 15–2 NFS Security Modes

Mode	Authentication Service Selected
`krb5`	Kerberos Version 5
`none`	No authentication
`dh`	Diffie-Hellman (DH) authentication
`sys`	Standard UNIX authentication

soft | hard

An NFS file system mounted with the `soft` option returns an error if the server does not respond. The `hard` option causes the mount to continue to retry until the server responds. The default is `hard` , which should be used for most file systems. Applications frequently do not check return values from `soft` -mounted file systems, which can make the application fail or can lead to corrupted files. Even if the application does check, routing problems and other conditions can still confuse the application or lead to file corruption if the `soft` option is used. In most cases the `soft` option should not be used. If a file system is mounted using the `hard` option and becomes unavailable, an application using this file system will hang until the file system becomes available.

Using the `mount` Command

Both of these commands mount an NFS file system from the server `bee` read-only:

```
# mount -F nfs -r bee:/export/share/man /usr/man
```

```
# mount -F nfs -o ro bee:/export/share/man /usr/man
```

This command uses the `-O` option to force the man pages from the server `bee` to be mounted on the local system even if `/usr/man` has already been mounted on:

```
# mount -F nfs -O bee:/export/share/man /usr/man
```

This command uses client failover:

```
# mount -F nfs -r bee,wasp:/export/share/man /usr/man
```

Note – When used from the command line, the listed servers must support the same version of the NFS protocol. Do not mix version 2 and version 3 servers when running `mount` from the command line. You can use mixed servers with autofs, in which case the best subset of version 2 or version 3 servers is used.

Here is an example of using an NFS URL with the `mount` command:

```
# mount -F nfs nfs://bee//export/share/man /usr/man
```

Use the `mount` command with no arguments to display file systems mounted on a client.

```
% mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid on Tues Jan 24 13:20:47 1995
/usr on /dev/dsk/c0t3d0s6 read/write/setuid on Tues Jan 24 13:20:47 1995
/proc on /proc read/write/setuid on Tues Jan 24 13:20:47 1995
/dev/fd on fd read/write/setuid on Tues Jan 24 13:20:47 1995
/tmp on swap read/write on Tues Jan 24 13:20:51 1995
/opt on /dev/dsk/c0t3d0s5 setuid/read/write on Tues Jan 24 13:20:51 1995
/home/kathys on bee:/export/home/bee7/kathys
```

```
intr/noquota/nosuid/remote on Tues Jan 24 13:22:13 1995
```

umount

This command enables you to remove a remote file system that is currently mounted. The `umount` command supports the `-V` option to allow for testing. You might also use the `-a` option to umount several file systems at one time. If *mount_points* are included with the `-a` option, those file systems are unmounted. If no mount points are included, an attempt is made to unmount all file systems listed in `/etc/mnttab`, except for the "required" file systems, such as `/`, `/usr`, `/var`, `/proc`, `/dev/fd`, and `/tmp`. Because the file system is already mounted and should have an entry in `/etc/mnttab`, you do not need to include a flag for the file-system type.

The `-f` option forces a busy file system to be unmounted. You can unhang a client that is hung trying to mount an unmountable file system by using this option.

Caution – Forcing an unmount of a file system can cause data loss if files are being written to.

Using the `umount` Command

This example unmounts a file system mounted on `/usr/man`:

```
# umount /usr/man
```

This example displays the results of running `umount -a -V`:

```
# umount -a -V
umount /home/kathys
umount /opt
umount /home
umount /net
```

Notice that this command does not actually unmount the file systems.

mountall

Use this command to mount all file systems or a specific group of file systems listed in a file-system table. The command provides a way to select the file-system type to be accessed with the `-F` *FSType* option, to select all the remote file systems listed in a file-system table with the `-r` option, and to select all the local file systems with the `-l`

option. Because all file systems labeled as NFS file-system type are remote file systems, some of these options are redundant. For more information, see the `mountall`(1M) man page.

Using the `mountall` Command

These two examples are equivalent:

```
# mountall -F nfs
```

```
# mountall -F nfs -r
```

`umountall`

Use this command to unmount a group of file systems. The `-k` option runs the `fuser -k` *mount_point* command to kill any processes associated with the *mount_point*. The `-s` option indicates that unmount is not to be performed in parallel. `-l` specifies that only local file systems are to be used, and `-r` specifies that only remote file systems are to be used. The `-h` *host* option indicates that all file systems from the named host should be unmounted. You cannot combine the `-h` option with `-l` or `-r` .

Using the `umountall` Command

This command unmounts all file systems that are mounted from remote hosts:

```
# umountall -r
```

This command unmounts all file systems currently mounted from the server `bee`:

```
# umountall -h bee
```

`share`

With this command, you can make a local file system on an NFS server available for mounting. You can also use the `share` command to display a list of the file systems on your system that are currently shared. The NFS server must be running for the `share` command to work. The NFS server software is started automatically during boot if there is an entry in `/etc/dfs/dfstab`. The command does not report an error if the NFS server software is not running, so you must check this yourself.

The objects that can be shared include any directory tree, but each file system hierarchy is limited by the disk slice or partition that the file system is located on. For instance, sharing the root (`/`) file system would not also share `/usr`, unless they are on

the same disk partition or slice. Normal installation places root on slice 0 and `/usr` on slice 6. Also, sharing `/usr` would not share any other local disk partitions that are mounted on subdirectories of `/usr`.

A file system cannot be shared that is part of a larger file system already being shared. For example, if `/usr` and `/usr/local` are on one disk slice, `/usr` can be shared or `/usr/local` can be shared, but if both need to be shared with different share options, `/usr/local` must to be moved to a separate disk slice.

Note – You can gain access to a file system that is shared read-only through the file handle of a file system that is shared read-write if the two file systems are on the same disk slice. It is more secure to place those file systems that need to be read-write on a separate partition or disk slice than the file systems that you need to share read-only.

Non-file System Specific `share` Options

Some of the options that you can include with the `-o` flag are as follows.

rw | ro
: The *pathname* file system is shared read-write or read-only to all clients.

rw=*accesslist*
: The file system is shared read-write to the listed clients only. All other requests are denied. Starting with the Solaris 2.6 release, the list of clients defined in *accesslist* has been expanded. See "Setting Access Lists With the `share` Command" on page 204 for more information. You can use this option to override an `-ro` option.

NFS Specific `share` Options

The options that you can use with NFS file systems include the following.

aclok
: This option enables an NFS server supporting the NFS version 2 protocol to be configured to do access control for NFS version 2 clients. Without this option all clients are given minimal access. With this option the clients have maximal access. For instance, on file systems shared with the `-aclok` option, if anyone has read permissions, everyone does. However, without this option, you can deny access to a client who should have access permissions. Whether it is preferred to permit too much access or too little, depends on the security systems already in place. See "Securing Files (Tasks)" in *System Administration Guide: Security Services* for more information about access control lists (ACLs).

Note – To take advantage of ACLs, it is best to have clients and servers run software that supports the NFS version 3 and NFS_ACL protocols. If the software only supports the NFS version 3 protocol, clients get correct access, but cannot manipulate the ACLs. If the software supports the NFS_ACL protocol, the clients get correct access and can manipulate the ACLs. Starting with release 2.5, the Solaris system supports both protocols.

anon=*uid*

You use *uid* to select the user ID of unauthenticated users. If you set *uid* to `-1`, the server denies access to unauthenticated users. You can grant root access by setting `anon=0`, but this will allow unauthenticated users to have root access, so use the `root` option instead.

index=*filename*

You can use the `-index=`*filename* option to force the loading of a HyperText Markup Language (HTML) file instead of displaying a listing of the directory when a user accesses an NFS URL. This option mimics the action of current browsers if an `index.html` file is found in the directory that the HTTP URL is accessing. This is the equivalent of setting the `DirectoryIndex` option for `httpd`. For instance, if the `dfstab` file entry looks like:

```
share -F nfs -o ro,public,index=index.html /export/web
```

these URLs will display the same information:

```
nfs://<server>/<dir>
nfs://<server>/<dir>/index.html
nfs://<server>//export/web/<dir>
nfs://<server>//export/web/<dir>/index.html
http://<server>/<dir>
http://<server>/<dir>/index.html
```

log=*tag*

This option specifies the tag in `/etc/nfs/nfslog.conf` that contains the NFS server logging configuration information for a file system. This option must be selected to enable NFS server logging.

nosuid

This option signals that all attempts to enable the `setuid` or `setgid` mode should be ignored. NFS clients cannot be able to create files with the `setuid` or `setgid` bits on.

public

The `-public` option has been added to the `share` command to enable WebNFS browsing. Only one file system on a server can be shared with this option.

root=*accesslist*

The server gives root access to the hosts in the list. By default, the server does not give root access to any remote hosts. If the selected security mode is anything other

than `-sec=sys`, you can only include client host names in the *accesslist*. Starting with the Solaris 2.6 release, the list of clients defined in *accesslist* is expanded. See "Setting Access Lists With the `share` Command" on page 204 for more information.

Caution – Granting root access to other hosts has far-reaching security implications; use the `-root=` option with extreme caution.

sec=*mode*[:*mode*]
mode selects the security modes that are needed to get access to the file system. By default, the security mode is UNIX authentication. You can specify multiple modes, but use each security mode only once per command line. Each `-mode` option applies to any subsequent `-rw` , `-ro` , `-rw=` , `-ro=` , `-root=` , and `-window=` options, until another `-mode` is encountered. Using `-sec=none` maps all users to user `nobody`.

window=*value*
value selects the maximum life time in seconds of a credential on the NFS server. The default value is 30000 seconds or 8.3 hours.

Setting Access Lists With the `share` Command

In Solaris releases prior to 2.6, the *accesslist* included with either the `-ro=`, `-rw=` , or `-root=` option of the `share` command were restricted to a list of host names or netgroup names. Starting with the Solaris 2.6 release, the access list can also include a domain name, a subnet number, or an entry to deny access. These extensions should make it easier to control file access control on a single server, without having to change the name space or maintain long lists of clients.

This command provides read-only access for most systems but allows read-write access for `rose` and `lilac`:

```
# share -F nfs -o ro,rw=rose:lilac /usr/src
```

In the next example, read-only access is assigned to any host in the `eng` netgroup. The client `rose` is specifically given read-write access.

```
# share -F nfs -o ro=eng,rw=rose /usr/src
```

Note – You cannot specify both `rw` and `ro` without arguments. If no read-write option is specified, the default is read-write for all clients.

To share one file system with multiple clients, you must enter all options on the same line, because multiple invocations of the `share` command on the same object

"remember" only the last command run. This command enables read-write access to three client systems, but only `rose` and `tulip` are given access to the file system as `root`.

```
# share -F nfs -o rw=rose:lilac:tulip,root=rose:tulip /usr/src
```

When sharing a file system using multiple authentication mechanisms, make sure to include the `-ro`, `-ro=` , `-rw` , `-rw=` , `-root` , and `-window` options after the correct security modes. In this example, UNIX authentication is selected for all hosts in the netgroup named `eng`. These hosts can only mount the file system in read-only mode. The hosts `tulip` and `lilac` can mount the file system read-write if they use Diffie-Hellman authentication. With these options, `tulip` and `lilac`can mount the file system read-only even if they are not using DH authentication, if the host names are listed in the `eng` netgroup.

```
# share -F nfs -o sec=dh,rw=tulip:lilac,sec=sys,ro=eng /usr/src
```

Even though UNIX authentication is the default security mode, it is not included if the `-sec` option is used, so it is important to include a `-sec=sys` option if UNIX authentication is to be used with any other authentication mechanism.

You can use a DNS domain name in the access list by preceding the actual domain name with a dot. The dot indicates that the string following it is a domain name, not a fully qualified host name. The following entry allows mount access to all hosts in the `eng.sun.com` domain:

```
# share -F nfs -o ro=.:.eng.sun.com /export/share/man
```

In this example, the single "`.`" matches all hosts that are matched through the NIS or NIS+ name spaces. The results returned from these name services do not include the domain name. The "`.eng.sun.com`" entry matches all hosts that use DNS for name space resolution. DNS always returns a fully qualified host name, so the longer entry is required if you use a combination of DNS and the other name spaces.

You can use a subnet number in an access list by preceding the actual network number or the network name with "`@`". This differentiates the network name from a netgroup or a fully qualified host name. You must identify the subnet in either `/etc/networks` or in a NIS or NIS+ name space. The following entries have the same effect if the `129.144` subnet has been identified as the `eng` network:

```
# share -F nfs -o ro=@eng /export/share/man
# share -F nfs -o ro=@129.144 /export/share/man
# share -F nfs -o ro=@129.144.0.0 /export/share/man
```

The last two entries show you do not need to include the full network address.

If the network prefix is not byte aligned, as with Classless Inter-Domain Routing (CIDR), the mask length can be explicitly specified on the command line. The mask length is defined by following either the network name or the network number with a slash and the number of significant bits in the prefix of the address. For example:

```
# share -f nfs -o ro=@eng/17 /export/share/man
# share -F nfs -o ro=@129.144.132/17 /export/share/man
```

In these examples, the "/17" indicates that the first 17 bits in the address are to be used as the mask. For additional information on CIDR, look up RFC 1519.

You can also select negative access by placing a "-" before the entry. Because the entries are read from left to right, you must place the negative access entries before the entry they apply to:

```
# share -F nfs -o ro=-rose:.eng.sun.com /export/share/man
```

This example would allow access to any hosts in the eng.sun.com domain except the host named rose.

unshare

This command allows you to make a previously available file system unavailable for mounting by clients. You can use the unshare command to unshare any file system—whether the file system was shared explicitly with the share command or automatically through /etc/dfs/dfstab. If you use the unshare command to unshare a file system that you shared through the dfstab file, remember that it will be shared again when you exit and reenter run level 3. You must remove the entry for this file system from the dfstab file if the change is to continue.

When you unshare an NFS file system, access from clients with existing mounts is inhibited. The file system might still be mounted on the client, but the files will not be accessible.

Using the unshare Command

This command unshares a specific file system:

```
# unshare /usr/src
```

shareall

This command allows for multiple file systems to be shared. When used with no options, the command shares all entries in /etc/dfs/dfstab. You can include a file name to specify the name of a file that lists share command lines. If you do not include a file name, /etc/dfs/dfstab is checked. If you use a "-" to replace the file name, you can type share commands from standard input.

Using the shareall Command

This command shares all file systems listed in a local file:

```
# shareall /etc/dfs/special_dfstab
```

unshareall

This command makes all currently shared resources unavailable. The `-F` *FSType* option selects a list of file-system types defined in `/etc/dfs/fstypes`. This flag enables you to choose only certain types of file systems to be unshared. The default file system type is defined in `/etc/dfs/fstypes`. To choose specific file systems, use the `unshare` command.

Using the unshareall Command

This example should unshare all NFS type file systems:

```
# unshareall -F nfs
```

showmount

This command displays all clients that have remotely mounted file systems that are shared from an NFS server, or only the file systems that are mounted by clients, or the shared file systems with the client access information. The command syntax is:

`showmount [ -ade ] [` *hostname* `]`

where `-a` prints a list of all the remote mounts (each entry includes the client name and the directory), `-d` prints a list of the directories that are remotely mounted by clients, `-e` prints a list of the files shared (or exported), and *hostname* selects the NFS server to gather the information from. If *hostname* is not specified the local host is queried.

Using the showmount Command

This command lists all clients and the local directories that they have mounted.

```
# showmount -a bee
lilac:/export/share/man
lilac:/usr/src
rose:/usr/src
tulip:/export/share/man
```

This command lists the directories that have been mounted.

```
# showmount -d bee
/export/share/man
/usr/src
```

This command lists file systems that have been shared.

```
# showmount -e bee
/usr/src                                (everyone)
/export/share/man                    eng
```

setmnt

This command creates an `/etc/mnttab` table. The `mount` and `umount` commands consult the table. Generally, there is no reason to run this command manually; it runs automatically when a system is booted.

Other Useful Commands

These commands can be useful when troubleshooting NFS problems.

nfsstat

You can use this command to gather statistical information about NFS and RPC connections. The syntax of the command is:

```
nfsstat [ -cmnrsz ]
```

where `-c` displays client-side information, `-m` displays statistics for each NFS mounted file system, `-n` specifies that NFS information is to be displayed (both client and server side), `-r` displays RPC statistics, `-s` displays the server-side information, and `-z` specifies that the statistics should be set to zero. If no options are supplied on the command line, the `-cnrs` options are used.

Gathering server-side statistics can be important for debugging problems when new software or hardware is added to the computing environment. Running this command at least once a week, and storing the numbers, provides a good history of previous performance.

Using the `nfsstat` Command

```
# nfsstat -s

Server rpc:
Connection oriented:
calls      badcalls   nullrecv   badlen      xdrcall    dupchecks  dupreqs
11420263   0          0          0           0          1428274    19
Connectionless:
calls      badcalls   nullrecv   badlen      xdrcall    dupchecks  dupreqs
14569706   0          0          0           0          953332     1601

Server nfs:
calls      badcalls
24234967   226
Version 2: (13073528 calls)
null       getattr    setattr    root        lookup     readlink   read
138612 1%  1192059 9% 45676 0%   0 0%        9300029 71% 9872 0%   1319897 10%
wrcache    write      create     remove      rename     link       symlink
0 0%       805444 6%  43417 0%   44951 0%    3831 0%    4758 0%    1490 0%
mkdir      rmdir      readdir    statfs
2235 0%    1518 0%    51897 0%   107842 0%
Version 3: (11114810 calls)
null       getattr    setattr    lookup      access     readlink   read
141059 1%  3911728 35% 181185 1% 3395029 30% 1097018 9% 4777 0%   960503 8%
write      create     mkdir      symlink     mknod      remove     rmdir
763996 6%  159257 1%  3997 0%    10532 0%    26 0%      164698 1%  2251 0%
rename     link       readdir    readdirplus fsstat     fsinfo     pathconf
53303 0%   9500 0%    62022 0%   79512 0%    3442 0%    34275 0%   3023 0%
commit
73677 0%

Server nfs_acl:
Version 2: (1579 calls)
null       getacl     setacl     getattr     access
0 0%       3 0%       0 0%       1000 63%    576 36%
Version 3: (45318 calls)
null       getacl     setacl
0 0%       45318 100% 0 0%
```

This is an example of NFS server statistics. The first five lines deal with RPC and the remaining lines report NFS activities. In both sets of statistics, knowing the average number of `badcalls` or `calls` and the number of calls per week, can help identify when something is going wrong. The `badcalls` value reports the number of bad messages from a client and can point out network hardware problems.

Some of the connections generate write activity on the disks. A sudden increase in these statistics could indicate trouble and should be investigated. For NFS version 2 statistics, the connections to note are: `setattr`, `write`, `create`, `remove`, `rename`, `link`, `symlink`, `mkdir`, and `rmdir`. For NFS version 3 statistics, the value to watch is `commit`. If the `commit` level is high in one NFS server as compared to another almost

identical one, check that the NFS clients have enough memory. The number of `commit` operations on the server grow when clients do not have available resources.

pstack

This command displays a stack trace for each process. It must be run by `root`. You can use it to determine where a process is hung. The only option allowed with this command is the `PID` of the process that you want to check (see the `proc`(1) man page).

The following example is checking the `nfsd` process that is running.

```
# /usr/proc/bin/pstack 243
243:    /usr/lib/nfs/nfsd -a 16
 ef675c04 poll     (24d50, 2, ffffffff)
 000115dc ???????? (24000, 132c4, 276d8, 1329c, 276d8, 0)
 00011390 main     (3, effff14, 0, 0, ffffffff, 400) + 3c8
 00010fb0 _start   (0, 0, 0, 0, 0, 0) + 5c
```

It shows that the process is waiting for a new connection request. This is a normal response. If the stack shows that the process is still in poll after a request is made, the process might be hung. Follow the instructions in "How to Restart NFS Services" on page 180 to fix this problem. Review the instructions in "NFS Troubleshooting Procedures" on page 177 to fully verify that your problem is a hung program.

rpcinfo

This command generates information about the RPC service running on a system. You can also use it to change the RPC service. Many options are available with this command (see the `rpcinfo`(1M) man page). This is a shortened synopsis for some of the options that you can use with the command:

`rpcinfo [ -m | -s ] [` *hostname* `]`

`rpcinfo -T` *transport hostname* `[` *progname* `]`

`rpcinfo [ -t | -u ] [` *hostname* `] [` *progname* `]`

where `-m` displays a table of statistics of the `rpcbind` operations, `-s` displays a concise list of all registered RPC programs, `-T` displays information about services using specific transports or protocols, `-t` displays the RPC programs that use TCP, `-u` displays the RPC programs that use UDP, *transport* selects the transport or protocol for the services,*hostname* selects the host name of the server you need information from, and *progname* selects the RPC program to gather information about. If no value is given for *hostname*, the local host name is used. You can substitute the RPC program number for *progname*, but many users will remember the name and not the number.

You can use the `-p` option in place of the `-s` option on those systems that do not run the NFS version 3 software.

The data generated by this command can include:

- The RPC program number
- The version number for a specific program
- The transport protocol that is being used
- The name of the RPC service
- The owner of the RPC service

Using the `rpcinfo` Command

This example gathers information on the RPC services running on a server. The text generated by the command is filtered by the sort command to make it more readable. Several lines listing RPC services have been deleted from the example.

```
% rpcinfo -s bee |sort -n
   program version(s) netid(s)                          service     owner
    100000  2,3,4     udp,tcp,ticlts,ticotsord,ticots  portmapper  superuser
    100001  4,3,2     ticlts,udp                       rstatd      superuser
    100002  3,2       ticots,ticotsord,tcp,ticlts,udp  rusersd     superuser
    100003  3,2       tcp,udp                          nfs         superuser
    100005  3,2,1     ticots,ticotsord,tcp,ticlts,udp  mountd      superuser
    100008  1         ticlts,udp                       walld       superuser
    100011  1         ticlts,udp                       rquotad     superuser
    100012  1         ticlts,udp                       sprayd      superuser
    100021  4,3,2,1   ticots,ticotsord,ticlts,tcp,udp  nlockmgr    superuser
    100024  1         ticots,ticotsord,ticlts,tcp,udp  status      superuser
    100026  1         ticots,ticotsord,ticlts,tcp,udp  bootparam   superuser
    100029  2,1       ticots,ticotsord,ticlts          keyserv     superuser
    100068  4,3,2     tcp,udp                          cmsd        superuser
    100078  4         ticots,ticotsord,ticlts          kerbd       superuser
    100083  1         tcp,udp                          -           superuser
    100087  11        udp                              adm_agent   superuser
    100088  1         udp,tcp                          -           superuser
    100089  1         tcp                              -           superuser
    100099  1         ticots,ticotsord,ticlts          pld         superuser
    100101  10        tcp,udp                          event       superuser
    100104  10        udp                              sync        superuser
    100105  10        udp                              diskinfo    superuser
    100107  10        udp                              hostperf    superuser
    100109  10        udp                              activity    superuser
    .
    .
    100227  3,2       tcp,udp                          -           superuser
    100301  1         ticlts                           niscachemgr superuser
    390100  3         udp                              -           superuser
1342177279  1,2       tcp                              -           14072
```

This example shows how to gather information about a particular RPC service using a particular transport on a server.

```
% rpcinfo -t bee mountd
program 100005 version 1 ready and waiting
program 100005 version 2 ready and waiting
program 100005 version 3 ready and waiting
% rpcinfo -u bee nfs
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
```

The first example checks the `mountd` service running over TCP. The second example checks the NFS service running over UDP.

snoop

This command is often used to watch for packets on the network. It must be run as `root`. It is a good way to make sure that the network hardware is functioning on both the client and the server. Many options are available (see the `snoop`(1M) man page). A shortened synopsis of the command follows:

`snoop [ -d` *device* `] [ -o` *filename* `] [ host` *hostname* `]`

where `-d` *device* specifies the local network interface, `-o` *filename* stores all the captured packets into the named file, and *hostname* indicates to display only packets going to and from a specific host.

The `-d` *device* option is useful on those servers that have multiple network interfaces. You can use many other expressions besides setting the host. A combination of command expressions with `grep` can often generate data that is specific enough to be useful.

When troubleshooting, make sure that packets are going to and from the proper host. Also, look for error messages. Saving the packets to a file can make it much easier to review the data.

truss

You can use this command to see if a process is hung. It must be run by the owner of the process or by `root`. You can use many options with this command (see the `truss`(1) man page). A shortened syntax of the command is:

`truss [ -t` *syscall* `] -p` *pid*

where `-t` *syscall* selects system calls to trace, and `-p` *pid* indicates the PID of the process to be traced. The *syscall* may be a comma-separated list of system calls to be traced. Also, starting *syscall* with a `!` selects to exclude the listed system calls from the trace.

This example shows that the process is waiting for another connection request from a new client.

```
# /usr/bin/truss -p 243
poll(0x00024D50, 2, -1)         (sleeping...)
```

This is a normal response. If the response does not change after a new connection request has been made, the process could be hung. Follow the instructions in "How to Restart NFS Services" on page 180 to fix the hung program. Review the instructions in "NFS Troubleshooting Procedures" on page 177 to fully verify that your problem is a hung program.

How It All Works Together

The following sections describe some of the complex functions of the NFS software.

Version 2 and Version 3 Negotiation

Because NFS servers might be supporting clients that are not using the NFS version 3 software, part of the initiation procedure includes negotiation of the protocol level. If both the client and the server can support version 3, that version will be used. If either the client or the server can only support version 2, that version will be selected.

You can override the values determined by the negotiation by using the `-vers` option to the `mount` command (see the `mount_nfs`(1M) man page). Under most circumstances, you should not have to specify the version level, as the best one is selected by default.

UDP and TCP Negotiation

During initiation, the transport protocol is also negotiated. By default, the first connection-oriented transport supported on both the client and the server is selected. If this does not succeed, the first available connectionless transport protocol is used. The transport protocols supported on a system are listed in `/etc/netconfig`. TCP is

the connection-oriented transport protocol supported by the release. UDP is the connectionless transport protocol.

When both the NFS protocol version and the transport protocol are determined by negotiation, the NFS protocol version is given precedence over the transport protocol. The NFS version 3 protocol using UDP is given higher precedence than the NFS version 2 protocol using TCP. You can manually select both the NFS protocol version and the transport protocol with the `mount` command (see the `mount_nfs`(1M) man page). Under most conditions, allow the negotiation to select the best options.

File Transfer Size Negotiation

The file transfer size establishes the size of the buffers that are used when transferring data between the client and the server. In general, larger transfer sizes are better. The NFS version 3 protocol has an unlimited transfer size, but starting with the Solaris 2.6 release, the software bids a default buffer size of 32 Kbytes. The client can bid a smaller transfer size at mount time if needed, but under most conditions this is not necessary.

The transfer size is not negotiated with systems using the NFS version 2 protocol. Under this condition the maximum transfer size is set to 8 Kbytes.

You can use the `-rsize` and `-wsize` options to set the transfer size manually with the `mount` command. You might need to reduce the transfer size for some PC clients. Also, you can increase the transfer size if the NFS server is configured to use larger transfer sizes.

How File Systems Are Mounted

When a client needs to mount a file system from a server, it must obtain a file handle from the server that corresponds to the file system. This process requires that several transactions occur between the client and the server. In this example, the client is attempting to mount `/home/terry` from the server. A `snoop` trace for this transaction follows.

```
client -> server PORTMAP C GETPORT prog=100005 (MOUNT) vers=3 proto=UDP
server -> client PORTMAP R GETPORT port=33492
client -> server MOUNT3 C Null
server -> client MOUNT3 R Null
client -> server MOUNT3 C Mount /export/home9/terry
server -> client MOUNT3 R Mount OK FH=9000 Auth=unix
client -> server PORTMAP C GETPORT prog=100003 (NFS) vers=3 proto=TCP
server -> client PORTMAP R GETPORT port=2049
client -> server NFS C NULL3
server -> client NFS R NULL3
client -> server NFS C FSINFO3 FH=9000
```

```
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

In this trace, the client first requests the mount port number from the portmap service on the NFS server. After the client received the mount port number (`33492`), that number is used to ping the service on the server. After the client has determined that a service is running on that port number, the client then makes a mount request. When the server responds to this request, it includes the file handle for the file system (`9000`) that is being mounted. The client then sends a request for the NFS port number. When the client receives the number from the server, it pings the NFS service (`nfsd`), and requests NFS information about the file system using the file handle.

In the following trace, the client is mounting the file system with the `-public` option.

```
client -> server NFS C LOOKUP3 FH=0000 /export/home9/terry
server -> client NFS R LOOKUP3 OK FH=9000
client -> server NFS C FSINFO3 FH=9000
server -> client NFS R FSINFO3 OK
client -> server NFS C GETATTR3 FH=9000
server -> client NFS R GETATTR3 OK
```

By using the default public file handle (which is `0000`), all of the transactions to get information from the portmap service and to determine the NFS port number are skipped.

Effects of the `-public` Option and NFS URLs When Mounting

Using the `-public` option can create conditions that cause a mount to fail. Adding an NFS URL can also confuse the situation. The following list describes the specifics of how a file system is mounted when using these options.

Public option with NFS URL – Forces the use of the public file handle. The mount fails if the public file handle is not supported.

Public option with regular path – Forces the use of the public file handle. The mount fails if the public file handle is not supported.

NFS URL only – Use the public file handle if enabled on the NFS server. If the mount fails using the public file handle, then try the mount using the MOUNT protocol.

Regular path only – Do not use the public file handle. The MOUNT protocol is used.

Client-Side Failover

Using client-side failover, an NFS client can switch to another server if the server supporting a replicated file system becomes unavailable. The file system can become unavailable if the server it is connected to crashes, if the server is overloaded, or if there is a network fault. The failover, under these conditions, is normally transparent to the user. After it is established, the failover can occur at any time without disrupting the processes running on the client.

Failover requires that the file system be mounted read-only. The file systems must be identical for the failover to occur successfully. See "What Is a Replicated File System?" on page 216 for a description of what makes a file system identical. A static file system or one that is not changed often is the best candidate for failover.

You cannot use file systems that are mounted using CacheFS with failover. Extra information is stored for each CacheFS file system. This information cannot be updated during failover, so only one of these two features can be used when mounting a file system.

The number of replicas that need to be established for each file system depends on many factors. In general, it is better to have a couple of servers, each supporting multiple subnets rather than have a unique server on each subnet. The process requires checking of each server in the list, so the more servers that are listed, the slower each mount will be.

Failover Terminology

To fully comprehend the process, two terms need to be understood.

- *failover* – Selecting a server from a list of servers supporting a replicated file system. Normally, the next server in the sorted list is used, unless it fails to respond.
- *remap* – Making use of a new server. Through normal use, the clients store the path name for each active file on the remote file system. During the remap, these path names are evaluated to locate the files on the new server.

What Is a Replicated File System?

For the purposes of failover, a file system can be called a *replica* when each file is the same size and has the same vnode type as the original file system. Permissions, creation dates, and other file attributes are not considered. If the file size or vnode types are different, the remap fails and the process hangs until the old server becomes available.

You can maintain a replicated file system using `rdist`, `cpio`, or another file transfer mechanism. Because updating the replicated file systems causes inconsistency, follow these suggestions for best results:

- Rename the old version of the file before installing a new one.
- Run the updates at night when client usage is low.
- Keep the updates small.
- Minimize the number of copies.

Failover and NFS Locking

Some software packages require read locks on files. To prevent these products from breaking, read locks on read-only file systems are allowed, but are visible to the client side only. The locks persist through a remap because the server doesn't "know" about them. Because the files should not be changing, you do not need to lock the file on the server side.

Large Files

Starting with 2.6, the Solaris release supports files that are over 2 Gbytes. By default, UFS file systems are mounted with the `-largefiles` option to support the new functionality. Previous releases cannot handle files of this size. See "How to Disable Large Files on an NFS Server" on page 152 for instructions.

No changes need to occur on a Solaris 2.6 NFS client to be able to access a large file, if the file system on the server is mounted with the `-largefiles` option. However, not all 2.6 commands can handle these large files. See `largefile(5)` for a list of the commands that can handle the large files. Clients that cannot support the NFS version 3 protocol with the large file extensions cannot access any large files. Although clients running the Solaris 2.5 release can use the NFS version 3 protocol, large file support was not included in that release.

How NFS Server Logging Works

NFS server logging provides records of NFS reads and writes, as well as operations that modify the file system. This data can be used to track access to information. In addition, the records can provide a quantitative way to measure interest in the information.

When a file system with logging enabled is accessed, the kernel writes raw data into a buffer file. This data includes a timestamp, the client IP address, the UID of the requester, the file handle of the file or directory object that is being accessed, and the type of operation that occurred.

The `nfslogd` daemon converts this raw data into ASCII records that are stored in log files. During the conversion the IP addresses are modified to host names and the UIDs are modified to logins if the name service that is enabled can find matches. The file handles are also converted into path names. To accomplish this, the daemon keeps track of the file handles and stores information in a separate file handle to path table, so that the path does not have to be re-identified each time a file handle is accessed. Because there is no tracking of changes to the mappings in the file handle to path table if `nfslogd` is turned off, it is important to keep the daemon running.

How the WebNFS Service Works

The WebNFS service makes files in a directory available to clients using a public file handle. A file handle is an address generated by the kernel that identifies a file for NFS clients. The *public file handle* has a predefined value, so the server does not need to generate a file handle for the client. The ability to use this predefined file handle reduces network traffic by eliminating the `MOUNT` protocol and should make things run faster for the clients.

By default the public file handle on an NFS server is established on the root file system. This default provides WebNFS access to any clients that already have mount privileges on the server. You can change the public file handle to point to any file system by using the `share` command.

When the client has the file handle for the file system, a `LOOKUP` is run to determine the file handle for the file to be accessed. The NFS protocol allows the evaluation of only one path name component at a time. Each additional level of directory hierarchy requires another `LOOKUP`. A WebNFS server can evaluate an entire path name with a single transaction, called multicomponent lookup, when the `LOOKUP` is relative to the public file handle. With multicomponent lookup, the WebNFS server is able to deliver the file handle to the desired file without having to exchange the file handles for each directory level in the path name.

In addition, an NFS client can initiate concurrent downloads over a single TCP connection, which provides quick access without the additional load on the server caused by setting up multiple connections. Although Web browser applications support concurrent downloading of multiple files, each file has its own connection. By using one connection, the WebNFS software reduces the overhead on the server.

If the final component in the path name is a symbolic link to another file system, the client can access the file if the client already has access through normal NFS activities.

Normally, an NFS URL is evaluated relative to the public file handle. The evaluation can be changed to be relative to the server's root file system by adding an additional slash to the beginning of the path. In this example, these two NFS URLs are equivalent if the public file handle has been established on the `/export/ftp` file system.

```
nfs://server/junk
nfs://server//export/ftp/junk
```

How WebNFS Security Negotiation Works

The Solaris 8 release includes a new protocol so a WebNFS client can negotiate a selected security mechanism with a WebNFS server. The new protocol uses security negotiation multicomponent lookup, which is an extension to the multicomponent lookup used in earlier versions of the WebNFS protocol.

The WebNFS client initiates the process by making a regular multicomponent lookup request using the public file handle. Because the client has no knowledge of how the path is protected by the server, the default security mechanism is used. If the default security mechanism is not sufficient, the server replies with an `AUTH_TOOWEAK` error, indicating that the default mechanism is not valid and the client needs to use a stronger one.

When the client receives the `AUTH_TOOWEAK` error, it sends a request to the server to determine which security mechanisms are required. If the request succeeds, the server responds with an array of security mechanisms required for the specified path. Depending on the size of the array of security mechanisms, the client might have to make more requests to get the complete array. If the server does not support WebNFS security negotiation, the request fails.

After a successful request, the WebNFS client selects the first security mechanism from the array that it supports and issues a regular multicomponent lookup request using the selected security mechanism to acquire the file handle. All subsequent NFS requests are made using the selected security mechanism and the file handle.

WebNFS Limitations With Web Browser Use

Several functions that a Web site using HTTP can provide are not supported by the WebNFS software. These differences stem from the fact that the NFS server only sends the file, so any special processing must be done on the client. If you need to have one web site configured for both WebNFS and HTTP access, consider the following issues:

- NFS browsing does not run CGI scripts, so a file system with an active web site that uses many CGI scripts might not be appropriate for NFS browsing.
- The browser might start different viewers, to handle files in different file formats. Accessing these files through an NFS URL starts an external viewer as long as the file type can be determined by the file name. The browser should recognize any file name extension for a standard MIME type when an NFS URL is used. Because the WebNFS software does not check inside the file to determine the file type—unlike some Web browser applications—the only way to determine a file type is by the file name extension.

- NFS browsing cannot utilize server-side image maps (clickable images). However, it can utilize client-side image maps (clickable images) because the URLs are defined with the location. No additional response is required from the document server.

Secure NFS System

The NFS environment is a powerful and convenient way to share file systems on a network of different computer architectures and operating systems. However, the same features that make sharing file systems through NFS operation convenient also pose some security problems. Historically, most NFS implementations have used UNIX (or AUTH_SYS) authentication, but stronger authentication methods such as AUTH_DH have also been available. When using UNIX authentication, an NFS server authenticates a file request by authenticating the computer making the request, but not the user, so a client user can run `su` and impersonate the owner of a file. If DH authentication is used, the NFS server authenticates the user, making this sort of impersonation much harder.

With root access and knowledge of network programming, anyone can introduce arbitrary data into the network and extract any data from the network. The most dangerous attacks are those involving the introduction of data, such as impersonating a user by generating the right packets or recording "conversations" and replaying them later. These attacks affect data integrity. Attacks involving passive eavesdropping—merely listening to network traffic without impersonating anybody—are not as dangerous, as data integrity is not compromised. Users can protect the privacy of sensitive information by encrypting data that goes over the network.

A common approach to network security problems is to leave the solution to each application. A better approach is to implement a standard authentication system at a level that covers all applications.

The Solaris operating environment includes an authentication system at the level of remote procedure call (RPC)—the mechanism on which NFS operation is built. This system, known as Secure RPC, greatly improves the security of network environments and provides additional security to services such as the NFS system. When the NFS system uses the facilities provided by Secure RPC, it is known as a Secure NFS system.

Secure RPC

Secure RPC is fundamental to the Secure NFS system. The goal of Secure RPC is to build a system at least as secure as a time-sharing system (one in which all users share a single computer). A time-sharing system authenticates a user through a login password. With data encryption standard (DES) authentication, the same is true. Users

can log in on any remote computer just as they can on a local terminal, and their login passwords are their passports to network security. In a time-sharing environment, the system administrator has an ethical obligation not to change a password to impersonate someone. In Secure RPC, the network administrator is trusted not to alter entries in a database that stores *public keys*.

You need to be familiar with two terms to understand an RPC authentication system: credentials and verifiers. Using ID badges as an example, the credential is what identifies a person: a name, address, birthday, and so on. The verifier is the photo attached to the badge: you can be sure the badge has not been stolen by checking the photo on the badge against the person carrying it. In RPC, the client process sends both a credential and a verifier to the server with each RPC request. The server sends back only a verifier because the client already "knows" the server's credentials.

RPC's authentication is open ended, which means that a variety of authentication systems can be plugged into it. Currently, there are several systems: UNIX, DH and KERB.

When UNIX authentication is used by a network service, the credentials contain the client's host name, UID, GID, and group-access list, but the verifier contains nothing. Because there is no verifier, a superuser could falsify appropriate credentials, using commands such as `su`. Another problem with UNIX authentication is that it assumes all computers on a network are UNIX computers. UNIX authentication breaks down when applied to other operating systems in a heterogeneous network.

To overcome the problems of UNIX authentication, Secure RPC uses DH authentication.

DH Authentication

DH authentication uses the data encryption standard (DES) and Diffie-Hellman public-key cryptography to authenticate both users and computers in the network. DES is a standard encryption mechanism; Diffie-Hellman public-key cryptography is a cipher system that involves two keys: one public and one secret. The public and secret keys are stored in the name space. NIS stores the keys in the publickey map. These maps contain the public key and secret key for all potential users. See the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)* for more information on how to set up the maps.

The security of DH authentication is based on a sender's ability to encrypt the current time, which the receiver can then decrypt and check against its own clock. The timestamp is encrypted with DES. The requirements for this scheme to work are:

- The two agents must agree on the current time.
- The sender and receiver must be using the same encryption key.

If a network runs a time-synchronization program, the time on the client and the server is synchronized automatically. If a time-synchronization program is not available, timestamps can be computed using the server's time instead of the network time. The client asks the server for the time before starting the RPC session, then computes the time difference between its own clock and the server's. This difference is used to offset the client's clock when computing timestamps. If the client and server clocks get out of synchronization to the point where the server begins to reject the client's requests, the DH authentication system on the client resynchronizes with the server.

The client and server arrive at the same encryption key by generating a random *conversation key*, also known as the *session key*, and by using public-key cryptography to deduce a *common key*. The common key is a key that only the client and server are capable of deducing. The conversation key is used to encrypt and decrypt the client's timestamp; the common key is used to encrypt and decrypt the conversation key.

KERB Authentication

Kerberos is an authentication system developed at MIT. Encryption in Kerberos is based on DES. Kerberos support is no longer supplied as part of Secure RPC, but a server and client-side implementation is included with the Solaris 9 release. See "Introduction to SEAM" in *System Administration Guide: Security Services* for more information about the Solaris 9 implementation of Kerberos Authentication.

Using Secure RPC With NFS

Be aware of the following points if you plan to use Secure RPC:

- If a server crashes when no one is around (after a power failure for example), all the secret keys that are stored on the system are deleted. Now no process can access secure network services or mount an NFS file system. The important processes during a reboot are usually run as root, so these processes would work if root's secret key were stored away, but nobody is available to type the password that decrypts it. `keylogin -r` allows root to store the clear secret key in `/etc/.rootkey`, which `keyserv` reads.
- Some systems boot in single-user mode, with a root login shell on the console and no password prompt. Physical security is imperative in such cases.
- Diskless computer booting is not totally secure. Somebody could impersonate the boot server and boot a devious kernel that, for example, makes a record of your secret key on a remote computer. The Secure NFS system provides protection only after the kernel and the key server are running. Before that, there is no way to authenticate the replies given by the boot server. This could be a serious problem, but it requires a sophisticated attack, using kernel source code. Also, the crime would leave evidence. If you polled the network for boot servers, you would

discover the devious boot server's location.

- Most setuid programs are owned by `root`; if the secret key for `root` is stored in `/etc/.rootkey`, these programs behave as they always have. If a setuid program is owned by a user, however, it might not always work. For example, if a setuid program is owned by dave and dave has not logged into the computer since it booted, the program would not be able to access secure network services.
- If you log in to a remote computer (using `login`, `rlogin`, or `telnet`) and use `keylogin` to gain access, you give access to your account. This is because your secret key gets passed to that computer's key server, which then stores it. This is only a concern if you do not trust the remote computer. If you have doubts, however, do not log in to a remote computer if it requires a password. Instead, use the NFS environment to mount file systems shared by the remote computer. As an alternative, you can use `keylogout` to delete the secret key from the key server.
- If a home directory is shared with the `-o sec=dh` option, remote logins can be a problem. If the `/etc/hosts.equiv` or `~/.rhosts` files are not set to prompt for a password, the login will succeed, but the users cannot access their home directories because no authentication has occurred locally. If the user is prompted for a password, the user will have access to his or her home directory as long as the password matches the network password.

Autofs Maps

Autofs uses three types of maps:

- Master map
- Direct maps
- Indirect maps

Master Autofs Map

The `auto_master` map associates a directory with a map. It is a master list specifying all the maps that autofs should check. The following example shows what an `auto_master` file could contain.

EXAMPLE 15–1 Sample `/etc/auto_master` File

```
# Master map for automounter
#
+auto_master
/net            -hosts          -nosuid,nobrowse
/home           auto_home       -nobrowse
```

EXAMPLE 15–1 Sample `/etc/auto_master` File *(Continued)*

```
/xfn            -xfn
/-              auto_direct     -ro
```

This example shows the generic `auto_master` file with one addition for the `auto_direct` map. Each line in the master map `/etc/auto_master` has the following syntax:

mount-point map-name [*mount-options*]

mount-point	*mount-point* is the full (absolute) path name of a directory. If the directory does not exist, autofs creates it if possible. If the directory exists and is not empty, mounting on it hides its contents. In this case, autofs issues a warning. The notation `/-` as a mount point indicates that the map in question is a direct map, and no particular mount point is associated with the map as a whole.
map-name	*map-name* is the map autofs uses to find directions to locations, or mount information. If the name is preceded by a slash (`/`), autofs interprets the name as a local file. Otherwise, autofs searches for the mount information using the search specified in the name service switch configuration file (`/etc/nsswitch.conf`). Special maps are also used for `/net` and `/xfn` (see "Mount Point `/net`" on page 225 and "Mount Point `/xfn`" on page 225).
mount-options	*mount-options* is an optional, comma-separated list of options that apply to the mounting of the entries specified in map-name, unless the entries in map-name list other options. Options for each specific type of file system are listed in the mount man page for that file system (for example, see the `mount_nfs`(1M) man page for NFS specific mount options). For NFS specific mount points, the `bg` (background) and `fg` (foreground) options do not apply.

A line beginning with # is a comment. Everything that follows until the end of the line is ignored.

To split long lines into shorter ones, put a backslash (\) at the end of the line. The maximum number of characters of an entry is 1024.

Note – If the same mount point is used in two entries, the first entry is used by the `automount` command. The second entry is ignored.

Mount Point `/home`

The mount point `/home` is the directory under which the entries listed in `/etc/auto_home` (an indirect map) are to be mounted.

Note – Autofs runs on all computers and supports `/net` and `/home` (automounted home directories) by default. These defaults can be overridden by entries in the NIS `auto.master` map or NIS+ `auto_master` table, or by local editing of the `/etc/auto_master` file.

Mount Point `/net`

Autofs mounts under the directory `/net` all the entries in the special map `-hosts`. This is a built-in map that uses only the hosts database. For example, if the computer `gumbo` is in the hosts database and it exports any of its file systems, the command:

```
%cd /net/gumbo
```

changes the current directory to the root directory of the computer `gumbo`. Autofs can mount only the *exported* file systems of host `gumbo`, that is, those on a server available to network users as opposed to those on a local disk. Therefore, all the files and directories on `gumbo` might not be available through `/net/gumbo`.

With the `/net` method of access, the server name is in the path and is location dependent. If you want to move an exported file system from one server to another, the path might no longer work. Instead, you should set up an entry in a map specifically for the file system you want rather than use `/net`.

Note – Autofs checks the server's export list only at mount time. After a server's file systems are mounted, autofs does not check with the server again until the server's file systems are automatically unmounted. Therefore, newly exported file systems are not "seen" until the file systems on the client are unmounted and then remounted.

Mount Point `/xfn`

This mount point provides the autofs directory structure for the resources that are shared through the FNS name space (see the *System Administration Guide: Naming and Directory Services (FNS and NIS+)* for more information about FNS).

Direct Autofs Maps

A direct map is an automount point. With a direct map, there is a direct association between a mount point on the client and a directory on the server. Direct maps have a full path name and indicate the relationship explicitly. This is a typical `/etc/auto_direct` map:

```
/usr/local          -ro \
   /bin                    ivy:/export/local/sun4 \
   /share                  ivy:/export/local/share \
   /src                    ivy:/export/local/src
   /usr/man         -ro    oak:/usr/man \
                           rose:/usr/man \
                           willow:/usr/man
   /usr/games       -ro    peach:/usr/games
   /usr/spool/news  -ro    pine:/usr/spool/news \
                           willow:/var/spool/news
```

Lines in direct maps have the following syntax:

key [*mount-options*] *location*

key	*key* is the path name of the mount point in a direct map.
mount-options	*mount-options* is the options you want to apply to this particular mount. They are required only if they differ from the map default. Options for each specific type of file system are listed in the mount man page for that file system (for example, see the `mount_cachefs`(1M) man page for CacheFS specific mount options).
location	*location* is the location of the file system, specified (one or more) as *server*:*pathname* for NFS file systems or :*devicename* for High Sierra file systems (HSFS). **Note –** The *pathname* should not include an automounted mount point; it should be the actual absolute path to the file system. For instance, the location of a home directory should be listed as *server*`:/export/home/`*username*, not as *server*`:/home/`*username*.

As in the master map, a line beginning with # is a comment. All the text that follows until the end of the line is ignored. Put a backslash at the end of the line to split long lines into shorter ones.

Of all the maps, the entries in a direct map most closely resemble the corresponding entries in `/etc/vfstab` (`vfstab` contains a list of all file systems to be mounted). An entry that appears in `/etc/vfstab` as:

```
dancer:/usr/local - /usr/local/tmp nfs - yes ro
```

appears in a direct map as:

```
/usr/local/tmp     -ro     dancer:/usr/local
```

Note – No concatenation of options occurs between the automounter maps. Any options added to an automounter map override all options listed in maps that are searched earlier. For instance, options included in the `auto_master` map would be overridden by corresponding entries in any other map.

See "How Autofs Selects the Nearest Read-Only Files for Clients (Multiple Locations)" on page 234 for other important features associated with this type of map.

Mount Point /–

In Example 15–1, the mount point `/-` tells autofs not to associate the entries in `auto_direct` with any specific mount point. Indirect maps use mount points defined in the `auto_master` file. Direct maps use mount points specified in the named map. (Remember, in a direct map the key, or mount point, is a full path name.)

An NIS or NIS+ `auto_master` file can have only one direct map entry because the mount point must be a unique value in the name space. An `auto_master` file that is a local file can have any number of direct map entries, as long as entries are not duplicated.

Indirect Autofs Maps

An indirect map uses a substitution value of a key to establish the association between a mount point on the client and a directory on the server. Indirect maps are useful for accessing specific file systems, like home directories. The `auto_home` map is an example of an indirect map.

Lines in indirect maps have the following general syntax:

key `[` *mount-options* `]` *location*

key	*key* is a simple name (no slashes) in an indirect map.
mount-options	*mount-options* is the options you want to apply to this particular mount. They are required only if they differ from the map default. Options for each specific type of file system are listed in the mount man page for that file system (for example, see the `mount_nfs`(1M) man page for NFS specific mount options).

location	*location* is the location of the file system, specified (one or more) as *server*:*pathname*. **Note –** The *pathname* should not include an automounted mount point; it should be the actual absolute path to the file system. For instance, the location of a directory should be listed as *server*:`/usr/local`, not as *server*:`/net/`*server*`/usr/local`.

As in the master map, a line beginning with # is a comment. All the text that follows until the end of the line is ignored. Put a backslash (\) at the end of the line to split long lines into shorter ones. Example 15–1 shows an `auto_master` map that contains the entry:

```
/home      auto_home        -nobrowse
```

`auto_home` is the name of the indirect map that contains the entries to be mounted under `/home`. A typical `auto_home` map might contain:

```
david                  willow:/export/home/david
rob                    cypress:/export/home/rob
gordon                 poplar:/export/home/gordon
rajan                  pine:/export/home/rajan
tammy                  apple:/export/home/tammy
jim                    ivy:/export/home/jim
linda    -rw,nosuid    peach:/export/home/linda
```

As an example, assume that the previous map is on host `oak`. If user `linda` has an entry in the password database specifying her home directory as `/home/linda`, whenever she logs in to computer `oak`, autofs mounts the directory `/export/home/linda` residing on the computer `peach`. Her home directory is mounted read-write, nosuid.

Assume the following conditions occur: User linda's home directory is listed in the password database as `/home/linda`. Anybody, including Linda, has access to this path from any computer set up with the master map referring to the map in the previous example.

Under these conditions, user `linda` can run login or rlogin on any of these computers and have her home directory mounted in place for her.

Furthermore, now Linda can also type the following command:

```
% cd ~david
```

autofs mounts David's home directory for her (if all permissions allow).

Note – No concatenation of options between the automounter maps. Any options added to an automounter map override all options listed in maps that are searched earlier. For instance, options included in the `auto_master` map are overridden by corresponding entries in any other map.

On a network without a name service, you have to change all the relevant files (such as `/etc/passwd`) on all systems on the network to accomplish this. With NIS, make the changes on the NIS master server and propagate the relevant databases to the slave servers. On a network running NIS+, propagating the relevant databases to the slave servers is done automatically after the changes are made.

How Autofs Works

Autofs is a client-side service that automatically mounts the appropriate file system. When a client attempts to access a file system that is not presently mounted, the autofs file system intercepts the request and calls `automountd` to mount the requested directory. The `automountd` daemon locates the directory, mounts it within autofs, and replies. On receiving the reply, autofs allows the waiting request to proceed. Subsequent references to the mount are redirected by the autofs—no further participation is required by `automountd` until the file system is automatically unmounted by autofs after a period of inactivity.

The components that work together to accomplish automatic mounting are:

- The `automount` command
- The `autofs` file system
- The `automountd` daemon

The `automount` command, called at system startup time, reads the master map file `auto_master` to create the initial set of autofs mounts. These autofs mounts are not automatically mounted at startup time. They are points under which file systems are mounted in the future. These points are also known as trigger nodes.

After the autofs mounts are set up, they can trigger file systems to be mounted under them. For example, when autofs receives a request to access a file system that is not currently mounted, autofs calls `automountd`, which actually mounts the requested file system.

Starting with the Solaris 2.5 release, the `automountd` daemon is completely independent from the automount command. Because of this separation, it is possible to add, delete, or change map information without first having to stop and start the `automountd` daemon process.

After initially mounting autofs mounts, the automount command is used to update autofs mounts as necessary, by comparing the list of mounts in the `auto_master` map with the list of mounted file systems in the mount table file `/etc/mnttab` (formerly `/etc/mtab`) and making the appropriate changes. This allows system administrators to change mount information within `auto_master` and have those changes used by the autofs processes without having to stop and restart the autofs daemon. After the file system is mounted, further access does not require any action from `automountd` until the file system is automatically unmounted.

Unlike `mount`, `automount` does not read the `/etc/vfstab` file (which is specific to each computer) for a list of file systems to mount. The `automount` command is controlled within a domain and on computers through the name space or local files.

This is a simplified overview of how autofs works:

The automount daemon `automountd` starts at boot time from the `/etc/init.d/autofs` script (see Figure 15–1). This script also runs the `automount` command, which reads the master map (see "How Autofs Starts the Navigation Process (Master Map)" on page 231) and installs autofs mount points.

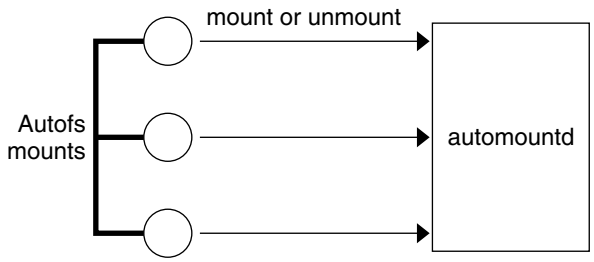


FIGURE 15–1 `/etc/init.d/autofs` Script Starts `automount`

Autofs is a kernel file system that supports automatic mounting and unmounting.

When a request is made to access a file system at an autofs mount point:

1. Autofs intercepts the request.
2. Autofs sends a message to the automountd for the requested file system to be mounted.
3. `automountd` locates the file system information in a map, creates the trigger nodes, and performs the mount.
4. Autofs allows the intercepted request to proceed.
5. Autofs unmounts the file system after a period of inactivity.

Note – Mounts managed through the autofs service should not be manually mounted or unmounted. Even if the operation is successful, the autofs service does not check that the object has been unmounted, resulting in possible inconsistencies. A reboot clears all of the autofs mount points.

How Autofs Navigates Through the Network (Maps)

Autofs searches a series of maps to navigate its way through the network. Maps are files that contain information such as the password entries of all users on a network or the names of all host computers on a network, that is, network-wide equivalents of UNIX administration files. Maps are available locally or through a network name service like NIS or NIS+. You create maps to meet the needs of your environment using the Solstice System Management Tools. See "Modifying How Autofs Navigates the Network (Modifying Maps)" on page 239.

How Autofs Starts the Navigation Process (Master Map)

The `automount` command reads the master map at system startup. Each entry in the master map is a direct or indirect map name, its path, and its mount options, as shown

in Figure 15–2. The specific order of the entries is not important. `automount` compares entries in the master map with entries in the mount table to generate a current list.

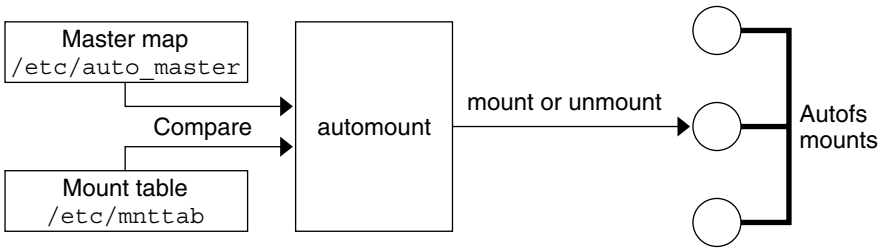


FIGURE 15–2 Navigation Through the Master Map

Autofs Mount Process

What the autofs service does when a mount request is triggered depends on how the automounter maps are configured. The mount process is generally the same for all mounts, but the final result changes with the mount point specified and the complexity of the maps. Starting with the Solaris 2.6 release, the mount process has also been changed to include the creation of the trigger nodes.

Simple Autofs Mount

To help explain the autofs mount process, assume that the following files are installed.

```
$ cat /etc/auto_master
# Master map for automounter
#
+auto_master
/net        -hosts        -nosuid,nobrowse
/home       auto_home     -nobrowse
/xfn        -xfn
/share      auto_share
$ cat /etc/auto_share
# share directory map for automounter
#
ws          gumbo:/export/share/ws
```

When the `/share` directory is accessed, the autofs service creates a trigger node for `/share/ws`, which can be seen in `/etc/mnttab` as an entry that resembles the following entry:

```
-hosts  /share/ws     autofs  nosuid,nobrowse,ignore,nest,dev=###
```

When the /share/ws directory is accessed, the autofs service completes the process with these steps:

1. Pings the server's mount service to see if it's alive.
2. Mounts the requested file system under /share. Now /etc/mnttab file contains the following entries:

```
-hosts  /share/ws     autofs  nosuid,nobrowse,ignore,nest,dev=###
gumbo:/export/share/ws /share/ws   nfs   nosuid,dev=####    #####
```

Hierarchical Mounting

When multiple layers are defined in the automounter files, the mount process becomes more complex. If the /etc/auto_shared file from the previous example is expanded to contain:

```
# share directory map for automounter
#
ws        /       gumbo:/export/share/ws
          /usr    gumbo:/export/share/ws/usr
```

The mount process is basically the same as the previous example when the /share/ws mount point is accessed. In addition, a trigger node to the next level (/usr) is created in the /share/ws file system so that the next level can be mounted if it is accessed. In this example, /export/share/ws/usr must exist on the NFS server for the trigger node to be created.

Caution – Do not use the -soft option when specifying hierarchical layers. Refer to "Autofs Unmounting" on page 233 for an explanation of this limitation.

Autofs Unmounting

The unmounting that occurs after a certain amount of idle time is from the bottom up (reverse order of mounting). If one of the directories at a higher level in the hierarchy is busy, only file systems below that directory are unmounted. During the unmounting process, any trigger nodes are removed and then the file system is unmounted. If the file system is busy, the unmount fails and the trigger nodes are reinstalled.

Caution – Do not use the `-soft` option when specifying hierarchical layers. If the `-soft` option is used, requests to reinstall the trigger nodes can time out. The failure to reinstall the trigger notes leaves no access to the next level of mounts. The only way to clear this problem is to have the automounter unmount all of the components in the hierarchy, either by waiting for the file systems to be automatically unmounted or by rebooting the system.

How Autofs Selects the Nearest Read-Only Files for Clients (Multiple Locations)

In the example of a direct map, which was:

```
/usr/local          -ro \
   /bin                   ivy:/export/local/sun4\
   /share                 ivy:/export/local/share\
   /src                   ivy:/export/local/src
/usr/man            -ro   oak:/usr/man \
                          rose:/usr/man \
                          willow:/usr/man
/usr/games          -ro   peach:/usr/games
/usr/spool/news     -ro   pine:/usr/spool/news \
                          willow:/var/spool/news
```

The mount points `/usr/man` and `/usr/spool/news` list more than one location (three for the first, two for the second). This means any of the replicated locations can provide the same service to any user. This procedure makes sense only when you mount a file system that is read-only, as you must have some control over the locations of files you write or modify. You don't want to modify files on one server on one occasion and, minutes later, modify the "same" file on another server. The benefit is that the best available server is used automatically without any effort required by the user.

If the file systems are configured as replicas (see "What Is a Replicated File System?" on page 216), the clients have the advantage of using failover. Not only is the best server automatically determined, but if that server becomes unavailable, the client automatically uses the next-best server. Failover was first implemented in the Solaris 2.6 release.

An example of a good file system to configure as a replica is man pages. In a large network, more than one server can export the current set of manual pages. Which server you mount them from does not matter, as long as the server is running and exporting its file systems. In the previous example, multiple mount locations are expressed as a list of mount locations in the map entry.

```
/usr/man -ro oak:/usr/man rose:/usr/man willow:/usr/man
```

Here you can mount the man pages from the servers `oak`, `rose`, or `willow`. Which server is best depends on a number of factors including: the number of servers supporting a particular NFS protocol level, the proximity of the server, and weighting.

During the sorting process, a count of the number of servers supporting the NFS version 2 and NFS version 3 protocols is made. Whichever protocol is supported on the most servers becomes the protocol supported by default. This provides the client with the maximum number of servers to depend on.

After the largest subset of servers with the same protocol version is found, that server list is sorted by proximity. Servers on the local subnet are given preference over servers on a remote subnet. The closest server is given preference, which reduces latency and network traffic. Figure 15–3 illustrates server proximity.

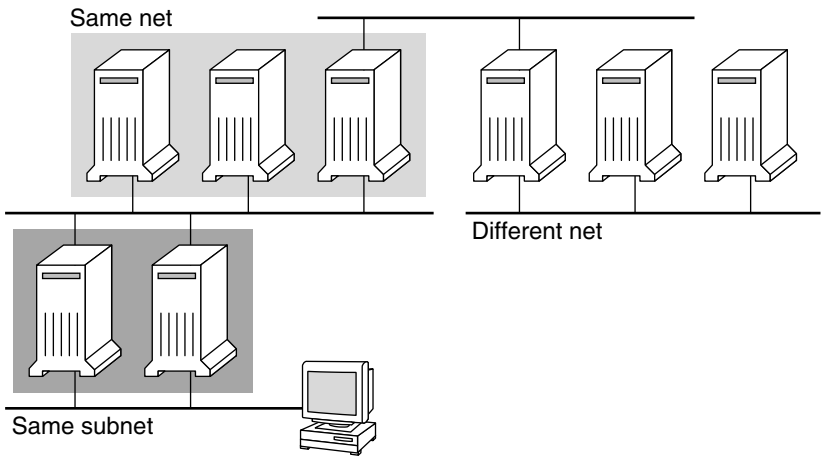


FIGURE 15–3 Server Proximity

If several servers supporting the same protocol are on the local subnet, the time to connect to each server is determined and the fastest is used. The sorting can also be influenced by using weighting (see "Autofs and Weighting" on page 236).

If version 3 servers are more abundant, the sorting process becomes more complex. Normally, servers on the local subnet are given preference over servers on a remote subnet. A version 2 server can complicate matters, as it might be closer than the nearest version 3 server. If there is a version 2 server on the local subnet and the closest version 3 server is on a remote subnet, the version 2 server is given preference. This preference is only checked if there are more version 3 servers than version 2 servers. If there are more version 2 servers, only a version 2 server is selected.

With failover, the sorting is checked once at mount time to select one server from which to mount, and again anytime the mounted server becomes unavailable.

Multiple locations are useful in an environment where individual servers might not export their file systems temporarily.

This feature is particularly useful in a large network with many subnets. Autofs chooses the nearest server and therefore confines NFS network traffic to a local network segment. In servers with multiple network interfaces, list the host name associated with each network interface as if it were a separate server. Autofs selects the nearest interface to the client.

Autofs and Weighting

You can influence the selection of servers at the same proximity level by adding a weighting value to the autofs map. For example:

```
/usr/man -ro oak,rose(1),willow(2):/usr/man
```

The numbers in parentheses indicate a weighting. Servers without a weighting have a value of zero (most likely to be selected). The higher the weighting value, the lower the chance the server will be selected.

Note – All other server selection factors are more important than weighting. Weighting is only considered when selecting between servers with the same network proximity.

Variables in a Map Entry

You can create a client-specific variable by prefixing a dollar sign ($) to its name. This helps you to accommodate different architecture types accessing the same file system location. You can also use curly braces to delimit the name of the variable from appended letters or digits. Table 15–3 shows the predefined map variables.

TABLE 15–3 Predefined Map Variables

Variable	Meaning	Derived From	Example
ARCH	Architecture type	`uname -m`	`sun4u`
CPU	Processor type	`uname -p`	`sparc`
HOST	Host name	`uname -n`	`dinky`
OSNAME	Operating system name	`uname -s`	`SunOS`
OSREL	Operating system release	`uname -r`	`5.8`

TABLE 15–3 Predefined Map Variables *(Continued)*

Variable	Meaning	Derived From	Example
OSVERS	Operating system version (version of the release)	uname -v	GENERIC

You can use variables anywhere in an entry line except as a key. For instance, if you have a file server exporting binaries for SPARC and IA architectures from /usr/local/bin/sparc and /usr/local/bin/x86 respectively, the clients can mount through a map entry like the following:

```
/usr/local/bin        -ro     server:/usr/local/bin/$CPU
```

Now the same entry for all clients applies to all architectures.

Note – Most applications written for any of the sun4 architectures can run on all sun4 platforms, so the -ARCH variable is hard-coded to sun4 instead of sun4m.

Maps That Refer to Other Maps

A map entry +*mapname* used in a file map causes automount to read the specified map as if it were included in the current file. If *mapname* is not preceded by a slash, autofs treats the map name as a string of characters and uses the name service switch policy to find it. If the path name is an absolute path name, automount checks a local map of that name. If the map name starts with a dash (-), automount consults the appropriate built-in map, such as xfn or hosts.

This name service switch file contains an entry for autofs labeled as automount, which contains the order in which the name services are searched. The following file is an example of a name service switch 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 contains "switch.so" as a nametoaddr library for "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
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
```

In this example, the local maps are searched before the NIS maps, so you can have a few entries in your local `/etc/auto_home` map for the most commonly accessed home directories, and use the switch to fall back to the NIS map for other entries.

```
bill              cs.csc.edu:/export/home/bill
bonny             cs.csc.edu:/export/home/bonny
```

After consulting the included map, if no match is found, `automount` continues scanning the current map. This means you can add more entries after a + entry.

```
bill              cs.csc.edu:/export/home/bill
bonny             cs.csc.edu:/export/home/bonny
+auto_home
```

The map included can be a local file (remember, only local files can contain + entries) or a built-in map:

```
+auto_home_finance      # NIS+ map
+auto_home_sales        # NIS+ map
+auto_home_engineering  # NIS+ map
+/etc/auto_mystuff      # local map
+auto_home              # NIS+ map
+-hosts                 # built-in hosts map
```

Note – You cannot use + entries in NIS+ or NIS maps.

Executable Autofs Maps

You can create an autofs map that will execute some commands to generate the autofs mount points. You could benefit from using an executable autofs map if you need to be able to create the autofs structure from a database or a flat file. The disadvantage to using an executable map is that the map will need to be installed on each host. An executable map cannot be included in either the NIS or the NIS+ name service.

The executable map must have an entry in the `auto_master` file.

```
/execute    auto_execute
```

Here is an example of an executable map:

```
#!/bin/ksh
#
# executable map for autofs
#

case $1 in
             src)  echo '-nosuid,hard bee:/export1' ;;
esac
```

For this example to work, the file must be installed as `/etc/auto_execute` and must have the executable bit set (set permissions to `744`). Under these circumstances running the following command:

```
% ls /execute/src
```

causes the `/export1` file system from `bee` to be mounted.

Modifying How Autofs Navigates the Network (Modifying Maps)

You can modify, delete, or add entries to maps to meet the needs of your environment. As applications and other file systems that users require change their location, the maps must reflect those changes. You can modify autofs maps at any time. Whether your modifications take effect the next time `automountd` mounts a file system depends on which map you modify and what kind of modification you make.

Default Autofs Behavior With Name Services

Booting invokes autofs using the `/etc/init.d/autofs` script and checks for the master `auto_master` map (subject to the rules discussed subsequently).

Autofs uses the name service specified in the automount entry of the `/etc/nsswitch.conf` file. If NIS+ is specified, as opposed to local files or NIS, all map names are used as is. If NIS is selected and autofs cannot find a map that it needs, but finds a map name that contains one or more underscores, the underscores are changed to dots, which allows the old NIS file names to work. Then autofs checks the map again, as shown in Figure 15–4.

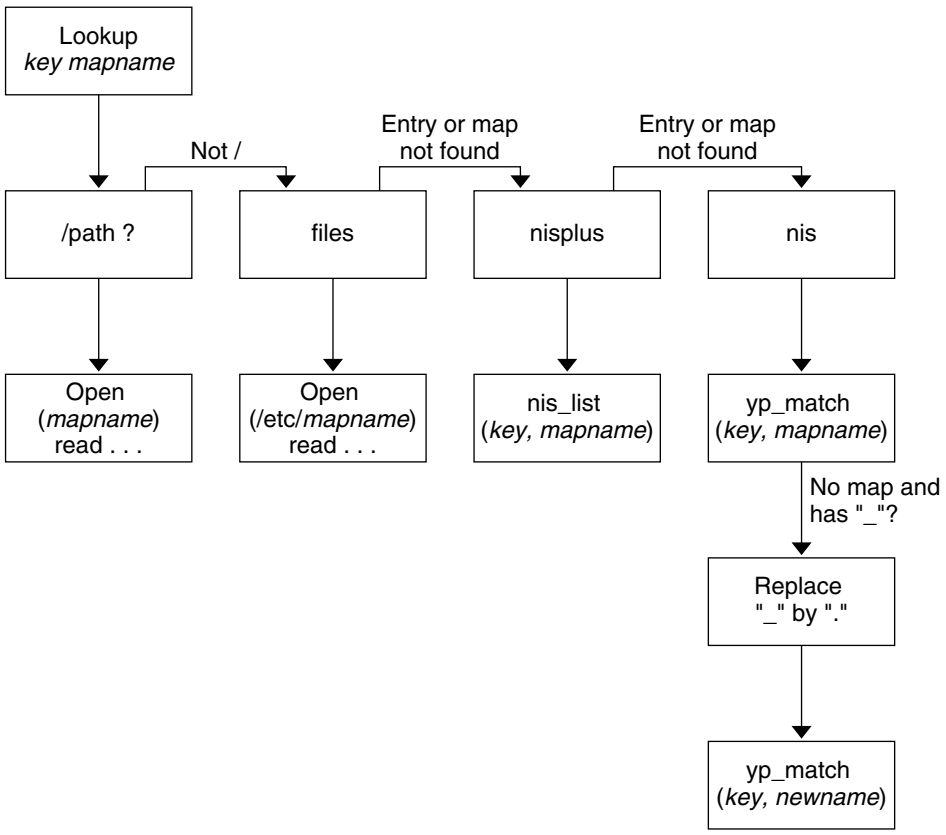


FIGURE 15–4 How Autofs Uses the Name Service

The screen activity for this session would look like the following example.

```
$ grep /home /etc/auto_master
/home           auto_home

$ ypmatch brent auto_home
Can't match key brent in map auto_home.  Reason: no such map in
server's domain.

$ ypmatch brent auto.home
diskus:/export/home/diskus1/&
```

If "files" is selected as the name service, all maps are assumed to be local files in the `/etc` directory. Autofs interprets a map name that begins with a slash (/) as local regardless of which name service it uses.

Autofs Reference

The rest of this chapter describes more advanced autofs features and topics.

Metacharacters

Autofs recognizes some characters as having a special meaning. Some are used for substitutions, some to protect other characters from the autofs map parser.

Ampersand (&)

If you have a map with many subdirectories specified, as in the following, consider using string substitutions.

```
john        willow:/home/john
mary        willow:/home/mary
joe         willow:/home/joe
able        pine:/export/able
baker       peach:/export/baker
```

You can use the ampersand character (&) to substitute the key wherever it appears. If you use the ampersand, the previous map changes to:

```
john        willow:/home/&
mary        willow:/home/&
joe         willow:/home/&
able        pine:/export/&
baker       peach:/export/&
```

You could also use key substitutions in a direct map, in situations like this:

```
/usr/man                        willow,cedar,poplar:/usr/man
```

which you can also write as:

```
/usr/man                        willow,cedar,poplar:&
```

Notice that the ampersand substitution uses the whole key string, so if the key in a direct map starts with a / (as it should), the slash is carried over, and you could not do, for example, the following:

```
/progs                &1,&2,&3:/export/src/progs
```

because autofs would interpret it as:

```
/progs                 /progs1,/progs2,/progs3:/export/src/progs
```

Asterisk (*)

You can use the universal substitute character, the asterisk (*), to match any key. You could mount the /export file system from all hosts through this map entry.

```
*                       &:/export
```

Each ampersand is substituted by the value of any given key. Autofs interprets the asterisk as an end-of-file character.

Special Characters

If you have a map entry that contains special characters, you might have to mount directories that have names which confuse the autofs map parser. The autofs parser is sensitive to names containing colons, commas, spaces, and so on. These names should be enclosed in double quotations, as in the following:

```
/vms    -ro    vmsserver: -  -  - "rc0:dk1 - "
/mac    -ro    gator:/ - "Mr Disk - "
```

CHAPTER 16

SLP Topics

The section contains the following chapters on configuring and deploying Service Location Protocol (SLP) in the Solaris 9 operating environment.

Chapter 17	Information on SLP architecture and implementation.
Chapter 18	SLP configuration considerations and the process the process to enable SLP
Chapter 19	Information and procedures for configuring SLP agents and processes
Chapter 20	Information on advertising SLP legacy services
Chapter 21	Tables listing SLP status codes and message types

CHAPTER 17

SLP (Overview)

The Service Location Protocol (SLP) provides a portable, platform-independent framework for the discovery and provisioning of SLP-enabled network services. This chapter describes the SLP architecture and the Solaris 9 implementation of SLP for IP intranets.

SLP Architecture

This section outlines the fundamental operation of SLP and describes agents and processes used in SLP administration.

SLP provides all of the following services automatically, with little or no configuration.

- **Client application requests** for information required to access a service.
- **Advertisement of services** on network hardware devices or software servers. For example: Printers, file servers, video cameras, HTTP servers, etc.
- **Organization of services and users** into *scopes* composed of logical or functional groups.
- Managed recovery from primary server failures

In addition, you can do the following to administer and tune SLP operation if necessary.

- **Enable SLP logging** to monitor and troubleshoot the SLP operation on your network.
- **Synchronize timing** on SLP message exchanges between agents.

- **Suppress SLP multicasts** to reduce network congestion.
- **Configure scopes** to strategically position SLP directory agents throughout the enterprise.

Summary of the SLP Design

In SLP, software-based agents represent user applications and network services. SLP maintains updated information about the nature and location of enterprise services. Additionally, SLP can use proxy registrations to advertise legacy services that are not SLP-enabled. For more information, see Chapter 20.

SLP Agents and Processes

The following table describes the SLP agents. For expanded definitions of these terms and others used this chapter, refer to Glossary

TABLE 17–1 SLP Agents

SLP Agent	Description
Directory Agent (DA)	Process that caches SLP service advertisements registered by Service Agents (SAs). The DA forwards service advertisements to User Agents (UAs) on demand.
Service Agent (SA)	SLP agent that acts on behalf of a service to distribute service advertisements and to register the service with Directory Agents (DAs).
User Agent (UA)	SLP agent that acts on behalf of a user or application to obtain service advertisement information.
scope	An administrative or logical grouping of services.

The following figure shows the basic agents and processes that implement the SLP architecture. The figure represents a default deployment of SLP. No special configuration has been done. Only two agents are required: the UA and SA. The SLP framework allows the UA to multicast requests for services to the SA. The SA unicasts a reply to the UA. The reply contains the service advertisement when requests for advertised services are received.

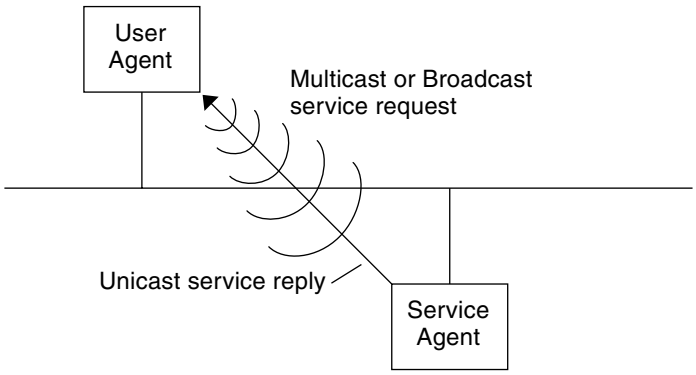


FIGURE 17–1 SLP Basic Agents and Processes

The following figure shows the basic agents and processes that implement the SLP architecture when a DA is deployed in the framework.

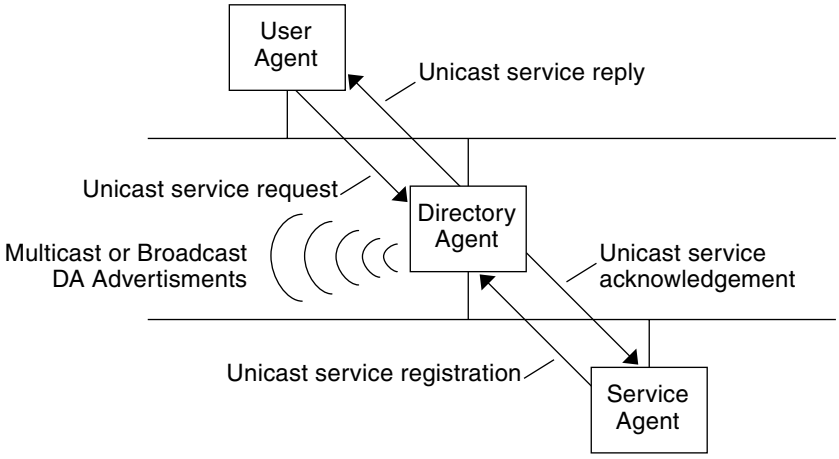


FIGURE 17–2 SLP Architectural Agents and Processes Implemented with a DA

In more complex enterprises, one or more DAs are used. The DA serves as a cache for registered service advertisements. SAs send register messages (`SrvReg`) that list all the services they advertise to DAs. SAs then receive acknowledgments (`SrvAck`) in reply. The service advertisements are refreshed with the DA, or they expire according to the lifetime set for the advertisement. Once a UA discovers a DA, the UA unicasts a request to the DA rather than multicasting requests to SAs.

For more information about Solaris SLP messages, refer to Chapter 21.

SLP Implementation

In the Solaris SLP implementation, the SLP SAs, UAs, DAs, SA servers, scopes, and other architectural components (listed in Table 17–1) are partially mapped into `slpd` and partially into application processes. The SLP daemon, `slpd`, organizes certain off-host SLP interactions and does the following:

- Employs passive and active directory agent discovery for all UAs and SAs on the local host.
- Maintains an updated table of DAs for the use of the UAs and SAs on the local host.
- Acts as a proxy SA server for legacy service advertisements (proxy registration)
- Can be configured to act as a DA

For more information about the SLP daemon, see `slpd`(1M).

In addition to `slpd`, the C/C++ and Java client libraries (`libslp.so` and `slp.jar`) enable access to the SLP framework for UA and SA clients. The client libraries provide the following feature:

- Communication required to register and deregister benefit service advertisements between SA clients and `slpd`.
- UA request capability for UA clients.
- Communication on DA accessibility between `slpd` and UA clients.

In the following figure, the SLP client library in the Service Provider Program implements SA functionality. The Service Provider Program uses the SLP client library to register services with `slpd`, and to deregister them. The SLP client library in the Service Client Program implements UA functionality. The Service Client Program uses the SLP client library to issue multicast and unicast (to DAs) service requests and to query `slpd` for information on DAs. The `slpd` process takes care of all SA functionality, such as answering multicast requests, registering with DAs, and so on.

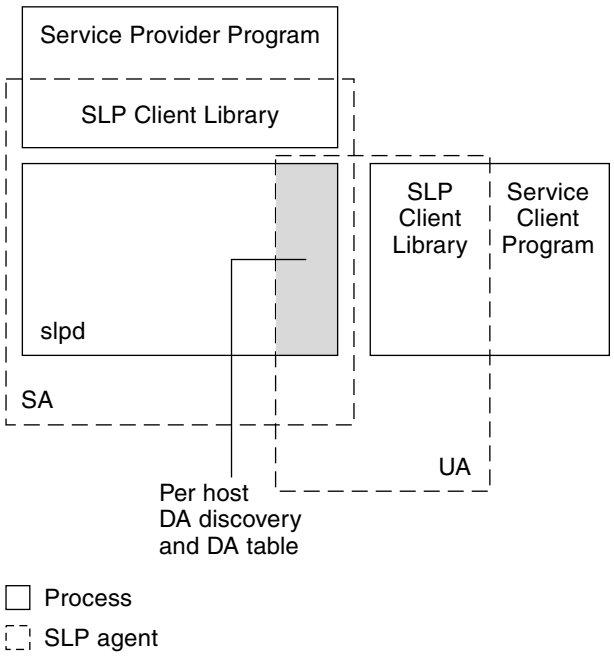


FIGURE 17–3 SLP Implementation

Other SLP Information Sources

Refer to the following documents for further information on SLP:

- Kempf, James, and Pete St. Pierre. *Service Location Protocol for Enterprise Networks.* Wiley and Son, Inc. (ISBN # 0–47–3158–7)
- *Authentication Management Infrastructure Administration Guide* (part # 805–1139–03)
- Guttman, Erik, Charles Perkins, John Veizades, and Michael Day. *Service Location Protocol, Version 2,* from the Internet Engineering Task Force (IETF). Available on line at http://www.ietf.org/ietf/1id-abstracts.txt.

CHAPTER 18

Planning and Enabling SLP (Tasks)

This chapter provides information on planning and enabling SLP. The following sections discuss SLP configuration and the process for enabling SLP.

SLP Configuration Considerations

The SLP daemon is pre-configured with default properties for installation with the Solaris 9 operating environment. If your enterprise functions well with default settings, the SLP deployment requires virtually no administration.

In some cases, however, you might want to modify the SLP properties to tune network operations or to activate certain features. With a few configuration changes you can enable SLP logging, for example. The information in the SLP log and in `snoop` traces can then help you decide if additional configuration is necessary.

SLP configuration properties reside in the `slp.conf` file located in the `/etc/inet` directory. If you decide to change the default property settings, consult the following chapters for the appropriate procedures.

Before you modify SLP configuration settings, consider the following questions related to key aspects of network administration:

- **Technologies**–What network technologies are operating in the enterprise?
- **Traffic**–How much network traffic can the technologies handle smoothly?
- **Services**–How many services, of what type, are available on the network?

- **Users**–How many users are on the network? What services do they require? Where are users located in relation to their most frequently accessed services?

Deciding What to Reconfigure

You can use the SLP-enabled `snoop` utility and SLP logging utilities to decide if and what reconfiguration is necessary. For example, you might reconfigure certain properties to do the following:

- Accommodate a mix of network media that have varying latencies and bandwidth characteristics
- Recover the enterprise from network failures or unplanned partitioning
- Add DAs to reduce proliferation of SLP multicasts
- Implement new scopes to organize users with their most frequently accessed services

Using `snoop` to Monitor SLP Activity

The `snoop` utility is a passive administrative tool that provides network traffic information. The utility itself generates minimal traffic and enables you to watch all activity on your network as it occurs.

The `snoop` utility provides traces of the actual SLP message traffic. For example, when you run `snoop` with the `slp` command line argument, the utility displays traces with information on SLP registrations and deregistrations. You can use the information to gauge the network load by checking what services are being registered and how much reregistration activity its occurring.

The `snoop` utility is also useful for observing the traffic flow between SLP hosts in your enterprise. When you run `snoop` with the `slp` command line argument, you can monitor the following types of SLP activity to determine whether network or agent reconfiguration is needed:

- **The number of hosts using a particular DA**–Use this information to decide whether to deploy additional DAs for load balancing.
- **What hosts are using which DAs**–Use this information to help you determine whether to configure certain hosts with new or different scopes.
- **If UA requests time out or DA acknowledgement is slow**–You can determine whether a DA is overloaded by monitoring UA time-outs and retransmissions. You can also check to see if the DA requires more than a few seconds to send

registration acknowledgments to an SA. Use this information to rebalance the network load on the DA, if necessary, by deploying additional DAs or changing the scope configurations.

Using `snoop` with the `-V` (verbose) command line argument, you can obtain registration lifetimes and value of the fresh flag in `SrvReg` to determine whether the number of reregistrations should be reduced.

You can also use `snoop` to trace other kinds of SLP traffic such as the following:

- Traffic between UA clients and DAs
- Traffic between multicasting UA clients and replying SAs

For more information about `snoop`, refer to the `snoop`(1M).

Tip – Use the `netstat` command in conjunction with `snoop` to view traffic and congestion statistics. For more information about `netstat`, refer to `netstat`(1M).

▼ How to Use `snoop` to Run SLP Traces

1. **Become superuser.**
2. **Type the following command to invoke `snoop` with the `slp` command line argument.**

   Brief Mode:

   ```
   # snoop slp
   ```

   When you run `snoop` in the default *brief* mode, ongoing output is delivered to your screen. SLP messages are truncated to fit on one line per SLP trace.

   Verbose Mode:

   ```
   # snoop -v slp
   ```

   When you run `snoop` in *verbose* mode, `snoop` delivers ongoing, unabbreviated output to your screen which provides the following information:

   - Complete address of the service URL
   - All service attributes
   - The registration lifetime
   - All security parameters and flags, if any are available

Note – You can use `slp` command line argument with other `snoop` options.

Analyzing a `snoop slp` Trace

In the following example, `slpd` runs on *slphost1* in the default mode as an SA server. The SLP daemon initializes and registers *slphost2* as an echo server. Then, the `snoop slp` process is invoked on *slphost1*

Note – To make it easier to describe the trace results, the lines in the following `snoop` output are flagged with line numbers.

```
[1]slphost1 -> 239.255.255.253 SLP V@ SrvRqst [24487] service:directory-agent []
[2]slphost2 -> slphost1 SLP V2 DAAdvert [24487] service:directory-agent://129
[3]slphost1 -> 239.255.255.253 SLP V2 SrvRqst [24487] service:directory-agent []
[4]slphost1 -> 239.255.255.253 SLP V2 SrvRqst [24487] service:directory-agent []
[5]slphost1 -> slphost2 SLP V2 SrvReg [24488/tcp]service:echo.sun:tcp://slphost1:
[6]slphost2 -> slphost1 SLP V2 SrvAck [24488/tcp] ok
[7]slphost1 -> slphost2 SLP V2 SrvDereg [24489/tcp] service:echo.sun:tcp://slphost1:
[8]slphost2 -> slphost1 SLP V2 SrvAck [24489/tcp] ok
```

1. Shows `slpd` on *slphost1* performing active directory agent discovery by multicasting to the SLP multicast group address in search of directory agents. The message number, 24487, for the active discovery is indicated in square brackets in the trace display.
2. Indicates that the active discovery request 24487 from trace 1 is answered by `slpd` running as a DA on the host *slphost2*. The service URL from *slphost2* has been truncated to fit on a single line. The DA has sent a DA advertisement in reply to the multicast directory agent discovery message, as indicated by the matching message numbers in traces 1 and 2.
3. Shows multicasts from the UAs on *slphost1* for additional DAs. Because *slphost2* has already answered the request, it refrains from responding again, and no other DAs reply.
4. Repeats the multicast operation shown in the previous line.
5. Shows a `slpd` on *slphost1* forwarding SA client registrations to the DA on *slphost2*. A unicast service registration (`SrvReg`) for an echo server is made by *slphost1* to the DA on *slphost2*.

   The `/tcp` parameter appended to the message number on lines 5 and 6 indicates that the message exchange occurred by TCP.
6.
7. Shows *slphost2* responding the *slphost1*`SrvReg` with an service acknowledgement (`SrvAck`) indicating the registration is successful.

Traffic between the echo server that runs the SA client and the SLP daemon on *slphost1* does not appear in the `snoop` trace. This is because the `snoop` operation is performed over the network loopback. .

8. Shows the echo server on *slphost1* deregistering the echo service advertisement. The SLP daemon on *slphost1* forwards the deregistration to the DA on *slphost2*.
9. Shows *slphost2* responding to *slphost1* with a service acknowledgment (`SrvAck`), to indicate that the deregistration is successful.

Where to Go From Here

After monitoring the SLP traffic, you can use the information collected from the `snoop` traces to help determine whether any reconfiguration of the SLP defaults is needed. Use the related information in Chapter 19 for configuring SLP property settings. For more information about SLP messaging and service registrations, refer to Chapter 21.

Enabling SLP

SLP is enabled by running the SLP daemon, `slpd`. The supported interface for starting `slpd` is the `/etc/init.d/slpd` script, which starts the daemon only if the SLP configuration file, `/etc/inet/slp.conf`, exists. The Solaris operating environment includes the file `/etc/inet/slp.conf.example`. Rename this file to `/etc/inet/slp.conf` to enable SLP at boot time.

CHAPTER 19

Administering SLP (Tasks)

The following sections provide information and tasks for configuring SLP agents and processes.

Configuring SLP Properties

SLP configuration properties control network interactions, SLP agent characteristics, status, and logging. In most cases, the default configuration of these properties requires no modification. However, you can use the procedures in this chapter when the network medium or topology changes and to take actions such as the following:

- Compensate for network latencies.
- Reduce congestion on the network.
- Add agents or reassign IP addresses.
- Activate SLP logging.

You can edit the SLP configuration file, `/etc/inet/slp.conf`, to perform operations such as those shown in the following table:

TABLE 19–1 SLP Configuration Operations

Operation	Description
Specify whether `slpd` should act as a DA server. SA server is the default.	Set the `net.slpisDA` property to `true`.
Set timing for DA multicast messages.	Set the `net.slp.DAHeartBeat` property to control how often a DA multicasts an unsolicited DA advertisement.
Enable DA logging to monitor network traffic.	Set the `net.slp.traceDATraffic` property to `true`.

SLP Configuration File: Basic Elements

The `/etc/inet/slp.conf` file defines and activates all SLP activity each time you restart the SLP daemon. The configuration file consists of the following elements:

- Configuration properties
- Comment lines and notations

Configuration Properties

All of the basic SLP properties, such as `net.slp.isDA` and `net.slp.DAHeartBeat`, are named in the following format:

```
net.slp.<keyword>
```

SLP behavior is defined by the value of a property or a combination of properties in the `slp.conf` file. Properties are structured as key-value pairs in the SLP configuration file. As shown in the following example, a key-value pair consists of a property name and an associated setting.

```
<property name>=<value>
```

The key for each property is the property name. The value sets the numeric (distance or time), true/false state, or string value parameters for the property. Property values consist of one of the following data types:

- True/False setting (Boolean)
- Integers
- List of integers
- Strings
- List of strings

Comment Lines and Notations

You can add comments to the `slp.conf` file that describe the nature and function of the line. Comment lines are optional in the file, but can be useful for administration.

Note – Settings in the configuration file are case insensitive. Use non-ASCII characters for escaping.

▼ How to Change Your SLP Configuration

Use this procedure to change the property settings in your SLP configuration file.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Edit the property settings in the `/etc/inet/slp.conf` file as necessary.**

   Refer to "Configuration Properties" on page 258 for general information about the SLP property settings. Consult the following sections for examples of different scenarios in which you might change the `slp.conf` properties. See `slp.conf`(4).
5. **Save your changes and close the file.**
6. **Restart `slpd` to activate your changes. Type the following command:**

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

Note – The SLP daemon obtains information from the configuration file when you stop or start `slpd`.

For example, you can change the SA server default to enable `slpd` to operate as a DA server by setting the `net.slp.isDA` property to `true` in the `slpd.conf` file.

```
net.slp.isDA=true
```

In each area, various properties control different aspects of the configuration. The following sections describe different scenarios in which you might change the default property settings used in SLP configuration.

Modifying DA Advertising and Discovery Frequency

In situations such as the following, you can modify properties that control the timing of DA advertisements and discovery requests .

- When you want the SA or UA to obtain DA configuration information statically from the `net.slp.DAAddresses` property in the `slp.conf` file, you can disable DA discovery.
- When the network is subject to recurrent partitioning, you can change the frequency of passive advertisements and active discovery.
- If UA and SA clients access DAs on the other side of a dial-up connection, you can decrease the DA heartbeat frequency and the active discovery interval to reduce the number of times a dialup line is activated.
- If network congestion is high, you can limit multicasting.

The procedures in this section explain how to modify the following properties.

TABLE 19–2 DA Advertisement Timing and Discovery Request Properties

Property	Description
`net.slp.passiveDADetection`	Boolean that specifies whether `slpd` listens for unsolicited DA advertisements.
`net.slp.DAActiveDiscoveryInterval`	Value that specifies how often `slpd` performs active DA discovery for a new DA
`net.slp.DAHeartBeat`	Value that specifies how often a DA multicasts an unsolicited DA advertisement

Limiting UAs and SAs to Statically Configured DAs

In some cases, you might need to limit UAs and SAs to obtaining DA addresses from the static configuration information in the `slp.conf` file. In the next procedure, you can modify two properties that cause `slpd` to obtain DA information exclusively from the `net.slp.DAAddresses` property.

▼ How to Limit UAs and SAs to Statically Configured DAs

Use the following procedure to change the `net.slp.passiveDADetection` property to and the `net.slp.DAActiveDiscoveryInterval` properties.

Note – Use this procedure only on hosts that execute UAs and SAs which are restricted to static configurations.

---

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Set the `net.slp.passiveDADetection` property to `False` in the `slp.conf` file to disable passive discovery. This causes `slpd` to ignore unsolicited DA advertisements.**

   ```
   net.slp.passiveDADetection=False
   ```

5. **Set the `net.slp.DAActiveDiscoveryInterval` to `-1` to disable initial and periodic active discovery.**

   ```
   net.slp.DAActiveDiscoveryInterval=-1
   ```

6. **Save your changes and close the file.**
7. **Restart `slpd` to activate your changes. Type the following command:**

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

Configuring DA Discovery for Dial-up Networks

If the UAs or SAs are separated from the DA by a dial-up network, you can configure DA discovery to reduce or eliminate the number of discovery requests and DA advertisements. Dial-up networks usually incur a charge when activated. Minimizing extraneous calls can reduce the cost of using the dial-up network.

Note – You can disable DA discovery completely with the method described in "Limiting UAs and SAs to Statically Configured DAs" on page 260.

▼ How to Configure DA Discovery for Dial-up Networks

You can use the following procedure to reduce unsolicited DA advertisements and active discovery by increasing the DA heartbeat period and the active discovery interval.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Increase the `net.slp.DAHeartbeat` property in the `slpd.conf` file:**

   ```
   net.slp.DAHeartbeat value
   ```

value	A 32–bit integer that sets the number of seconds for the passive DA advertisement heartbeat. Default Value=10800 seconds (3 hours) Range of Values=2000–259200000 seconds

   For example, you can set the DA heartbeat to approximately 18 hours on a host executing a DA:

   ```
   net.slp.DAHeartbeat=65535
   ```

5. **Increase the** `net.slp.DAActiveDiscoveryInterval` **property in the** `slpd.conf` **file:**

   `net.slp.DAActiveDiscoveryInterval` *value*

value	A 32–bit integer that sets the number of seconds for DA active discovery queries. Default Value=900 seconds (15 minutes) Range of Values=300–10800 seconds

   For example, you can set the DA active discovery interval to 18 hours on a host executing a UA and an SA :

   ```
   net.slp.DAActiveDiscoveryInterval=65535
   ```

6. **Save your changes and close the file.**

7. **Restart** `slpd` **to activate your changes. Type the following command:**

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

Configuring the DA Heartbeat for Frequent Partitions

SAs are required to register with all DAs that support their scopes. A DA can appear after `slpd` has performed active discovery. If the DA supports `slpd` scopes, the SLP daemon registers all advertisements on its host with the DA.

One way `slpd` discovers DAs is by the initial unsolicited advertisement a DA sends out when it boots. `slpd` uses the periodic unsolicited advertisement (the heartbeat) to determine whether a DA is still active, and removes the DAs it uses and offers to UAs if a heartbeat fails to appear.

Finally, when a DA undergoes a controlled shutdown, it transmits a special DA advertisement that informs listening SA services that it will be out of service. The SLP daemon also uses this advertisement to remove inactive DAs from the cache.

If your network is subject to frequent partitions and SAs are long-lived, `slpd` can remove cached DAs during the partitioning if heartbeat advertisements are not received. By decreasing the heartbeat time, you can decrease the delay before a deactivated DA is restored to the cache after the partition is repaired.

▼ How to Configure DA Heartbeat for Frequent Partitions

Use the following procedure to change the `net.slp.DAHeartBeat` property to decrease the DA heartbeat period.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Decrease the `net.slp.DAHeartBeat` value to one hour (3600 seconds).By default, the DA heartbeat period is set to 3 hours (10800 seconds).**

   ```
   net.slp.DAHeartBeat=3600
   ```

5. **Save your changes and close the file.**
6. **Restart `slpd` to activate your changes. Type the following command:**

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

Note – If DA discovery is completely disabled, the `net.slp.DAAddresses` property must be set in `slp.conf` on the hosts executing UAs and SAs so that they access the correct DA.

Relieving Network Congestion

If network congestion is high, you might limit the amount of multicast. If DAs have not already been deployed in the network, deploying DAs can drastically cut back on the amount of SLP-related multicast.

However, even after DAs are deployed, multicast is still necessary for DA discovery. You can reduce the amount of multicast necessary for DA discovery using the method described in “How to Configure DA Discovery for Dial-up Networks” on page 262. You can completely eliminate multicast for DA discovery using the method described in “Limiting UAs and SAs to Statically Configured DAs” on page 260.

Accommodating Different Network Media, Topology, or Configuration

You can use the procedures in this section to can tune SLP performance by modifying one of the following parameters:

This section describes possible scenarios in which you can change the following properties to tune SLP performance.

TABLE 19–3 SLP Performance Properties

Property	Description
`net.slp.passiveDAAttributes`	The `minimum refresh interval` that a DA will accept for advertisements.
`net.slp.multicastTTL`	The *time to live* value specified for multicast packets.
`net.slp.MTU`	The byte size set for network packets. The size includes IP and TCP or UDP headers.
`net.slp.isBroadcastOnly`	Boolean set to indicate if broadcast should be used for DA and non-DA based service discovery.

Reducing SA Reregistrations

SAs periodically need to refresh their service advertisements before lifetimes expire. If a DA is handling an extremely heavy load from many UAs and SAs, frequent refreshes can cause the DA to become overloaded. If the DA becomes overloaded, UA requests start to time out and are then dropped. UA request timeouts have many possible causes. Before you assume that DA overload is the problem, use a `snoop` trace to check the lifetimes of service advertisements registered with a service registration. If the lifetimes are short and reregistrations are occurring often, the timeouts are probably due to frequent reregistrations.

Note – A service registration is a *reregistration* if the fresh flag is not set. See Chapter 21 for more information on service registration messages.

▼ How to Reduce SA Reregistrations

Use the following procedure to increase the `minimum refresh interval` for SAs to reduce reregistrations.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Increase the value of the `min-refresh-interval` attribute of the `net.slp.DAAttributes` property..**

   The default minimum reregistration period is zero. The zero default allows SAs to reregister at any point. In the following example, the interval is increased to 3600 seconds (one hour).

   ```
   net.slp.DAAttributes(min-refresh-interval=3600)
   ```

5. **Save your changes and close the file.**
6. **Restart `slpd` to activate your changes. Type the following command:**

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

Configuring the Multicast Time to Live Property

The multicast time to live (`net.slp.multicastTTL` property) determines the range over which a multicast packet is propagated on your intranet. The multicast TTL is configured by setting the `net.slp.multicastTTL` property to an integer between 1 and 255. The default value of the multicast TTL is 255, which means, theoretically, that the packet routing is unrestricted. However, a TTL of 255 causes a multicast packet to penetrate the intranet to the border routers on the edge of your administrative domain. Correct configuration of multicast on border routers is required to prevent multicast packets from leaking into the Internet's multicast backbone, or to your ISP.

Multicast TTL scoping is similar to standard IP TTL, with the exception that a TTL comparison is made. Each interface on a router that is multicast-enabled is assigned a TTL value. When a multicast packet arrives, the router compares the TTL of the packet

with the TTL of the interface. If the TTL of the packet is greater than or equal to the TTL of the interface, the packet TTL is reduced by one, as is the case with the standard IP TTL. If the TTL becomes zero, the packet is discarded. When you use TTL scoping for SLP multicasting, your routers be properly configured to limit packets to a particular subsection of your intranet.

▼ How to Configure the Multicast Time to Live Property

Use the following procedure to reset the `net.slp.multicastTTL` property.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Change the `net.slp.multicastTTL` property in the `slpd.conf` file:**

   `net.slp.multicastTTL` *value*

value	A positive integer less than or equal to 255 that defines the multicast TTL.

   Note – You can reduce the range of multicast propagation by reducing the TTL value. If the TTL value is 1, then the packet is restricted to the subnet. If the value is 32, the packet is restricted to the site. Unfortunately, the term *site* is not defined by RFC 1075, where multicast TTLs are discussed. Values above 32 refer to theoretical routing on the Internet and should not be used. Values below 32 can be used to restrict multicast to a set of accessible subnets, provided the routers are properly configured with TTLs.

5. **Save your changes and close the file.**
6. **Restart `slpd` to activate your changes. Type the following command:**

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

Configuring the Packet Size

The default packet size for SLP is 1400 bytes. The size should be sufficient for most local area networks. For wireless networks or wide area networks, you can reduce the packet size to avoid message fragmentation and reduce network traffic. For local area networks that have larger packets, increasing the packet size can improve performance. You can determine whether the packet size needs to be reduced by checking the minimum packet size for your network. If the network medium has a smaller packet size, you can reduce the `net.slp.MTU` value accordingly.

You can increase the packet size if your network medium has larger packets. However, unless the service advertisements from SAs or queries from UAs frequently overflow the default packet size, configuration should not be necessary. You can use `snoop` to determine whether UA requests often overflow the default packet size and roll over to use TCP rather than UDP.

The `net.slp.MTU` property measures the complete IP packet size, including the link layer header, the IP header, the UDP or TCP header, and the SLP message.

▼ How to Configure the Packet Size

Use the following procedure to change the default packet size by adjusting the `net.slp.MTU` property.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Change the `net.slp.MTU` property in the `slpd.conf` file:**

   `net.slp.MTU` *value*

value	A 16–bit integer that specifies the network packet size, in bytes Default Value=1400 Range of Values=128–8192

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

6. **Restart `slpd` to activate your changes. Type the following command:**

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

Configuring Broadcast Only Routing

SLP is designed to use multicast for service discovery in the absence of DAs and for DA discovery. For various reasons, some networks do not deploy multicast routing. If your network does not deploy multicast routing, you can configure SLP to use broadcast by setting the `net.slp.isBroadcastOnly` property to `True`.

Unlike multicast, broadcast packets do not propagate across subnets by default. For this reason, service discovery without DAs in a non-multicast network works only on a single subnet. In addition, special considerations are required when deploying DAs and scopes in networks where broadcast is used. A DA on a multihomed host can bridge service discovery between multiple subnets with multicast disabled. See "DA Placement and Scope Name Assignment" on page 284 for more information on deploying DAs on multihomed hosts.

▼ How to Configure Broadcast Only Routing

Use the following procedure to change `net.slp.isBroadcastOnly` property to `True`.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Change the `net.slp.BroadcastOnly` property in the `slpd.conf` file to `True`:**

   ```
   net.slp.BroadcastOnly=True
   ```

5. **Save your changes and close the file.**
6. **Restart `slpd` to activate your changes. Type the following command:**

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

Modifying Timeouts on SLP Discovery Requests

There are two situations in which you might change the timeouts for SLP discovery requests:

- If the SLP agents are separated by multiple subnets, dial-up lines, or other WANs, the network latency can be high enough that the default timeouts are insufficient for a request or registration to complete. Conversely, if your network is low latency, performance can be improved by decreasing the timeouts.
- If the network is subject to heavy traffic or a high collision rates, the maximum period that SAs and UAs need to wait before sending a message might be insufficient to assure collision-free transactions.

Changing Default Timeouts

High network latency can cause UAs and SAs to time out before a response returns for requests and registrations. Latency can be a problem if a UA is separated from an SA, or if both a UA and SA are separated from a DA—either by multiple subnets, a dial-up line, or a WAN. You can determine if latency is a problem by checking to see whether SLP requests are failing due to timeouts on UA and SA requests and registrations. You can also use the `ping` command to measure the actual latency.

The following table lists of configuration properties that control timeouts. You can use the procedures in this section to modify these properties.

TABLE 19–4 Timeout Properties

Property	Description

`net.slp.multicastTimeouts`
`net.slp.DADiscoveryTimeouts`
`net.slp.datagramTimeouts`
	The properties that control timeouts for repeated multicast and unicast UDP message transmissions before transmission is abandoned.

`net.slp.multicastMaximumWait`	The property that controls the maximum amount of time a multicast message is transmitted before it is abandoned.

TABLE 19–4 Timeout Properties *(Continued)*

Property	Description
`net.slp.datagramTimeouts`	The upper bound of a DA timeout specified by the sum of values listed for this property. A UDP datagram is repeatedly sent to a DA until a response is received or the timeout bound is reached.

If frequent timeouts are occurring during multicast service discovery or DA discovery, the `net.slp.multicastMaximumWait` property should be increased from the default value of 15000 milliseconds (15 seconds). Increasing the maximum wait period allows more time for requests on high latency networks to complete. After you change the `net.slp.multicastMaximumWait`, you should also modify the `net.slp.multicastTimeouts` and `net.slp.DADiscoveryTimeouts`. The sum of the timeout values in for each property equals the `net.slp.multicastMaximumWait` value.

▼ How to Change Default Timeouts

Use the following procedure to change the SLP properties that control timeouts.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Change the `net.slp.multicastMaximumWait` property in the `slpd.conf` file:**

   ```
   net.slp.multicastMaximumWait=value
   ```

   value — A 32–bit integer that lists the sum of the values set for `net.slp.multicastTimeouts` and `net.slp.DADiscoveryTimeouts`

   Default Value=15000 milliseconds (15 seconds)

   Range of Values=1000 to 60000 milliseconds

   For example, if you determine that multicast requests require 20 seconds (20000 milliseconds), you would adjust the values listed for `net.slp.multicastTimeouts` and the `net.slp.DADiscoveryTimeouts` properties to equal 20000 milliseconds.

   ```
   net.slp.multicastMaximumWait=20000
   net.slp.multicastTimeouts=2000,5000,6000,7000
   ```

```
net.slp.DADiscoveryTimeouts=3000,3000,6000,8000
```

5. **If necessary, change the** `net.slp.datagramTimeouts` **property in the** `slpd.conf` **file:**

   ```
   net.slp.datagramTimeouts=value
   ```

value	A list of 32–bit integers that specify timeouts, in milliseconds, to implement unicast datagram transmission to DAs. Default=3000,3000,3000

   For example, you can by increase the datagram timeout to 20000 milliseconds to avoid frequent timeouts

   ```
   net.slp.datagramTimeouts=2000,5000,6000,7000
   ```

   In high performance networks, you can reduce the timeout bound for multicast and unicast UDP datagram transmission. This will reduce the amount of latency in satisfying SLP requests.

6. **Save your changes and close the file.**

7. **Restart** `slpd` **to activate your changes. Type the following command:**

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

Configuring the Random Wait Bound

In networks with heavy traffic or a high collision rate, communication with a DA might be affected. When collision rates are high, the sending agent must retransmit the UDP datagram. You can determine if retransmission is occurring by using `snoop` to monitor traffic on a network of hosts running `slpd` as an SA server and a host running `slpd` as a DA. If multiple service registration messages for the same service appear in the `snoop` trace from the host running `slpd` as an SA serve , you might have a problem with collisions.

Collisions can be particularly troubling at boot time. When a DA first comes up, it sends out unsolicited advertisements and the SAs respond with registrations. SLP requires the SAs to wait for a random amount of time after receiving a DA advertisement before responding. The random wait bound is uniformly distributed with a maximum value controlled by the `net.slp.randomWaitBound`. The default random wait bound is 1000 milliseconds (1 second).

▼ How to Configure the Random Wait Bound

Use the following procedure to change the `net.slp.RandomWaitBound` property in the `slp.conf` file.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Change the `net.slp.RandomWaitBound` property in the `slpd.conf` file:**

   ```
   net.slp.RandomWaitBound=value
   ```

value	The upper bound for calculating the random wait time before attempting to contact a DA Default Value=1000 milliseconds (1 second Range of Values=1000 to 3000 milliseconds

   For example, you can lengthen the maximum wait to 5000 milliseconds (5 seconds).

   ```
   net.slp.randomWaitBound=5000
   ```

   When you lengthen the random wait bound, there is a longer delay in registration. SAs can complete registrations with newly discovered DAs more slowly to avoid collisions and timeouts.

5. **If necessary, change the `net.slp.datagramTimeouts` property in the `slpd.conf` file:**

   ```
   net.slp.datgramTimeouts=value
   ```

value	A list of 32–bit integers that specify timeouts, in milliseconds, to implement unicast datagram transmission to DAs. Default=3000,3000,3000

   For example, you can by increase the datagram timeout to 20000 milliseconds to avoid frequent timeouts

   ```
   net.slp.datagramTimeouts=2000,5000,6000,7000
   ```

   In high performance networks, you can reduce the timeout bound for multicast and unicast UDP datagram transmission. This will reduce the amount of latency in satisfying SLP requests.

6. **Save your changes and close the file.**

7. **Restart `slpd` to activate your changes. Type the following command:**

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

Deploying Scopes

With scopes, you can provision services that depend on the logical, physical, and administrative grouping of users. You can use scopes to administer access to service advertisements.

Use the `net.slp.useScopes` property to create scopes. For example, in the `/etc/inet/slp.conf` file on a host, add a new scope, called `newscope`, as shown:

```
net.slp.useScopes=newscope
```

If, for example, your organization had an alcove of networked devices, such as printers and fax machines, at the end of the south hall on the second floor of Building 6. These devices might be used by everyone on the second floor, or the use could be restricted to members of a certain department. Scopes provide a way to provision access to the service advertisements for these machines.

If the devices are dedicated to a single department, you can create a scope. with the department name. For example, `mktg`. Devices that belong to other departments can be configured with different scope names.

In another scenario, the departments might be dispersed. For instance, the mechanical engineering and the CAD/CAM departments might be split between floors 1 and 2. However, you can provide the floor 2 machines for the hosts on both floors by assigning them to the same scope. You can deploy scopes in any manner that operates well with your network and users.

Note – UAs that have particular scope are not prevented from actually using services advertised in other scopes. Configuring scopes controls only which service advertisements a UA sees. It is up to the service itself to enforce any access control restrictions.

When to Configure Scopes

SLP can function adequately without any scope configuration at all. In the Solaris operating environment, the default scope for SLP is `default`. If no scopes are configured, `default` is the scope of all SLP messages.

You can configure scopes in any of the following circumstances.

1. The organizations you support want to restrict service advertisement access to their own members.
2. The physical layout of the organization you support suggests that services in a certain area be accessed by particular users.
3. There is a need to partition the service advertisements that users are allowed to see.

An example of the first case was cited in "Configuring DA Discovery for Dial-up Networks" on page 261. An example of the second case is a situation in which an organization is spread between two buildings, and you want users in a building to access local services in that building. Users in Building 1 can be configured with the `B1` scope, while users in Building 2 can be configured with the `B2` scope.

Considerations When Configuring Scopes

When you modify the `net.slp.useScopes` property in the `slpd.conf` file, you configure scopes for all agents on the host. If the host is running any SAs or is acting as a DA, you must configure this property if you want to configure the SAs or DA into scopes other than `default`. If only UAs are running on the machine and the UAs should discover SAs and DAs supporting scopes other than `default`, you do not need to configure the property unless you want to restrict the scopes the UAs use. If the property is not configured, UAs will automatically discover available DAs and scopes through `slpd`, which uses active and passive DA discovery to find DAs, or through SA discovery if no DAs are running. On the other hand, if the property is configured, UAs will use only the configured scopes and not discard them.

If you decide to configure scopes, you should consider keeping the `default` scope on the list of configured scopes unless you are sure that all SAs in the network have scopes configured. If any SAs are left unconfigured, UAs with configured scopes will

be unable to find them, because the unconfigured SAs automatically have scope `default`, but the UAs have the configured scopes.

If you also decide to configure DAs by setting the `net.slp.DAAddresses` property, be sure that the scopes supported by the configured DAs are the same as the scopes that you have configured with the `net.slp.useScopes` property. If this is not the case, `slpd` prints an error message when it is restarted.

▼ How to Configure Scopes

Use the following procedure to add scope names to the `net.slp.useScopes` property in the `slp.conf` file.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Change the `net.slp.useScopes` property in the `slpd.conf` file:**

   `net.slp.useScopes=<`*scope names*`>`

scope names	A list of strings that indicates which scopes a DA or SA is allowed to use when making requests, or what scopes a DA must support. Default Value=Default for SA and DA/Unassigned for UA

Note – Use the following to construct scope names:

- Any alphanumeric characters, upper or lower case
- Any punctuation characters (except for: ´´, \, !, <, =, >, and ~).
- Spaces that are considered part of the name
- Non-ASCII characters

  You use a backslash to escape non-ASCII characters. For example, UTF-8 encoding uses `0xc3a9` hex code to represent the letter *e* with the French *aigue* accent. If the platform does not support UTF-8, you use the UTF-8 hex code as the escape sequence `\c3\a9`.

For example, to specify scopes for `eng` and `mktg` groups in `bldg6`, you change the `net.slp.useScopes` line to:

```
net.slp.useScopes=eng,mktg,bldg6
```

5. **Save your changes and close the file.**
6. **Restart `slpd` to activate your changes. Type the following command:**

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

Deploying DAs

This chapter describes the strategic deployment of DAs in a network running SLP.

SLP functions adequately with only the base agents, UA and SA, and without any deployed DAs or configured scopes (all agents automatically have the `default` scope). However, DAs serve as caches for service advertisements, and they are useful for reducing multicast. This capability enables SLP to accommodate larger networks.

Why Deploy an SLP DA?

The primary reason to deploy DAs is to reduce the amount of multicast traffic involved in service discovery. In a large network with many UAs and SAs, the amount of multicast traffic involved in service discovery can become so large that network performance degrades. By deploying one or more DAs, UAs must unicast to DAs for service and SAs must register with DAs using unicast. The only SLP-registered multicast in a network with DAs is for active and passive DA discovery.

SAs register automatically with any DAs they discover within a set of common scopes, rather than taking multicast service requests. Multicast requests in scopes that are not supported by the DA are still answered directly by the SA, however.

Service requests from UAs are unicast to DAs rather than multicast onto the network when a DA is deployed within the UA's scopes. Consequently, DAs within the UA's scopes reduce multicast. By eliminating multicast for normal UA requests, delays and timeouts are eliminated.

DAs act as a focal point for SA and UA activity. Deploying one or several DAs for a collection of scopes provides a centralized point for monitoring SLP activity. By turning on DA logging, it is easier to monitor registrations and requests than to check the logs from multiple SAs scattered around the network. You can deploy any number of DAs for a particular scope or scopes, depending on the need to balance the load.

In networks without multicast routing enabled, you can configure SLP to use broadcast. However, broadcast is very inefficient, because it requires each host to process the message. Broadcast also does not normally propagate across routers. As a result, in a network without multicast, DAs can be deployed on multihomed hosts to bridge SLP advertisements between the subnets. See "Configuring Broadcast Only Routing" on page 269 for more information on how to deploy SLP on networks without multicast enabled.

Finally, the Solaris SLPv2 DA supports interoperability with SLPv1. SLPv1 interoperability is enabled by default in the Solaris DA. If your network contains SLPv1 devices, like printers, or you need to interoperate with Novell Netware 5, which uses SLPv1 for service discovery, you should deploy a DA. Without a DA, the Solaris SLP UAs are unable to find SLPv1 advertised services.

When to Deploy DAs

Deploy DAs on your enterprise if any of the following conditions are true:

- Multicast SLP traffic exceeds 1% of the bandwidth on your network, as measured by `snoop`.
- UA clients experience long delays or timeouts during multicast service requests.
- You would like to centralize monitoring of SLP service advertisements for particular scopes on one or several hosts.
- Your network does not have multicast enabled and consists of multiple subnets that must share services.
- Your network employs devices that support earlier versions of SLP (SLPv1) or you would like SLP service discovery to interoperate with Novell Netware 5.

▼ How to Deploy DAs

Use the following procedure to set the `net.slp.isDA` property to `True` in the `slp.conf` file.

Note – You can assign only one DA per host.

---

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Set the `net.slp.isDA` property in the `slpd.conf` file to `True`:**

   ```
   net.slp.isDA=True
   ```

5. **Save your changes and close the file.**
6. **Restart `slpd` to activate your changes. Type the following command:**

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

▼ How to Deploy a DA

Use this procedure to deploy a DA by changing the default setting for the `net.slp.isDA` property in the host's `slp.conf` file. You then stop and restart `slpd` to cause the SLP daemon as to start as a DA. You can assign only one DA per host.

1. **Become superuser.**
2. **Edit the `/etc/inet/slp.conf` file and set the `net.slp.isDA` property to true.**

   ```
   net.slp.isDA=True
   ```

3. **Save the file and exit.**
4. **Restart `slpd` to deploy it as a DA.**

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

Where to Place DAs

This section provides suggestions for where to place DAs in different situations.

1. When multicast routing is not enabled and DAs are required to bridge service discovery between subnets

   In this case, a DA must be placed on a host with interfaces and all subnets that share services. The `net.slp.interfaces` configuration property does *not* need to be set, unless IP packets are not routed among the interfaces. See "Multihoming Configuration" on page 281 for more information on configuring the `net.slp.interfaces` property.
2. When DAs are deployed for scalability, the primary consideration is optimizing agent access

   UAs typically make many requests for services to DAs. An SA registers with the DA once, and can refresh the advertisement at periodic, but not frequent, intervals. As a result, UA access to DAs is far more frequent than SA access. The number of service advertisements is also usually smaller than the number of requests. Consequently, most DA deployments are more efficient if the deployment is optimized for UA access.
3. Situating DAs so that they are topologically close to UAs on the network to optimize UA access

   Placing UAs topologically close to their DAs reduces the amount of routing delay for answering SLP requests. Naturally, you must configure the DA with a scope shared by both the UA and SA clients.

Placing Multiple DAs for Load Balancing

You can deploy multiple DAs for the same collection of scopes as a means of load balancing. Deployment of multiple DAs is suggested in any of the following circumstances:

- UA requests to a DA are timing out, or are returning with the `DA_BUSY_NOW` error.
- The DA log shows that many SLP requests are being dropped.
- The network of users sharing services in the scopes spans a number of buildings or physical sites.

You can do a `snoop` trace of SLP traffic to determine how many UA requests return with the `DA_BUSY_NOW` error. If the number of UA requests returned is high, which is likely if a single DA is deployed for all users, UAs in buildings physically and topologically distant from the DA can exhibit slow response or excessive timeouts. You might want to deploy a DA in each building to improve response for UA clients within the building.

Links connecting buildings are often slower than the local area networks within the buildings. If your network spans multiple buildings or physical sites, set the

net.slp.DAAddresses property in the /etc/inet/slp.conf file to a list of specific host names or addresses so that the UAs access only the DAs you specify.

If a particular DA is using large amounts of host memory for service registrations, reduce the number of SA registrations by reducing the number of scopes the DA supports. You can split a scope having many registrations into two and support one of the new scopes by deploying another DA on another host.

Multihoming

A multihomed server acts as a host on multiple IP subnets. The server can sometimes have more than one network interface card and can act as a router. IP packets, including multicast packets, are routed between the interfaces. In some cases, routing between interfaces is disabled. The following sections describe how to configure SLP for those cases.

Multihoming Configuration

Without configuration, slpd listens for multicast and for UDP/TCP unicast on the default network interface. If unicast and multicast routing is enabled between interfaces on a multihomed machine, no additional configuration is needed, because multicast packets that arrive at another interface are properly routed to the default. As a result, multicast requests for DA or other service advertisements arrive at slpd. If routing is not turned on for some reason, then configuration is required.

When to Configure for Nonrouted, Multiple Network Interfaces

The most likely cases where configuration might be required on multihomed machines are:

- Unicast routing is enabled between the interfaces and multicast routing is disabled.
- Unicast routing and multicast routing are both disabled between the interfaces.

When multicast routing is disabled between interfaces, it is usually because multicast has not been deployed in the network. In that case, broadcast is normally used for non-DA-based service discovery and for DA discovery on the individual subnets. Broadcast is configured by setting the net.slp.isBroadcastOnly property to true.

Tasks for Configuring Nonrouted, Multiple Network Interfaces (Task Map)

TABLE 19–5 Tasks for Administering SLP

Task	Description	Instructions
Configure the `net.slp.interfaces` property	Set this property to enable `slpd` to listen for unicast and multicast/broadcast SLP requests on the specified interfaces .	"Configuring the `net.slp.interfaces` Property" on page 282
Arrange proxy service advertisements so that UAs on subnets get service URLs with reachable addresses.	Restrict proxy advertisement to a machine running `slpd` connected to a single subnet rather than a multihomed host.	"Proxy Advertising on Multihomed Hosts" on page 284
Place the DAs and configure scopes to assure reachability between UAs and SAs.	Configure the `net.slp.interfaces` property on multihomed hosts with a single interface host name or address.	"DA Placement and Scope Name Assignment" on page 284
	Run a DA on a multihomed host, but configure scopes such that SAs and UAs on each subnet use different hosts	

Configuring the `net.slp.interfaces` Property

If the `net.slp.interfaces` property is set, `slpd` listens for unicast and multicast/broadcast SLP requests on the interfaces listed in the property, rather than on the default interface.

Usually, you set the `net.slp.interfaces` property in conjunction with enabling broadcast by setting the `net.slp.isBroadcastOnly` property, because multicast has not been deployed in the network. However, if multicast has been deployed, but is not being routed on this particular multihomed host, it is possible for a multicast request to arrive at `slpd` from more than one interface. This can happen because another multihomed host or router connecting the subnets served by the interfaces routes the packets.

When this happens, the SA server or UA sending the request gets two responses from `slpd` on the multihomed host. These responses are filtered out by the client libraries, so the client should not see them. They are visible on the `snoop` trace, however.

Note – If unicast routing is turned off, services advertised by SA clients on multihomed hosts might not be reachable from all the subnets. If services are unreachable, SA clients can do the following:

- Advertise one service URL for each individual subnet
- Assure that requests from a particular subnet are answered with a reachable URL

The SA client library makes no effort to assure that reachable URLs are advertised. Therefore, it is up to the service program, which might or might not handle a multihomed host with no routing, to assure that reachable URLs are advertised.

Before deploying a service on a multihomed host with unicast routing disabled, you should use `snoop` to determine whether the service handles requests from multiple subnets correctly. Furthermore, if you are planning on deploying a DA on the multihomed host, see "DA Placement and Scope Name Assignment" on page 284.

▼ How to Configure the `net.slp.interfaces` Property

Use the following procedure to change the `net.slp.interfaces` property in the `slp.conf` file.

1. **Become superuser.**
2. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**
4. **Change the `net.slp.interfaces` property in the `slpd.conf` file:**

   ```
   net.slp.interfaces=value
   ```

value	List of IPv4 addresses or host names of the network interface cards on which the DA or SA should listen for multicast, unicast UDP, and TCP messages on port 427.

For example, a server with three network cards and multicast routing turned off is connected to three subnets. The IP addresses of the three network interfaces are: `192.147.142.42`, `192.147.143.42`, and `192.147.144.42`. The subnet mask is `255.255.255.0`. The following property setting causes `slpd` to listen on all three interfaces for unicast and multicast/broadcast messaging:

```
net.slp.interfaces=192.147.142.42,192.147.143.42,192.147.144.42
```

Note – You can specify IP addresses or resolvable host names for the `net.slp.interfaces` property.

---

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

6. **Restart `slpd` to activate your changes. Type the following command:**

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

Proxy Advertising on Multihomed Hosts

If a host with multiple interfaces advertises services using `slpd` and proxy registration, the service URLs advertised by `slpd` must contain reachable host names or addresses. If unicast routing is enabled between the interfaces, then hosts on all subnets are able to reach hosts on the others, and proxy registrations can be made for a service on any subnet. If, however, unicast routing is disabled, then service clients on one subnet cannot reach services on another through the multihomed host (although they might be able to reach them through another router).

For example, suppose the host with default host name `bigguy` has three interface cards on three different unrouted subnets. The host names on these subnets are `bigguy`, with IP address `192.147.142.42`, `bigguy1`, with IP address `192.147.143.42`, and `bigguy2`, with IP address `192.147.144.42`. Now suppose that a legacy printer, `oldprinter`, is connected to the `143` subnet. If the URL:

```
service:printing:lpr://oldprinter/queue1
```

is proxy-advertised on all interfaces by configuring `net.slp.interfaces` to listen on all interfaces, machines on the `142` and `144` subnets will receive the URL in response to service requests, though they will be unable to access the service.

The solution to this problem is to perform the proxy advertisement with `slpd` running on a machine connected the `143` subnet only, rather than on the multihomed host. Only hosts on the `143` subnet are able to obtain the advertisement in response to a service request.

DA Placement and Scope Name Assignment

The placement of DAs and assignment of scope names in a network with a multihomed host, in which routing is disabled but the `net.slp.interfaces` property is configured, must be done carefully to assure that clients obtain accessible services. Again, if unicast routing is enabled between the interfaces on a multihomed

machine, no special DA and scope configuration is necessary, because advertisements cached with the DA identify services that are accessible from any of the subnets. However, if unicast routing is disabled, poor placement of DAs can result in problems.

To see what can go wrong in the previous example, consider what would happen if `bigguy` runs a DA, and clients on all subnets have the same scopes. SAs on the `143` subnet register their service advertisements with the DA. UAs on the `144` subnet can obtain those service advertisements, even though hosts on the `143` subnet are unreachable.

One solution to this problem is to run a DA on each subnet and not on the multihomed host. In this case, the `net.slp.interfaces` property on the multihomed hosts should be configured with a single interface host name or address, or it should be left unconfigured, forcing the default interface to be used. A drawback of this solution is that multihomed hosts are often large machines that could better handle the computational load of a DA.

Another solution is to run a DA on the multihomed host, but configure scopes such that the SAs and UAs on each subnet have a different scope. For example, in the above case, scopes could be configured so that UAs and SAs on the `142` net have scope `scope142`, UAs and SAs on the `143` net have scope `scope143`, and UAs and SAs on the `144` net have scope `scope144`. You can configure the `net.slp.interfaces` property on `bigguy` with the three interfaces, so that the DA serves three scopes on the three subnets.

Considerations When Configuring for Nonrouted, Multiple Network Interfaces

If multicast routing is turned off in the network, but unicast routing between interfaces on a multihomed host is enabled, configuring the `net.slp.interfaces` property enables a DA on the multihomed host to bridge service advertisements between the subnets. Because unicast is routed between the interfaces, hosts on a subnet different from the subnet on which the service is located can contact the service when they receive the service URL. Without the DA, SA servers on a particular subnet receive only broadcasts made on the same subnet, so they cannot locate services off of their subnet.

The most common case requiring configuration of `net.slp.interfaces` is when multicast is not deployed in the network and broadcast is used instead. Other cases require careful thought and planning to avoid unnecessary duplicate responses or unreachable services.

CHAPTER 20

Incorporating Legacy Services

Legacy services are network services that predate the development and implementation of SLP. Solaris services such as the line printer daemon (`lpsched`), the NFS file service, and NIS/NIS+ name service, for example, do not contain internal SAs for SLP. This chapter describes when and how to advertise legacy services.

When to Advertise Legacy Services

With legacy service advertising, you can enable the SLP UAs to find devices and services such as the following on your network:

- Hardware devices without SLP drivers.
- Hardware devices and software services that do that do not contain SLP SAs. When applications with SLP UAs need to find printers or databases that do not contain SLP SAs, for example, legacy advertising might required.

Advertising Legacy Services

You can take any of the following actions to advertise legacy services.

- Modify the service to incorporate an SLP SA.

- Write a small program that functions as an SLP SA to advertise the service.
- Use proxy advertising to have `slpd` advertise the service.

Modifying the Service

If the source code for the software server is available, you to incorporate a SLP SA. The C and Java APIs for SLP are relatively straightforward to use. See the man pages for information on the C API and documentation on the Java API. If the service is a hardware device, the manufacturer might have an updated PROM that incorporates SLP. Contact the device manufacturer for more information.

Writing an SLP SA to Advertise the Service

If the source code or an updated PROM that contains SLP is not available, you can write a small SLP SA to advertise the service. The SA then functions as a small daemon that you start or stop from the same shell script you use to start and stop the service. The SLP APIs for C and Java provide programmatic access to SLP.

SLP Proxy Registration

Solaris `slpd` supports legacy service advertising with a proxy registration file. The proxy registration file is a list of service advertisements in a portable format.

▼ How to Enable SLP Proxy Registration

1. **Create a proxy registration file on the host file system or in any network directory that is accessible by HTTP.**

2. **Check to see if a service type template exists for the service. The template is a description of the service URL and attributes of a service type. A template is used to define the components of an advertisement for a particular service type. If a service type template exists, use the template to construct the proxy registration. See RFC 2609 for more information on service type templates.**

   a. **If a service type template is not available for the service, select a collection of attributes that precisely describe the service. Use a naming authority other than the default for the advertisement. The default naming authority is allowed only for service types that have been standardized. See RFC 2609 for more information on naming authorities.**

      For example, suppose a company called *BizApp* has a local database used to track software defects. To advertise the database, the company might use a URL with the service type `service:bugdb.bizapp`. The naming authority would then be `bizapp`.

3. **Follow the next steps to configure the `net.slp.serializedRegURL` property in the `/etc/inet/slp.conf` file`net.slp.serializedRegURL` with the location of the registration file created in the previous steps.**

4. **Become superuser.**

5. **Type the following command to stop `slpd` and all SLP activity on the host:**

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

6. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

7. **Specify the location of the proxy registration file in the `net.slp.serializedRegURL` property of the `/etc/inet/slp.conf` file.**

   `net.slp.net.slp.serializedRegURL=`*proxy registration file URL*

   For example, if the serialized registration file is `/net/inet/slp.reg`, you configure the property as shown in the example:

   ```
   net.slp.serializedRegURL=file:/etc/inet/slp.reg
   ```

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

9. **Restart `slpd` to activate your changes. Type the following command:**

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

Using SLP Proxy Registration to Advertise

A service advertisement consists of lines that identify the service URL, an optional scope, and a series of attribute definitions. The SLP daemon reads, registers, and

maintains proxy advertisements exactly as an SA client would. The following is an example of an advertisement from a proxy registration file.

In the example, a legacy printer that supports LPR protocol and an FTP server are advertised. Line numbers have been added for description purposes and are not part of the file.

```
1#Advertise legacy printer.
2
3service:lpr://bizserver/mainspool,en,65535
4scope=eng,corp
5make-model=Laserwriter II
6location-description=B16-2345
7color-supported=monochromatic
8fonts-supported=Courier,Times,Helvetica 9 10
9
10#Advertise FTP server
11
12ftp://archive/usr/src/public,en,65535,src-server
13content=Source code for projects
14
```

Note – The proxy registration file supports the same convention for escaping non-ASCII characters as the configuration file does. For more information about the format of the proxy registration file, see RFC 2614.

TABLE 20–1 SLP Proxy Registration File Description

Line numbers	Description
1 and 10	Comment lines begin with a cross-hatch symbol (#) and do not affect the file's operation. Everything up through the end of a comment line is ignored.
2, 9, and 14	Blank lines that delimit the advertisements

TABLE 20–1 SLP Proxy Registration File Description *(Continued)*

Line numbers	Description
3, 12	Service URLs that each have three required fields and one optional field separated by commas: ■ 1. Generic or `service: URL` advertised. See RFC 2609 for the specification of how to form a service: URL. ■ 2. Language of the advertisement. In the previous example, the field designated English, *en*. Language is an RFC 1766 language tag. ■ 3. Lifetime of the registration, measured in seconds. The lifetime is restricted to an unsigned 16 bit-integer. If the lifetime is less than the maximum, 65535, `slpd` times out the advertisement. If the lifetime is 65535, `slpd` refreshes the advertisement periodically, and the lifetime is considered permanent, until `slpd` exits. ■ (Optional)Service type field (optional) – If used, this field defines the service type. If the service URL is generic, then it is possible to change the service type under which the URL is advertised. In the previous example of a proxy registration file, line 12 contains a generic FTP URL. The optional type field causes the URL to be advertised under the service type name *src-server*. The `service` prefix is not added by default to the type name.
4	Scope designation Optional line consists of the token `scope` followed by an equal sign and a comma-separated list of scope names. Scope names are defined by the `net.slp.useScopes` configuration property. Only scopes configured for the host should be included list. When a scope line is not added, the registration is made in all scopes with which `slpd` is configured. The scope line must appear immediately after the URL line. Otherwise, scope names are recognized as attributes.
5–8	Attribute definitions After the optional scope line, the bulk of the service advertisement contains attribute/value list pair lines. Each pair consists of the attribute tag, followed by an equal sign, and an attribute value or a comma-separated list of values. In the previous example of a proxy registration file, line 8 illustrates an attribute list with multiple values. All other lists have single values. The format for the attribute names and values is the same as on-the-wire SLP messages.

Considerations When Advertising Legacy Services

Generally, modifying the source code to add SLP is preferable to either writing a standalone SA or using proxy registration. By modifying the source code, it is possible to add service-specific features and to closely track service availability. If the source code is unavailable, writing a standalone SA is preferable to using proxy registration. A standalone SA allows the SA to track service availability by being integrated into the service start/stop procedure. Proxy advertising is generally the third choice, when no source code is available and writing a standalone SA is impractical.

Proxy advertisements are maintained only as long as `slpd` is running to read the proxy registration file. No direct connection exists between the proxy advertisement and the service. If an advertisement times out or `slpd` is halted, the proxy advertisement is no longer available.

If the service is brought down, `slpd` must be stopped. The serialized registration file is edited to comment out or remove the proxy advertisement, and `slpd` is restarted. You must follow the same procedure when the service is restarted or reinstalled. The lack of connection between the proxy advertisement and the service is a major drawback of proxy advertisements.

CHAPTER 21

SLP (Reference)

This chapter describes the SLP status codes and message types. SLP message types are listed with the abbreviations and function codes. SLP status codes are shown descriptions and function codes used to indicate that that the request is received (code 0), or that the receiver is busy.

Note – The SLP daemon (`slpd`) returns status codes for unicast messages only.

SLP Status Codes

TABLE 21–1 SLP Status Codes

Status Type	Status Code	Description
No Error	0	Request was processed without error
LANGUAGE_NOT_SUPPORTED	1	For an AttrRqst or SrvRqst, there is data for the service type in the scope, but not in the language indicated.
PARSE_ERROR	2	The message fails to obey SLP syntax.
INVALID_REGISTRATION	3	The SrvReg has problems—for example, a zero lifetime or an omitted language tag
SCOPE_NOT_SUPPORTED	4	The SLP message did not include a scope in its scope list that is supported by the SA or DA that answered the request.

TABLE 21–1 SLP Status Codes *(Continued)*

Status Type	Status Code	Description
AUTHENTICATION_UNKNOWN	5	The DA or SA received a request for an unsupported SLP SPI.
AUTHENTICATION_ABSENT	6	The UA or DA expected URL and attribute authentication in the SrvReg and did not receive it.
AUTHENTICATION_FAILED	7	The UA or DA detected an authentication error in an Authentication block.
VER_NOT_SUPPORTED	9	Unsupported version number in message
INTERNAL_ERROR	10	An unknown error occurred in the DA or SA. For example, the operating system ran out of file space.
DA_BUSY_NOW	11	The UA or SA should retry, using exponential back off. The DA is busy processing other messages.
OPTION_NOT_UNDERSTOOD	12	The DA or SA received an unknown option from the mandatory range.
INVALID_UPDATE	13	The DA received a SrvReg without FRESH set, for an unregistered service or with inconsistent service types.
MSG_NOT_SUPPORTED	14	The SA received an AttrRqst or SrvTypeRqst and does not support it.
REFRESH_REJECTED	15	The SA sent a SrvReg or partial SrvDereg to a DA more frequently than the DA's min-refresh-interval.

SLP Message Types

TABLE 21–2 SLP Message Types

Message Type	Abbreviation	Function Code	Description
Service Request	`SrvRqst`	1	Issued by a UA to find services or by a UA or SA server during active DA discovery
Service Reply	`SrvRply`	2	The DA or SA response to a service request
Service Registration	`SrvReg`	3	Enables SAs to register new advertisements, to update existing advertisements with new and changed attributes, and to refresh URL lifetimes
Service Deregistration	`SrvDereg`	4	Used by the SA to deregister its advertisements when the service they represent is no longer available
Acknowledgment	`SrvAck`	5	The DA response to an SA's service request or service deregistration message
Attribute Request	`AttrRqst`	6	Made either by URL or by service type to request a list of attributes
Attribute Reply	`AttrRply`	7	Used to return the list of attributes
DA Advertisement	`DAAdvert`	8	The DA response to multicast service requests
Service Type Request	`SrvTypeRqst`	9	Used to inquire about registered service types that have a particular naming authority and are in a particular set of scopes
Service Type Reply	`SrvTypeRply`	10	The message returned in response to the service type request
SA Advertisement	`SAAdvert`	11	UAs employ the `SAAdvert` to discover SAs and their scopes in networks where no DAs are deployed

CHAPTER 22

Mail Services Topics

Chapter 23	Provides overview information for the mail service
Chapter 24	Provides step-by-step instructions for setting up and troubleshooting the mail service
Chapter 25	Provides background information on the mail service
Chapter 26	Provides information on what's new for the mail service

CHAPTER 23

Mail Services (Overview)

To set up an electronic mail service and maintain it are complex tasks, both of which are critical to the daily operation of your network. As a network administrator, you might need to expand an existing mail service, or perhaps you might need to set up a mail service on a new network or subnet. The chapters on mail services can help you plan and set up a mail service for your network. This chapter provides a list of new features in `sendmail`, as well as a list of other sources of information. The chapter also provides overviews of the software and hardware components that are required to establish a mail service.

Look in Chapter 24 for procedural information on how to set up and administer mail services. For details, refer to "Mail Services Task Map" on page 304.

Look in Chapter 25 for a more detailed description of the components of mail services. This chapter also describes the mail service programs and files, the mail routing process, and the interactions of `sendmail` with name services.

What's New in Version 8.12 of `sendmail`

Version 8.12 of `sendmail` has been included in this Solaris 9 release. Chapter 26 describes all of its new features. The following list highlights some of the important changes to `sendmail`.

Other `sendmail` Information Sources

The following is a list of additional information sources about `sendmail`.

- Home page for `sendmail` – http://www.sendmail.org
- FAQ for `sendmail` – http://www.sendmail.org/faq
- README for new `sendmail` configuration files – http://www.sendmail.org/m4/readme.html
- A guide for issues that are related to migrating to more recent 8.*+Sun versions of `sendmail` – http://www.sendmail.org/vendor/sun/
- `Fatbrain.com` for books about `sendmail`, particularly the second edition of *sendmail* from O'Reilly & Associates, Inc. – http://www1.fatbrain.com/catalogs/computing/subjects.asp?SubjectCode=OML

Introduction to the Components of Mail Services

Many software and hardware components are required to establish a mail service. The following sections give a quick introduction to these components and some of the terms that are used to describe them.

The first section, "Overview of the Software Components" on page 301, defines the terms that are used when discussing the software parts of the mail delivery system. The next section, "Overview of the Hardware Components" on page 301, focuses on the functions of the hardware systems in a mail configuration.

Overview of the Software Components

The following table introduces some of the software components of a mail system. Refer to "Software Components" on page 351 for a complete description of all of the software components.

Component	Description
`.forward` files	Files that you can set up in a user's home directory to redirect mail or to send mail to a program automatically
mailbox	A file on a mail server that is the final destination for email messages
mail addresses	Address that contains the name of the recipient and the system to which a mail message is delivered
mail aliases	An alternate name that is used in a mail address
mail queue	A collection of mail messages that needs to be processed by the mail server
postmaster	A special mail alias that is used to report problems and ask questions about the mail service
sendmail configuration file	A file that contains all the information necessary for mail routing

Overview of the Hardware Components

A mail configuration requires three elements, which you can combine on the same system or provide in separate systems.

- A mail host – A system that is configured to handle email addresses that are difficult to resolve
- At least one mail server – A system that is configured to hold one or more mailboxes
- Mail clients – Systems that access mail from a mail server

If users are to communicate with networks outside your domain, you must also add a fourth element, a mail gateway.

Figure 23–1 shows a typical electronic mail configuration, using the three basic mail elements plus a mail gateway.

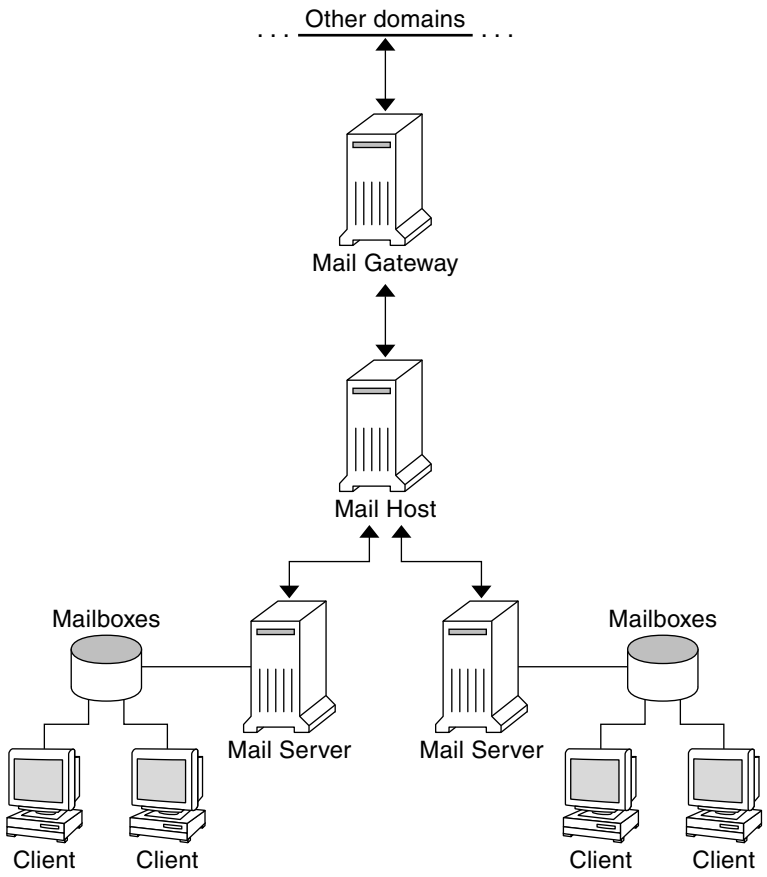


FIGURE 23–1 Typical Electronic Mail Configuration

Each element is described in detail in "Hardware Components" on page 358.

CHAPTER 24

Mail Services (Tasks)

This chapter describes how to set up and administer mail services. If you are not familiar with administering mail services, read Chapter 23 for an introduction to the components of mail services and for a description of a typical mail service configuration (as shown in Figure 23–1). The following list can help you find groups of related procedures that are covered in this chapter.

Look in Chapter 25 for a more detailed description of the components of mail services. This chapter also describes the mail service programs and files, the mail routing process, and the interactions of `sendmail` with name services.

Look in Chapter 26 for a description of the new features that are included in version 8.12 of `sendmail`, the version that is in this Solaris 9 release. You can also read about changes to `mail.local`, `mailstats`, and `makemap`. Chapter 26 also provides a description of a new maintenance command, `editmap`.

Mail Services Task Map

The following table refers you to task maps throughout this chapter that focus on a specific group of procedures.

Task	Description	For Instructions
Setting up mail services	Use these procedures to set up each component of your mail service. Learn how to set up a mail server, a mail client, a mail host, a mail gateway, a virtual host, and how to use DNS with `sendmail`.	"Setting Up Mail Services (Task Map)" on page 308
Building a `sendmail` configuration file	Use this procedure to modify your `sendmail.cf` file. See an example of how to enable domain masquerading.	"Building the `sendmail.cf` Configuration File (Task)" on page 317
Managing mail delivery with an alternate configuration	Use this procedure to prevent mail delivery problems that can occur if the master daemon is disabled.	"Managing Mail Delivery by Using an Alternate Configuration (Task)" on page 319
Administering mail alias files	Use these procedures to provide aliasing on your network. Learn how to manage entries in NIS+ tables. Also, learn how to set up an NIS map, a local mail alias, a keyed map file, and a postmaster alias.	"Administering Mail Alias Files (Task Map)" on page 320
Administering the mail queue	Use these procedures to provide smooth queue processing. Learn how to display and move the mail queue, force mail queue processing, run a subset of the mail queue, and how to run the old mail queue.	"Administering the Queue Directories (Task Map)" on page 332
Administering `.forward` files	Use these procedures to disable `.forward` files or change the search path of the `.forward` file. Also, learn how to permit users to use the `.forward` file by creating and populating `/etc/shells`.	"Administering `.forward` Files (Task Map)" on page 335

Task	Description	For Instructions
Troubleshooting procedures and tips for mail services	Use these procedures and tips to resolve problems with your mail service. Learn how to test the mail configuration, check mail aliases, test the `sendmail` rule sets, verify connections to other systems, and log messages. Find out where to look for other mail diagnostic information.	"Troubleshooting Procedures and Tips for Mail Services (Task Map)" on page 338
Resolving error messages	Use the information in this section to resolve some mail-related error messages.	"Resolving Error Messages" on page 343

Planning Your Mail System

The following list describes some concerns that should be part of your planning process.

- Determine the type of mail configuration that meets your requirements. This section describes two basic types of mail configuration and briefly lists what you need to set up each configuration. If you need to set up a new mail system or if you are expanding an existing one, you might find this section useful. "Local Mail Only" on page 306 describes the first configuration type, and "Local Mail and a Remote Connection" on page 307 describes the second type.
- As necessary, choose the systems that are to act as mail servers, mail hosts, and mail gateways.
- Make a list of all the mail clients for which you are providing service and include the location of their mailboxes. This list can help you when you are ready to create mail aliases for your users.
- Decide how you plan to update aliases and forward mail messages. You might set up an `aliases` mailbox as a place for users to send requests for mail forwarding and for changes to their default mail alias. If your system uses NIS or NIS+, you can administer mail forwarding, rather than requiring users to manage it themselves. "Administering Mail Alias Files (Task Map)" on page 320 provides a list of tasks that are related to aliasing. "Administering `.forward` Files (Task Map)" on page 335 provides a list of tasks that are related to managing `.forward` files.

After you have completed the planning process, you need to set up systems on your site to perform the functions that are described in “Setting Up Mail Services (Task Map)” on page 308. For other task information, refer to “Mail Services Task Map” on page 304.

Local Mail Only

The simplest mail configuration, shown in Figure 24–1, is two or more workstations that are connected to one mail host. Mail is completely local. All the clients store mail on their local disks and act as mail servers. Mail addresses are parsed by using the `/etc/mail/aliases` files.

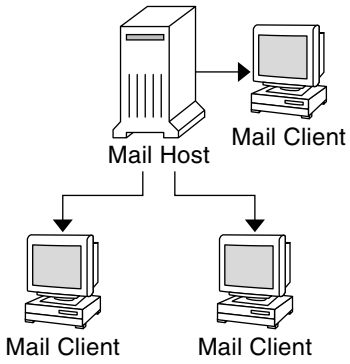


FIGURE 24–1 Local Mail Configuration

To set up this kind of mail configuration, you need the following.

- The default `/etc/mail/sendmail.cf` file on each mail client system (no editing is required).
- A server that is designated as the mail host. You can make this designation by adding `mailhost.`*domain_name* to the `/etc/hosts` file on the mail host. Then, if you are not running NIS or NIS+, add the mail host IP address line to the `/etc/hosts` file of all mail clients.
- Matching `/etc/mail/aliases` files on any system that has a local mailbox (unless you are running NIS or NIS+).
- Enough space in `/var/mail` on each mail client system to hold the mailboxes.

For task information on setting up your mail service, refer to “Setting Up Mail Services (Tasks)” on page 308. If you are looking for a particular procedure that is related to setting up your mail service, refer to “Setting Up Mail Services (Task Map)” on page 308.

Local Mail and a Remote Connection

The most common mail configuration in a small network is shown in Figure 24–2. One system includes the mail server, the mail host, and the mail gateway to the outside world. Mail is distributed by using the `/etc/mail/aliases` files on the mail gateway. No name service is required.

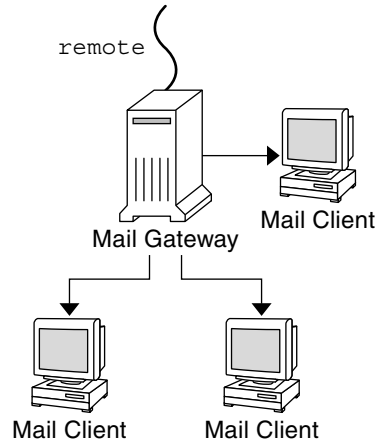


FIGURE 24–2 Local Mail Configuration With a UUCP Connection

In this configuration, you can assume that the mail clients mount their mail files from `/var/mail` on the mail host. To set up this kind of mail configuration, you need the following.

- The `main.cf` file on the mail gateway (no editing is required if Mail Exchanger (MX) records are used).
- The default `/etc/mail/sendmail.cf` file on each mail client system (no editing is required).
- A server that is designated as the mail host. You can make this designation by adding `mailhost.`*domain_name* to the `/etc/hosts` file on the mail host. Then, if you are not running NIS or NIS+, add the IP address line for the mail host to the `/etc/hosts` file of every mail client.
- Matching `/etc/mail/aliases` files on any system that has a local mailbox (unless you are running NIS or NIS+).
- Enough space in `/var/mail` on the mail server to hold the client mailboxes.

For task information on setting up your mail service, refer to "Setting Up Mail Services (Tasks)" on page 308. If you are looking for a particular procedure that is related to setting up your mail service, refer to "Setting Up Mail Services (Task Map)" on page 308.

Setting Up Mail Services (Task Map)

The following table describes the procedures for setting up mail services.

Task	Description	For Instructions
Setting up a mail server	Steps to enable a server to route mail	"How to Set Up a Mail Server" on page 309
Setting up a mail client	Steps to enable a user to receive mail	"How to Set Up a Mail Client" on page 311
Setting up a mail host	Steps to establish a mail host that can resolve email addresses	"How to Set Up a Mail Host" on page 313
Setting up a mail gateway	Steps to manage communication with networks outside your domain	"How to Set Up a Mail Gateway" on page 314
Using DNS with `sendmail`	Steps to enable DNS host lookups	"How to Use DNS With `sendmail`" on page 316
Setting up a virtual host	Steps to assign more than one IP address to a host	"How to Set Up a Virtual Host" on page 316

Setting Up Mail Services (Tasks)

You can readily set up a mail service if your site does not provide connections to email services outside your company or if your company is in a single domain.

Mail requires two types of configurations for local mail. Refer to Figure 24–1 in "Local Mail Only" on page 306 for a representation of these configurations. Mail requires two more configurations for communication with networks outside your domain. Refer to Figure 23–1 in "Overview of the Hardware Components" on page 301 or Figure 24–2 in "Local Mail and a Remote Connection" on page 307 for a representation of these configurations. You can combine these configurations on the same system or provide them on separate systems. For example, if your mail host and mail server functions are on the same system, follow the directions in this section for setting up that system as a mail host. Then, follow the directions in this section for setting up the same system as a mail server.

Note – The following procedures for setting up a mail server and mail client apply when mailboxes are NFS mounted. However, mailboxes typically are maintained in locally mounted `/var/mail` directories, which eliminates the need for the following procedures.

▼ How to Set Up a Mail Server

No special steps are required to set up a mail server that is only serving mail for local users. The user must have an entry in the password file or in the name space, and the user should have a local home directory (for checking the `~/.forward` file) for mail to be delivered. For this reason, home directory servers are often set up as the mail server. "Hardware Components" on page 358 in Chapter 25 provides more information about the mail server.

The mail server can route mail for many mail clients. The only resource requirement for this type of mail server is that it have adequate spooling space for client mailboxes.

Note – Either the `/var/mail` directory should be available for remote mounting or a service such as Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) should be available from the server for clients to access their mailboxes. The following task shows you how to set up a mail server by using the `/var/mail` directory. To provide configuration guidelines for POP or IMAP is beyond the scope of this document.

For the following task, ensure that the `/etc/dfs/dfstab` file shows that the `/var/mail` directory is exported.

1. **Become superuser or assume an equivalent role on the mail server.**
2. **Stop `sendmail`.**

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

3. **See if the `/var/mail` directory is available for remote access.**

   ```
   # share
   ```

   If the `/var/mail` directory is listed, skip to step 5.

   If the `/var/mail` directory is not listed or if no list appears, continue with the appropriate substep.

   a. **(Optional) If no list appears, start NFS services.**

      Follow the procedure, "How to Set Up Automatic File-System Sharing" on page 147, to use the `/var/mail` directory to start NFS services.

b. **(Optional) If the** `/var/mail` **directory is not included in the list, add it to** `/etc/dfs/dfstab`**.**

Add the following command line to the `/etc/dfs/dfstab` file.

```
share -F nfs -o rw /var/mail
```

4. **Make the file system available for mounting.**

Run the following command.

```
shareall
```

5. **Ensure that your name service has been started.**

a. **(Optional) If you are running NIS, use this command.**

```
# ypwhich
```

For more information, refer to the `ypwhich`(1) man page.

b. **(Optional) If you are running NIS+, use this command.**

```
# nisls
```

For more information, refer to the `nisls`(1) man page.

c. **(Optional) If you are running DNS, use this command.**

```
# nslookup hostname
```

hostname	Use your host name.

For more information, refer to the `nslookup`(1M) man page.

d. **(Optional) If you are running LDAP, use this command.**

```
# ldaplist
```

For more information, refer to the `ldaplist`(1) man page.

6. **Restart** `sendmail`**.**

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

Note – The `mail.local` program automatically creates mailboxes in the `/var/mail` directory the first time a message is delivered. You do not need to create individual mailboxes for your mail clients.

▼ How to Set Up a Mail Client

A mail client is a user of mail services with a mailbox on a mail server and a mail alias in the `/etc/mail/aliases` file that points to the location of the mailbox. "Hardware Components" on page 358 in Chapter 25 provides a brief description of a mail client.

Note – You can also perform the task of setting up a mail client by using a service such as Post Office Protocol (POP) or Internet Message Access Protocol (IMAP). However, to provide configuration guidelines for POP or IMAP is beyond the scope of this document.

1. **Become superuser or assume an equivalent role on the mail client's system.**
2. **Stop** `sendmail`**.**

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

3. **Ensure that a** `/var/mail` **mount point exists on the mail client's system.**

   The mount point should have been created during the installation process. You can use `ls` to ensure that the file system exists. The following example shows the response you receive if the file system has not been created.

   ```
   # ls -l /var/mail
   /var/mail not found
   ```

4. **Ensure that no files are in the** `/var/mail` **directory.**

   If mail files do exist in this directory, you should move them so that they are not covered when the `/var/mail` directory is mounted from the server.

5. **Mount the** `/var/mail` **directory from the mail server.**

   You can mount the mail directory automatically or at boot time.

   a. **(Optional) Mount** `/var/mail` **automatically.**

      Add an entry such as the following to the `/etc/auto_direct` file.

      ```
      /var/mail -rw,hard,actimeo=0 server:/var/mail
      ```

server	Use the assigned server name.

b. (Optional) Mount `/var/mail` at boot time.

Add the following entry to the `/etc/vfstab` file. This entry permits the `/var/mail` directory on the mail server that is specified to mount the local `/var/mail` directory.

```
server:/var/mail - /var/mail nfs - no rw,hard,actimeo=0
```

The client's mailbox is automatically mounted any time the system is rebooted. If you are not rebooting the system, type the following command to mount the client mailbox.

```
# mountall
```

Caution – For mailbox locking and mailbox access to work properly, you must include the `actimeo=0` option when mounting mail from an NFS server.

6. Update `/etc/hosts`.

Edit the `/etc/hosts` file and add an entry for the mail server. This step is not required if you are using a name service.

```
# cat /etc/hosts
#
# Internet host table
#
..
IP_address       mailhost   mailhost   mailhost.example.com
```

IP_address	Use the assigned IP addresses.
example.com	Use the assigned domain.
mailhost	Use the assigned mailhost.

For more information, refer to the `hosts`(4) man page.

7. Add an entry for the client to one of the alias files.

Refer to "Administering Mail Alias Files (Task Map)" on page 320 for a task map about administering mail alias files.

Note – The `mail.local` program automatically creates mailboxes in the `/var/mail` directory the first time a message is delivered. You do not need to create individual mailboxes for your mail clients.

8. Restart `sendmail`.

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

▼ How to Set Up a Mail Host

A mail host resolves email addresses and reroutes mail within your domain. A good candidate for a mail host is a system that connects your network to the outside world or to a parent domain. The following procedure shows you how to set up a mail host.

1. **Become superuser or assume an equivalent role on the mail host system.**

2. **Stop** `sendmail`**.**

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

3. **Verify the host-name configuration.**

   Run the `check-hostname` script to verify that `sendmail` can identify the fully qualified host name for this server.

   ```
   % /usr/lib/mail/sh/check-hostname
   hostname phoenix OK: fully qualified as phoenix.example.com
   ```

   If this script is not successful in identifying the fully qualified host name, you need to add the fully qualified host name as the first alias for the host in `/etc/hosts`.

4. **Update the** `/etc/hosts` **file.**

   Choose the step that is appropriate for you.

   a. **(Optional) If you are using NIS or NIS+, edit the** `/etc/hosts` **file on the system that is assigned to be the new mail host.**

      Add the word `mailhost` and `mailhost.`*domain* after the IP address and system name of the mail host system.

      IP_address mailhost `mailhost mailhost.`*domain* `loghost`

IP_address	Use the assigned IP address.
mailhost	Use the system name of the mail host system.
domain	Use the expanded domain name.

      The system is now designated as a mail host. The *domain* should be identical to the string that is given as the subdomain name in the output of the following command.

      ```
      % /usr/lib/sendmail -bt -d0 </dev/null
      Version 8.12.0+Sun
       Compiled with: LDAPMAP MAP_REGEX LOG MATCHGECOS MIME7TO8 MIME8TO7
                      NAMED_BIND NDBM NETINET NETINET6 NETUNIX NEWDB NIS NISPLUS
                      QUEUE SCANF SMTP USERDB XDEBUG

      ============ SYSTEM IDENTITY (after readcf) ============
            (short domain name) $w = phoenix
      ```

```
  (canonical domain name) $j = phoenix.example.com
         (subdomain name) $m = example.com
              (node name) $k = phoenix
========================================================
```

See the following example of how the `hosts` file should look after these changes.

```
# cat /etc/hosts
#
# Internet host table
#
172.31.255.255   localhost
192.168.255.255  phoenix mailhost mailhost.example.com loghost
```

b. (Optional) If you are not using NIS or NIS+, edit the `/etc/hosts` file on each system in the network and create the following entry.

IP_address mailhost `mailhost` *mailhost.domain* `loghost`

5. **Select the correct configuration file to copy and rename.**

   The following command copies and renames the `/etc/mail/main.cf` file.

```
# cp /etc/mail/main.cf /etc/mail/sendmail.cf
```

6. **Restart `sendmail`.**

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

7. **Test your mail configuration.**

   See "How to Test the Mail Configuration" on page 338 for instructions.

   For further information about mail hosts, refer to "Hardware Components" on page 358 in Chapter 25.

▼ How to Set Up a Mail Gateway

A mail gateway manages communication with networks outside your domain. The mailer on the sending mail gateway can match the mailer on the receiving system.

A good candidate for a mail gateway is a system that is attached to Ethernet and phone lines or a system that is configured as a router to the Internet. You can configure the mail host or another system as the mail gateway. You might choose to configure more than one mail gateway for your domain. If you have UNIX-to-UNIX Copy Program (UUCP) connections, you should configure the system (or systems) with UUCP connections as the mail gateway.

1. **Become superuser or assume an equivalent role on the mail gateway.**

2. **Stop** `sendmail`**.**

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

3. **Select the correct configuration file to copy and rename.**

   The following command copies and renames the `main.cf` file.

   ```
   # cp /etc/mail/main.cf /etc/mail/sendmail.cf
   ```

4. **Verify the host-name configuration.**

   Run the `check-hostname` script to verify that `sendmail` can identify the fully qualified host name for this server.

   ```
   # /usr/lib/mail/sh/check-hostname
   hostname phoenix OK: fully qualified as phoenix.example.com
   ```

   If this script is not successful in identifying the fully qualified host name, you need to add the fully qualified host name as the first alias for the host in `/etc/hosts`. If you need help with this step, refer to step 4 of "How to Set Up a Mail Host" on page 313.

5. **Ensure that your name service has been started.**

   a. **(Optional) If you are running NIS, use this command.**

      ```
      # ypwhich
      ```

      For more information, refer to the `ypwhich`(1) man page.

   b. **(Optional) If you are running NIS+, use this command.**

      ```
      # nisls
      ```

      For more information, refer to the `nisls`(1) man page.

   c. **(Optional) If you are running DNS, use this command.**

      ```
      # nslookup hostname
      ```

hostname	Use your host name.

      For more information, refer to the `nslookup`(1M) man page.

   d. **(Optional) If you are running LDAP, use this command.**

      ```
      # ldaplist
      ```

      For more information, refer to the `ldaplist`(1) man page.

6. **Restart** `sendmail`**.**

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

7. **Test your mail configuration**

   See "How to Test the Mail Configuration" on page 338 for instructions.

For more information about the mail gateway, refer to “Hardware Components” on page 358 in Chapter 25.

▼ How to Use DNS With `sendmail`

The DNS name service does not support aliases for individuals. This name service does support aliases for hosts or domains that use Mail Exchanger (MX) records and `cname` records. You can specify host names, domain names, or both name in the DNS database. For more information about `sendmail` and DNS, see “Interactions of `sendmail` With Name Services” on page 379 in Chapter 25, or see the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

1. **Become superuser or assume an equivalent role.**
2. **Enable DNS host lookups (NIS+ only).**

   Edit the `/etc/nsswitch.conf` file and remove the # from the `hosts` definition that includes the `dns` flag. The host entry must include the `dns` flag, as the following example shows, for the DNS host aliases to be used.

   ```
   # grep hosts /etc/nsswitch.conf
   #hosts:     nisplus [NOTFOUND=return] files
   hosts:      nisplus dns [NOTFOUND=return] files
   ```

3. **Check for a `mailhost` and `mailhost.`*domain* entry.**

   Use `nslookup` to ensure that an entry exists for `mailhost` and `mailhost.`*domain* in the DNS database. For more information, refer to the `nslookup`(1M) man page.

▼ How to Set Up a Virtual Host

If you need to assign more than one IP address to a host, see this Web site: http://www.sendmail.org/virtual-hosting.html. This site provides complete instructions on how to use `sendmail` to set up a virtual host. However, in the “Sendmail Configuration” section, do not perform step 3b, as shown in the following.

```
# cd sendmail-VERSION/cf/cf
# ./Build mailserver.cf
# cp mailserver.cf /etc/mail/sendmail.cf
```

Instead, for the Solaris operating environment, perform the following steps.

```
# cd /usr/lib/mail/cf
# /usr/ccs/bin/make mailserver.cf
# cp mailserver.cf /etc/mail/sendmail.cf
```

mailserver	Use the name of the `.cf` file.

"Building the `sendmail.cf` Configuration File (Task)" on page 317 outlines these same three steps as part of the build process.

After you have generated your `/etc/mail/sendmail.cf` file, you can continue with the next steps to create a virtual user table, and so forth.

Building the `sendmail.cf` Configuration File (Task)

"How to Build a New `sendmail.cf` File" on page 317 shows you how to build the configuration file. Although you can still use older versions of `sendmail.cf` files, the best practice is to use the new format.

For more details, you should read from the following resources.

- `/usr/lib/mail/README` provides a complete description of the configuration process.
- http://www.sendmail.org provides online information about `sendmail` configuration.
- "Versions of the Configuration File" on page 350 and "`sendmail` Configuration File" on page 371, in Chapter 25, also provide some guidance.

The following sections in Chapter 26 identify new `m4` configuration features.

- "New and Revised Configuration File Options and Related Topics" on page 389
- "New and Revised `m4` Configuration Macros for `sendmail`" on page 405

▼ How to Build a New `sendmail.cf` File

The following procedure shows you how to build a new configuration file.

Note – `/usr/lib/mail/cf/main-v7sun.mc` is now `/usr/lib/mail/cf/main.mc`.

---

1. **Become superuser or assume an equivalent role.**

2. **Stop** `sendmail`**.**

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

3. **Make a copy of the configuration files that you are changing.**

   ```
   # cd /usr/lib/mail/cf
   # cp main.mc myhost.mc
   ```

myhost	Select a new name for your `.mc` file.

4. **Edit the new configuration files (for example,** *myhost* `.mc`**), as necessary.**

   For example, add the following command line to enable domain masquerading.

   ```
   # cat myhost.mc
   ..
   MASQUERADE_AS('host.domain')
   ```

host.domain	Use the desired host name and domain name.

   In this example, `MASQUERADE_AS` causes mail that is sent to be labeled as coming from *host.domain*, rather than `$j`.

5. **Build the configuration file by using** `m4`**.**

   ```
   # /usr/ccs/bin/make myhost.cf
   ```

6. **Test the new configuration file by using the** `-C` **option to specify the new file.**

   ```
   # /usr/lib/sendmail -C myhost.cf -v testaddr </dev/null
   ```

   This command sends a message to `testaddr` while displaying messages as it runs. Only outgoing mail can be tested without restarting the `sendmail` service on the system. For systems that are not handling mail yet, use the full testing procedure in "How to Test the Mail Configuration" on page 338.

7. **Install the new configuration file after making a copy of the original.**

   ```
   # cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.save
   # cp myhost.cf /etc/mail/sendmail.cf
   ```

8. **Restart the** `sendmail` **service.**

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

Managing Mail Delivery by Using an Alternate Configuration (Task)

To facilitate the transport of inbound and outbound mail, the new default configuration of `sendmail` uses a daemon and a client queue-runner. If you have disabled your daemon, you should perform the following task. For a detailed explanation, refer to "New Configuration File, `submit.cf`" on page 386.

▼ How to Manage Mail Delivery by Using an Alternate Configuration of `sendmail.cf`

In the default configuration of `sendmail`, the client queue-runner must be able to submit mail to the daemon on the local SMTP port. If the daemon is not listening on the SMTP port, the mail remains in the queue. To avoid this problem, perform the following task. For more information about the daemon and client queue-runner and to understand why you might have to use this alternate configuration, refer to "New Configuration File, `submit.cf`" on page 386.

This procedure ensures that your daemon runs only to accept connections from the local host.

1. **Become superuser or assume an equivalent role.**
2. **Stop `sendmail`.**

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

3. **Make a copy of the configuration file (either `subsidiary.mc` or `main.mc`, depending on your requirements) that you are changing. In this example, the `subsidiary.mc` file is used.**

   ```
   # cd /usr/lib/mail/cf
   # cp subsidiary.mc myhost.mc
   ```

myhost	Select a new name for your `.mc` file.

4. **Edit the new configuration file (for example, *myhost*`.mc`).**

   Add the following lines before the `MAILER()` lines.

   ```
   # cat myhost.mc
   ..
   DAEMON_OPTIONS('NAME=NoMTA4, Family=inet, Addr=127.0.0.1')dnl
   ```

```
DAEMON_OPTIONS('NAME=NoMTA6, Family=inet6, Addr=::1')dnl
```

5. **Build the configuration file by using** `m4`**.**

```
# /usr/ccs/bin/make myhost.cf
```

6. **Install the new configuration file after making a copy of the original.**

```
# cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.save
# cp myhost.cf /etc/mail/sendmail.cf
```

7. **Restart the** `sendmail` **service.**

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

Administering Mail Alias Files (Task Map)

The following table describes the procedures for administering mail alias files. For more information on this topic, refer to "Mail Alias Files" on page 373 in Chapter 25.

Task	Description	For Instructions
Managing alias entries in an NIS+ `mail_aliases` table	If your name service is NIS+, use these procedures to manage the contents of your `mail_aliases` table. Learn how to list, add, edit, and delete entries.	"How to Manage Alias Entries in an NIS+ `mail_aliases` Table" on page 321
Setting up an NIS `mail.aliases` map	If your name service is NIS, follow these instructions to facilitate aliasing with a `mail.aliases` map.	"How to Set Up an NIS `mail.aliases` Map" on page 326
Setting up a local mail alias file	If you are not using a name service (such as NIS or NIS+), follow these instructions to facilitate aliasing with the `/etc/mail/aliases` file.	"How to Set Up a Local Mail Alias File" on page 327
Creating a keyed map file	Use these steps to facilitate aliasing with a keyed map file.	"How to Create a Keyed Map File" on page 328
Setting up the `postmaster` alias	Use the procedures in this section to manage the `postmaster` alias. You must have this alias.	"Managing the `Postmaster` Alias" on page 329

Administering Mail Alias Files (Tasks)

Mail aliases must be unique within the domain. This section provides the procedures for administering mail alias files. Alternately, you can use the AdminTool's Database Manager application to perform these tasks on the aliases database.

In addition, you can create database files for the local mail host by using `makemap`. Refer to the `makemap`(1M) man page. The use of these database files does not provide all of the advantages of using a name service like NIS or NIS+. However, you should be able to retrieve the data from these local database files faster because no network lookups are involved. For more information, refer to "Interactions of `sendmail` With Name Services" on page 379 and "Mail Alias Files" on page 373 in Chapter 25.

▼ How to Manage Alias Entries in an NIS+ `mail_aliases` Table

Since NIS+ may be going away, should this section be changed to procedures for NIS or LDAP? If so, can you help me develop the new procedures? JB doesn't know.

To manage entries in an NIS+ table, you can use the `aliasadm` command. To list, add, modify, or delete table entries with the `aliasadm` command, you begin a particular task with the following steps.

1. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**

2. **Complete your task by following the instructions from the example that meets your requirements.**

   - "Example—Initiating an NIS+ `mail_aliases` Table" on page 322
   - "Example—Listing the Entire Contents of the NIS+ `mail_aliases` Table" on page 322
   - "Example—Listing an Individual Entry From the NIS+ `mail_aliases` Table" on page 322
   - "Example—Listing Partial Matches From the NIS+ `mail_aliases` Table" on page 323
   - "Example—Adding Aliases to the NIS+ `mail_aliases` Table From the Command Line" on page 323
   - "Example—Adding Entries by Editing an NIS+ `mail_aliases` Table" on page 324
   - "Example—Editing Entries in an NIS+ `mail_aliases` Table" on page 325
   - "Example—Deleting Entries From an NIS+ `mail_aliases` Table" on page 326

In some instances, you should begin the task by compiling a list of each of your mail clients, the locations of their mailboxes, and the names of the mail server systems.

▼ Example—Initiating an NIS+ `mail_aliases` Table

To create a table, follow these instructions.

1. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**
2. **Initiate an NIS+ table.**

   ```
   # aliasadm -I
   ```

3. **Add entries to the table.**
   - To add two or three aliases, refer to "Example—Adding Aliases to the NIS+ `mail_aliases` Table From the Command Line" on page 323.
   - To add more than two or three aliases, refer to "Example—Adding Entries by Editing an NIS+ `mail_aliases` Table" on page 324.

   For more information, refer to the `aliasadm`(1M) man page.

▼ Example—Listing the Entire Contents of the NIS+ `mail_aliases` Table

To see a complete list of the contents of the table, follow these instructions.

1. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**
2. **List all of the entries in alphabetical order by alias.**

   ```
   # aliasadm -1
   ```

   For more information, refer to the `aliasadm`(1M) man page.

▼ Example—Listing an Individual Entry From the NIS+ `mail_aliases` Table

To see an individual entry from the table, follow these instructions.

1. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**

2. **List an individual entry.**

```
# aliasadm -m ignatz
ignatz: ignatz@saturn # Alias for Iggy Ignatz
```

The input, **aliasadm -m ignatz**, matches only the complete alias name, not partial strings. You cannot use metacharacters (such as * and ?) with `aliasadm -m`.

For more information, refer to the `aliasadm`(1M) man page.

▼ Example—Listing Partial Matches From the NIS+ `mail_aliases` Table

To see partial matches from the table, follow these instructions.

1. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**
2. **List partial matches from the table.**

```
# aliasadm -l | grep partial_string
```

partial_string	Use the string of your choice for your search.

For more information, refer to the `aliasadm`(1M) man page.

▼ Example—Adding Aliases to the NIS+ `mail_aliases` Table From the Command Line

To add two or three aliases to the table, follow these instructions.

1. **Compile a list of each of your mail clients, the locations of their mailboxes, and the names of the mail server systems.**
2. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**
3. **(Optional) If necessary, initiate an NIS+ table.**

   If you are creating a completely new NIS+ `mail_aliases` table, you must first initiate the table. To complete this task, refer to "Example—Initiating an NIS+ `mail_aliases` Table" on page 322.
4. **Add aliases to the table.**

   See this example of a typical entry.

```
# aliasadm -a iggy iggy.ignatz@saturn "Iggy Ignatz"
```

The following list describes the input from the preceding example.

`-a`	The option for adding an alias
`iggy`	The short form of the alias name
`iggy.ignatz@saturn`	The expanded alias name
`"Iggy Ignatz"`	The name for the alias in quotation marks

5. **Display the entry you created and ensure that it is correct.**

   # **aliasadm -m** *alias*

alias	The entry that you created

For more information, refer to the `aliasadm`(1M) man page.

▼ Example—Adding Entries by Editing an NIS+ `mail_aliases` Table

To add more than two or three aliases to the table, follow these instructions.

1. **Compile a list of each of your mail clients, the locations of their mailboxes, and the names of the mail server systems.**
2. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**
3. **Display and edit the aliases table.**

   # **aliasadm -e**

   This command displays the table and enables you to edit it. The editor you use has been set with the `$EDITOR` environment variable. If this variable is not set, `vi` is the default editor.
4. **Use the following format to type each alias on a separate line.**

   alias`:` *expanded_alias* `# ["`*option*`" # "`*comments*`"]`

alias	This column is for the short form of the alias name.
expanded_alias	This column is for the expanded alias name.
option	This column is reserved for future use.

comments	This column is used for comments about the individual alias, such as a name for the alias.

If you leave the option column blank, type an empty pair of quotation marks (`""`) and add the comments.

The order of the entries is not important to the NIS+ `mail_aliases` table. The `aliasadm -l` command sorts the list and displays the entries in alphabetical order.

For more information, refer to "Mail Alias Files" on page 373 and the `aliasadm`(1M) man page.

▼ Example—Editing Entries in an NIS+ `mail_aliases` Table

To edit entries in the table, follow these instructions.

1. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**

2. **Display the alias entry.**

   # **aliasadm -m** *alias*

alias	Use the assigned alias name.

3. **Edit the alias entry, as necessary.**

   # **aliasadm -c** *alias expanded_alias* **[***options comments***]**

alias	If necessary, edit the alias name.
expanded_alias	If necessary, edit the expanded alias name.
options	If necessary, edit the option.
comments	If necessary, edit the comment for this entry.

   For more information, refer to the `aliasadm`(1M) man page, as well as "Mail Alias Files" on page 373.

4. **Display the entry that you have edited and ensure that the entry is correct.**

   # **aliasadm -m** *alias*

   For more information, refer to the `aliasadm`(1M) man page.

▼ Example—Deleting Entries From an NIS+ `mail_aliases` Table

To delete entries from the table, follow these instructions.

1. **Either be a member of the NIS+ group that owns the table, or become `root` on the mail server, or assume an equivalent role.**
2. **Delete an entry from the table.**

```
# aliasadm -d alias
```

alias	Use the alias name for the entry that you are deleting.

For more information, refer to the `aliasadm`(1M) man page.

▼ How to Set Up an NIS `mail.aliases` Map

Use the following procedure to facilitate aliasing with an NIS `mail.aliases` map.

1. **Compile a list of each of your mail clients, the locations of their mailboxes, and the names of the mail server systems.**
2. **Become `root` on the NIS master server or assume an equivalent role.**
3. **Edit the `/etc/mail/aliases` file, and make the following entries.**

   a. **Add an entry for each mail client.**

```
# cat /etc/mail/aliases
..
alias:expanded_alias
```

alias	Use the short alias name.
expanded_alias	Use the expanded alias name (`user@host.domain.com`)

   b. **Ensure that you have a `Postmaster: root` entry.**

```
# cat /etc/mail/aliases
..
Postmaster: root
```

c. **Add an alias for `root`. Use the mail address of the person who is designated as the postmaster.**

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com	Use the assigned address of the designated postmaster.

4. **Ensure that the NIS master server is running a name service to resolve the host names on each mail server.**
5. **Change to the `/var/yp` directory.**

```
# cd /var/yp
```

6. **Apply the `make` command.**

```
# make
```

The changes in the `/etc/hosts` and `/etc/mail/aliases` files are propagated to NIS slave systems and are active in only a few minutes, at most.

▼ How to Set Up a Local Mail Alias File

Use the following procedure to resolve aliases with a local mail alias file.

1. **Compile a list of each of your users and the locations of their mailboxes.**
2. **Become `root` on the mail server or assume an equivalent role.**
3. **Edit the `/etc/mail/aliases` file and make the following entries.**

a. **Add an entry for each user.**

user1`:` *user2@host.domain*

user1	Use the new alias name.
user2@host.domain	Use the actual address for the new alias.

b. **Ensure that you have a `Postmaster: root` entry.**

```
# cat /etc/mail/aliases
..
Postmaster: root
```

c. **Add an alias for `root`. Use the mail address of the person who is designated as the postmaster.**

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com	Use the assigned address of the designated postmaster.

4. **Rebuild the alias database.**

```
# newaliases
```

Depending on the configuration of the `AliasFile` option in `/etc/mail/sendmail.cf`, this command generates in binary form either the single file, `/etc/mail/aliases.db`, or the pair of files, `/etc/mail/aliases.dir` and `/etc/mail/aliases.pag`.

5. **Perform one of the following steps to copy the file or files that were generated.**

a. **(Optional) Copy the `/etc/mail/aliases`, the `/etc/mail/aliases.dir`, and the`/etc/mail/aliases.pag` files to each of the other systems.**

You can copy the three files by using the `rcp` or `rdist` commands. Refer to the `rcp`(1) man page or the `rdist`(1) man page for more information. Alternately, you can create a script for this purpose.

When you copy these files, you do not need to run the `newaliases` command on each of the other systems. However, you should remember that you must update all the `/etc/mail/aliases` files each time you add or remove a mail client.

b. **(Optional) Copy the `/etc/mail/aliases.db` file to each of the other systems.**

You can copy the file by using the `rcp` or `rdist` commands. Refer to the `rcp`(1) man page or the `rdist`(1) man page for more information. Alternately, you can create a script for this purpose.

When you copy this file, you do not need to run the `newaliases` command on each of the other systems. However, you should remember that you must update all the `/etc/mail/aliases` files each time you add or remove a mail client.

▼ How to Create a Keyed Map File

To create a keyed map file, follow these instructions.

1. **Become superuser or assume an equivalent role on the mail server.**

2. **Create an input file.**

   Entries can have the following syntax.

   old_name@newdomain.com *new_name@newdomain.com*
   old_name@olddomain.com `error:nouser No such user here`
   @olddomain.com `%1@`*newdomain.com*

old_name@newdomain.com	Use the user name that was previously assigned with the domain that is newly assigned.
new_name@newdomain.com	Use the address that is newly assigned.
old_name@olddomain.com	Use the user name that was previously assigned with the domain that was previously assigned.
olddomain.com	Use the domain that was previously assigned.
newdomain.com	Use the domain that is newly assigned.

   The first entry redirects mail to a new alias. The next entry creates a message when an incorrect alias is used. The last entry redirects all incoming mail from `olddomain` to `newdomain`.

3. **Create the database file.**

   # **/usr/sbin/makemap** *maptype newmap* **<** *newmap*

maptype	Select a database type, such as `dbm`, `btree`, or `hash`.
newmap	Use the name of the input file and the first part of the name of the database file. If the `dbm` database type is selected, then the database files are created by using a `.pag` and a `.dir` suffix. For the other two database types, the file name is followed by `.db`.

Managing the `Postmaster` Alias

Every system must be able to send mail to a `postmaster` mailbox. You can create an NIS or NIS+ alias for `postmaster`, or you can create the alias in each local `/etc/mail/aliases` file. Refer to these procedures.

- "How to Create a `postmaster` Alias in Each Local `/etc/mail/aliases` File" on page 330
- "How to Create a Separate Mailbox for `postmaster`" on page 330

- "How to Add the `postmaster` Mailbox to the Aliases in the `/etc/mail/aliases` File" on page 331

▼ How to Create a `postmaster` Alias in Each Local `/etc/mail/aliases` File

If you are creating the `postmaster` alias in each local `/etc/mail/aliases` file, follow these instructions.

1. **Become superuser or assume an equivalent role on each local system.**
2. **View the `/etc/mail/aliases` entry.**

```
# cat /etc/mail/aliases
# Following alias is required by the mail protocol, RFC 2821
# Set it to the address of a HUMAN who deals with this system's
# mail problems.
Postmaster: root
```

3. **Edit each system's `/etc/mail/aliases` file.**

   Change `root` to the mail address of the person who is designated as the postmaster.

   `Postmaster:` *mail_address*

mail_address	Use the assigned address for the person who is designated as the postmaster.

4. **(Optional) Create a separate mailbox for the postmaster.**

   You can create a separate mailbox for the postmaster to keep postmaster mail separate from personal mail. If you create a separate mailbox, use the mailbox address instead of the postmaster's personal mail address when you edit the `/etc/mail/aliases` files. For details, refer to "How to Create a Separate Mailbox for `postmaster`" on page 330.

▼ How to Create a Separate Mailbox for `postmaster`

If you are creating a separate mailbox for `postmaster`, follow these instructions.

1. **Become `root` or assume an equivalent role on the mail server.**
2. **Create a user account for the person who is designated as `postmaster` and put an asterisk (`*`) in the password field.**

   For details about adding a user account, refer to "Setting Up User Accounts and Groups (Tasks)," in the *Solaris System Administration Guide: Basic Administration*.

3. **After mail has been delivered, enable the `mail` program to read and write to the mailbox name.**

```
# mail -f postmaster
```

postmaster	Use the assigned address.

▼ How to Add the `postmaster` Mailbox to the Aliases in the `/etc/mail/aliases` File

If you are adding a `postmaster` mailbox to the aliases in the `/etc/mail/aliases` file, follow these instructions.

1. **Become `root` or assume an equivalent role on each system.**
2. **Add an alias for `root`. Use the mail address of the person who is designated as the postmaster.**

```
# cat /etc/mail/aliases
..
root: user@host.domain.com
```

user@host.domain.com	Use the assigned address of the person who is designated as postmaster.

3. **On the postmaster's local system, create an entry in the `/etc/mail/aliases` file that defines the name of the alias (`sysadmin`, for example) and include the path to the local mailbox.**

```
# cat /etc/mail/aliases
..
sysadmin: /usr/somewhere/somefile
```

sysadmin	Create a name for a new alias.
/usr/somewhere/somefile	Use the path to the local mailbox.

4. **Rebuild the alias database.**

```
# newaliases
```

Administering the Queue Directories (Task Map)

The following table describes the procedures for administering the mail queue.

Task	Description	For Instructions
Displaying the contents of the mail queue, `/var/spool/mqueue`	Use this procedure to see how many messages are in the queue and how fast the messages are being cleared from the queue.	"How to Display the Contents of the Mail Queue, `/var/spool/mqueue`" on page 333
Forcing mail queue processing for the mail queue, `/var/spool/mqueue`	Use this procedure to process messages to a system that previously was unable to receive messages.	"How to Force Mail Queue Processing in the Mail Queue, `/var/spool/mqueue`" on page 333
Running a subset of the mail queue, `/var/spool/mqueue`	Use this procedure to force a substring of an address, such as a host name, to be processed or to force a particular message out of the queue.	"How to Run a Subset of the Mail Queue, `/var/spool/mqueue`" on page 333
Moving the mail queue, `/var/spool/mqueue`	Use this procedure to move the mail queue.	"How to Move the Mail Queue, `/var/spool/mqueue`" on page 334
Running the old mail queue, `/var/spool/omqueue`	Use this procedure to run an old mail queue.	"How to Run the Old Mail Queue, `/var/spool/omqueue`" on page 335

Administering the Queue Directories (Tasks)

This section describes some helpful tasks for queue administration. For information about the client-only queue, refer to "New Configuration File, `submit.cf`" on page 386. For other related information, you can refer to "New Queue Features" on page 411.

▼ How to Display the Contents of the Mail Queue, /var/spool/mqueue

Use this procedure to see how many messages are in the queue and how fast they are being cleared from the queue.

- **Use the following command to display this information.**
  - The queue IDs
  - The size of the message
  - The date the message entered the queue
  - The message status
  - The sender and the recipients

```
# /usr/bin/mailq | more
```

This command now checks for the authorization attribute, `solaris.admin.mail.mailq`. If the check is successful, the equivalent of specifying the `-bp` flag with `sendmail` is executed. If the check fails, an error message is printed. By default, this authorization attribute is enabled for all users. The authorization attribute can be disabled by modifying the user entry in `prof_attr`. For more information, refer to the man pages for `prof_attr`(4) and `mailq`(1).

▼ How to Force Mail Queue Processing in the Mail Queue, /var/spool/mqueue

Use this procedure, for example, to process messages to a system that was previously unable to receive messages.

1. **Become `root` or assume an equivalent role.**
2. **Force queue processing and display the progress of the jobs as the queue is cleared.**

```
# /usr/lib/sendmail -q -v
```

▼ How to Run a Subset of the Mail Queue, /var/spool/mqueue

Use this procedure, for example, to force a substring of an address, such as a host name, to be processed or to force a particular message from the queue.

1. **Become `root` or assume an equivalent role.**

2. **Run a subset of the mail queue at any time with** `-qR`*string*.

   ```
   # /usr/lib/sendmail -qRstring
   ```

string	Use a recipient's alias or a substring (like a host name) of *user@host.domain*.

   Alternately, you can run a subset of the mail queue with `-qI`*nnnnn*.

   ```
   # /usr/lib/sendmail -qInnnnn
   ```

nnnnn	Use a queue ID.

▼ How to Move the Mail Queue, `/var/spool/mqueue`

If you are moving the mail queue, follow these instructions.

1. **Become** `root` **on the mail host or assume an equivalent role.**
2. **Kill the** `sendmail` **daemon.**

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

   Now `sendmail` is no longer processing the queue directory.
3. **Change to the** `/var/spool` **directory.**

   ```
   # cd /var/spool
   ```
4. **Move the directory,** `mqueue`**, and all its contents to the** `omqueue` **directory. Then create a new empty directory named** `mqueue`**.**

   ```
   # mv mqueue omqueue; mkdir mqueue
   ```
5. **Set the permissions of the directory to read/write/execute by owner, and read/execute by group. Also, set the owner and group to** `daemon`**.**

   ```
   # chmod 750 mqueue; chown root:bin mqueue
   ```
6. **Start** `sendmail`**.**

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

▼ How to Run the Old Mail Queue, /var/spool/omqueue

To run an old mail queue, follow these instructions.

1. **Become `root` or assume an equivalent role.**
2. **Run the old mail queue.**

   ```
   # /usr/lib/sendmail -oQ/var/spool/omqueue -q
   ```

   The `-oQ` flag specifies an alternate queue directory and the `-q` flag says to run every job in the queue. Use the `-v` flag if you are displaying the verbose output on the screen.
3. **Remove the empty directory.**

   ```
   # rmdir /var/spool/omqueue
   ```

Administering `.forward` Files (Task Map)

The following table describes the procedures for administering `.forward` files. For more information, refer to "`.forward` Files" on page 376 in Chapter 25.

Task	Description	For Instructions
Disabling `.forward` files	Use this procedure if, for example, you want to prevent automated forwarding.	"How to Disable `.forward` Files" on page 336
Changing the `.forward` file search path	Use this procedure if, for example, you want to move all `.forward` files into a common directory.	"How to Change the `.forward` File Search Path" on page 336
Creating and populating `/etc/shells`	Use this procedure to enable users to use the `.forward` file to forward mail to a program or to a file.	"How to Create and Populate `/etc/shells`" on page 337

Administering .forward Files (Tasks)

This section contains several procedures that are related to .forward file administration. Because these files can be edited by users, they can cause problems. For more information, refer to ". forward Files" on page 376 in Chapter 25.

▼ How to Disable .forward Files

This procedure, which prevents automated forwarding, disables the .forward file for a particular host.

1. **Become root or assume an equivalent role.**
2. **Make a copy of /usr/lib/mail/domain/solaris-generic.m4 or your site-specific domain m4 file.**

```
# cd /usr/lib/mail/domain
# cp solaris-generic.m4 mydomain.m4
```

mydomain	Use the file name of your choice.

3. **Add the following line to the file you just created.**

```
define('confFORWARD_PATH','')dnl
```

If a value for confFORWARD_PATH already exists in the m4 file, replace it with this null value.

4. **Build and install a new configuration file.**

If you need help with this step, refer to "How to Build a New sendmail.cf File" on page 317.

▼ How to Change the .forward File Search Path

If, for example, you want to put all .forward files in a common directory, follow these instructions.

1. **Become root or assume an equivalent role.**

2. **Make a copy of** `/usr/lib/mail/domain/solaris-generic.m4` **or your site-specific domain m4 file.**

```
# cd /usr/lib/mail/domain
# cp solaris-generic.m4 mydomain.m4
```

mydomain	Use the file name of your choice.

3. **Add the following line to the file that you just created.**

```
define('confFORWARD_PATH','$z/.forward:/var/forward/$u')dnl
```

If a value for `confFORWARD_PATH` already exists in the `m4` file, replace it with this new value.

4. **Build and install a new configuration file.**

If you need help with this step, refer to "How to Build a New `sendmail.cf` File" on page 317.

▼ How to Create and Populate `/etc/shells`

This file is not included in the standard release, so you must add it if users are to be allowed to use `.forward` files to forward mail to a program or to a file. You can create the file manually by using `grep` to identify all of the shells that are listed in your password file. You can then type the shells into the file. However, it is easier to use the following procedure, which employs a script that can be downloaded.

1. **Download the script.**

http://www.sendmail.org/vendor/sun/gen-etc-shells.html

2. **Become** `root` **or assume an equivalent role.**

3. **To generate a list of shells, run the** `gen-etc-shells` **script.**

```
# ./gen-etc-shells.sh > /tmp/shells
```

This script uses the `getent` command to collect the names of shells that are included in the password file sources that are listed in `/etc/nsswitch.conf`.

4. **Inspect and edit the list of shells in** `/tmp/shells`**.**

With the editor of your choice, remove any shells that you are not including.

5. **Move the file to** `/etc/shells`**.**

```
# mv /tmp/shells /etc/shells
```

Troubleshooting Procedures and Tips for Mail Services (Task Map)

The following table describes troubleshooting procedures and tips for mail services.

Task	Description	For Instructions
Testing mail configuration	Steps for testing changes to the `sendmail` configuration file	"How to Test the Mail Configuration" on page 338
Checking mail aliases	A step to confirm that mail can or cannot be delivered to a specified recipient	"How to Check Mail Aliases" on page 339
Testing the rule sets	Steps for checking the input and returns of the `sendmail` rule sets	"How to Test the `sendmail` Rule Sets" on page 340
Verifying connections to other systems	Tips for verifying connections to other systems	"How to Verify Connections to Other Systems" on page 341
Logging messages by using the `syslogd` program	Tips for gathering error message information	"How to Log Messages" on page 341
Checking other sources for diagnostic information	Tips for getting diagnostic information from other sources	"Other Sources for Mail Diagnostic Information" on page 342

Troubleshooting Procedures and Tips for Mail Services (Tasks)

This section provides some procedures and tips that you can use for troubleshooting problems with mail services.

▼ How to Test the Mail Configuration

To test the changes you make to your configuration file, follow these instructions.

1. **Restart `sendmail` on any system that has a revised configuration file.**

```
# pkill -HUP sendmail
```

2. **Send test messages from each system.**

   ```
   # /usr/lib/sendmail -v names </dev/null
   ```

names	Specify a recipient's email address.

   This command sends a null message to the specified recipient and displays the message activity on your monitor.

3. **Send mail to yourself or other people on the local system by addressing the message to a regular user name.**

4. **(Optional) If you are on Ethernet, send mail in three directions to someone on another system.**

   - From the main system to a client system
   - From a client system to the main system
   - From a client system to another client system

5. **(Optional) If you have a mail gateway, send mail to another domain from the mail host to ensure that the relay mailer and host are configured properly.**

6. **(Optional) If you have set up a UUCP connection on your phone line to another host, send mail to someone at that host and have that person send mail back or call you when the message is received.**

7. **Ask someone to send mail to you over the UUCP connection.**

   The `sendmail` program cannot detect whether the message is delivered because it passes the message to UUCP for delivery.

8. **From different systems, send a message to `postmaster` and ensure that it comes to your postmaster's mailbox.**

▼ How to Check Mail Aliases

To verify aliases and to confirm that mail can or cannot be delivered to a specified recipient, follow these instructions.

- **Display the aliases and identify that the final address is deliverable or not deliverable.**

  ```
  % /usr/lib/sendmail -v -bv recipient
  ```

recipient	Specify a recipient's alias.

The following is an example of the output.

```
% /usr/lib/sendmail -v -bv sandy
sandy... aliased to ssmith
ssmith... aliased to sandy@phoenix
sandy@phoenix... deliverable: mailer esmtp, host phoenix, user sandy@phoenix.example.com
%
```

Avoid loops and inconsistent databases when both local and domain-wide aliases are used. Be especially careful when you move a user from one system to another to avoid the creation of alias loops.

▼ How to Test the `sendmail` Rule Sets

To check the input and returns of the `sendmail` rule sets, follow these instructions.

1. **Change to address test mode.**

   ```
   # /usr/lib/sendmail -bt
   ```

2. **Test a mail address.**

   Provide the following numbers and address at the last prompt (>).

   ```
   > 3,0 mail_address
   ```

mail_address	Use the mail address that you are testing.

3. **End the session.**

   Press Control-d.

   The following is an example of the output.

```
% /usr/lib/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 sandy@phoenix
canonify           input: sandy @ phoenix
Canonify2          input: sandy < @ phoenix >
Canonify2        returns: sandy < @ phoenix . example . com . >
canonify         returns: sandy < @ phoenix . example . com . >
parse              input: sandy < @ phoenix . example . com . >
Parse0             input: sandy < @ phoenix . example . com . >
Parse0           returns: sandy < @ phoenix . example . com . >
ParseLocal         input: sandy < @ phoenix . example . com . >
ParseLocal       returns: sandy < @ phoenix . example . com . >
Parse1             input: sandy < @ phoenix . example . com . >
MailerToTriple     input: < mailhost . phoenix . example . com >
     sandy < @ phoenix . example . com . >
MailerToTriple   returns: $# relay $@ mailhost . phoenix . example . com
```

```
     $: sandy < @ phoenix . example . com . >
Parse1           returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
parse            returns: $# relay $@ mailhost . phoenix . example . com
     $: sandy < @ phoenix . example . com . >
```

▼ How to Verify Connections to Other Systems

The `mconnect` program opens a connection to a mail server on a host that you specify and enables you to test that connection. The program runs interactively, so you can issue various diagnostic commands. See the `mconnect`(1) man page for a complete description. The following example verifies that mail to the user name `sandy` is deliverable.

```
% mconnect phoenix

connecting to host phoenix (172.31.255.255), port 25
connection open
220 phoenix.example.com ESMTP Sendmail 8.12.0+Sun/8.12.0; Sun, 4 Sep 2001 3:52:56 -0700(PDT)
expn sandy
250 2.1.5 <sandy@phoenix.example.com>
quit
```

If you cannot use `mconnect` to connect to an `SMTP` port, check these conditions.

- Is the system load too high?
- Is the `sendmail` daemon running?
- Does the system have the appropriate `/etc/mail/sendmail.cf` file?
- Is port 25 (the port that `sendmail` uses) active?

How to Log Messages

Your mail services logs most error messages by using the `syslogd` program. By default, the `syslogd` program sends these messages to a system called `loghost`, which is specified in the `/etc/hosts` file. You can define `loghost` to hold all logs for an entire NIS domain. If no `loghost` is specified, error messages from `syslogd` are not reported.

The `/etc/syslog.conf` file controls where the `syslogd` program forwards messages. You can change the default configuration by editing the `/etc/syslog.conf` file. You must restart the `syslog` daemon for any changes to become active. To gather information about mail, you can add the following selections to the file.

- `mail.alert` – Messages about conditions that should be fixed now
- `mail.crit` – Critical messages

- `mail.warning` – Warning messages
- `mail.notice` – Messages that are not errors, but might need attention
- `mail.info` – Informational messages
- `mail.debug` – Debugging messages

The following entry in the `/etc/syslog.conf` file sends a copy of all critical, informational, and debug messages to `/var/log/syslog`.

```
mail.crit;mail.info;mail.debug              /var/log/syslog
```

Each line in the system log contains a timestamp, the name of the system that generated the line, and a message. The `syslog` file can log a large amount of information.

The log is arranged in a succession of levels. At the lowest level, only unusual occurrences are logged. At the highest level, even the most mundane and uninteresting events are recorded. As a convention, log levels under 10 are considered "useful." Log levels higher than 10 are usually used for debugging. See the "Customizing System Message Logging" in *System Administration Guide: Advanced Administration* for information about `loghost` and the `syslogd` program.

Other Sources for Mail Diagnostic Information

For other diagnostic information, check the following sources.

- Look at the `Received` lines in the header of the message. These lines trace the route the message took as it was relayed. Remember to consider time–zone differences.
- Look at the messages from `MAILER-DAEMON`. These messages typically report delivery problems.
- Check the system log that records delivery problems for your group of systems. The `sendmail` program always records its activities in the system log. You might want to modify the `crontab` file to run a shell script nightly that searches the log for `SYSERR` messages and mails any messages that it finds to the postmaster.
- Use the `mailstats` program to test mail types and determine the number of incoming and outgoing messages.

Resolving Error Messages

This section describes how you can resolve some `sendmail`–related error messages that are in the Solaris 9 operating environment. You can also refer to http://www.sendmail.org/faq/.

The following error messages contain two or more of the following types of information.

- *Cause:* What might have happened to cause the message
- *Description:* What the user was doing when the error message occurred
- *Action:* What you can do to fix the problem or to continue with your work
- *Technical Notes:* Background information that might be interesting or helpful to a technical audience, such as developers
- *See Also:* Suggests further reading

`451 timeout waiting for input during` *source*

Cause: When `sendmail` reads from any source that might time out, such as an SMTP connection, it sets a timer to the value of various `Timeout` options before reading begins. If the read is not completed before the timer expires, this message appears and reading stops. (Usually this situation occurs during `RCPT`.) The mail message is then queued for later delivery.

Action: If you see this message often, increase the value of various `Timeout` options in the `/etc/mail/sendmail.cf` file. If the timer is already set to a large number, look for hardware problems, such as poor network cabling or connections.

See Also: For more information about the `Timeout` option, refer to "Changes to the `Timeout` Option" on page 401. If you are using online documentation, the term "timeouts" is a good search string.

`550` *hostname*`... Host unknown`

Cause: This `sendmail` message indicates that the destination host machine, which is specified by the portion of the address after the at sign (@), was not found during domain name system (DNS) lookup.

Action: Use the `nslookup` command to verify that the destination host exists in that domain or other domains, perhaps with a slightly different spelling. Otherwise, contact the intended recipient and ask for a proper address.

`550` *username*`... User unknown`

Cause: This `sendmail` message indicates that the intended recipient, who is specified by the portion of the address before the at sign (@), could not be located on the destination host machine.

Action: Check the email address and try again, perhaps with a slightly different spelling. If this remedy does not work, contact the intended recipient and ask for a proper address.

`554` *`hostname`*`... Local configuration error`

Cause: This `sendmail` message usually indicates that the local host is trying to send mail to itself.

Action: Check the value of the `$j` macro in the `/etc/mail/sendmail.cf` file to ensure that this value is a fully qualified domain name.

Technical Notes: When the sending system provides its host name to the receiving system (in the SMTP `HELO` command), the receiving system compares its name to the sender's name. If these names are the same, the receiving system issues this error message and closes the connection. The name that is provided in the `HELO` command is the value of the `$j` macro.

See Also: For additional information, refer to http://www.sendmail.org/faq/section4.html#4.5.

`config error: mail loops back to myself.`

Cause: If you set up an MX record and make host *bar* the mail exchanger for domain *foo*, but you do not configure host *bar* to know that it is the mail exchanger for domain *foo*, you get this error message.

Also, another possibility is that both the sending system and the receiving system are identifying as the same domain.

Action: For instructions, refer to http://www.sendmail.org/faq/section4.html#4.5.

`host name configuration error`

Action: Follow the instructions that were provided for resolving this error message, `554` *`hostname`*`... Local configuration error`.

Technical Notes: This is an old `sendmail` message, which replaced `I refuse to talk to myself` and is now replaced by the `Local configuration error` message.

`user unknown`

Description: When you try to send mail to a user, the error `Username... user unknown` is displayed. The user is on the same system.

Action: Check for a typographical error in the entered email address. Otherwise, the user could be aliased to a nonexistent email address in `/etc/mail/aliases` or in the user's `.mailrc` file. Also, check for uppercase characters in the user name. Preferably, email addresses should not be case sensitive.

See Also: For additional information, refer to http://www.sendmail.org/faq/section4.html#4.17.

CHAPTER 25

Mail Services (Reference)

The `sendmail` program is a mail transport agent that uses a configuration file to provide aliasing and forwarding, automatic routing to network gateways, and flexible configuration. The Solaris operating environment supplies standard configuration files that most sites can use. Chapter 23 provided an introduction to the components of mail services and a description of a typical mail service configuration. Chapter 24 explained how to set up and administer an electronic mail system. This chapter provides information on the following topics.

Look in Chapter 26 for a description of the new features that are included in version 8.12 of `sendmail`, the version that is in this Solaris 9 release. You can also read about changes to `mail.local`, `mailstats`, `makemap`, and about a new maintenance utility, `editmap`. For details not covered in these chapters, you can look in the man pages for `sendmail`(1M), `mail.local`(1M), `mailstats`(1), `makemap`(1M), and `editmap`(1M).

Solaris Version of `sendmail`

This section, which includes the following topics, describes some of the differences in the Solaris version of `sendmail` as compared to the generic Berkeley version.

Flags Used and Not Used to Compile `sendmail`

The following tables list the flags that are used when compiling the version of `sendmail` that is delivered with the Solaris 9 release. If your configuration requires other flags, you need to download the source and recompile the binary yourself. You can find information about this process at http://www.sendmail.org.

TABLE 25–1 General `sendmail` Flags

Flag	Description
`SOLARIS=20900`	Support for the Solaris 9 operating environment.
`MILTER`	Support for the Mail Filter API.
`NETINET6`	Support for IPv6. This flag has been moved from `conf.h` to `Makefile`.

TABLE 25–2 Maps and Database Types

Flag	Description
`NDBM`	Support for `ndbm` databases.
`NEWDB`	Support for `db` databases.
`USERDB`	Support for the `User` database.
`NIS`	Support for `nis` databases.
`NISPLUS`	Support for `nisplus` databases.
`LDAPMAP`	Support for `LDAP` maps.
`MAP_REGEX`	Support for regular expression maps.

TABLE 25–3 Solaris Flags

Flag	Description
`SUN_EXTENSIONS`	Support for Sun extensions that are included in `sun_compat.o`.
`SUN_LOOKUP_MACRO`	Support for the `L` and `G` configuration commands in `sendmail.cf`. Use of these commands is not recommended.
`SUN_DEFAULT_VALUES`	Support only for the default values in the Solaris flag, `SUN_CONTENT_LENGTH`.

TABLE 25–3 Solaris Flags *(Continued)*

Flag	Description
SUN_INIT_DOMAIN	For backward compatibility, support for the use of NIS domain names to fully qualify the local host name. For more information, look for vendor-specific information in http://www.sendmail.org.
SUN_CONTENT_LENGTH	Support for the Content-Length: header in messages to files. For more information, look for vendor-specific information in http://www.sendmail.org.
SUN_SIMPLIFIED_LDAP	Support for a simplified LDAP API, which is specific to Sun. For more information, look for vendor-specific information in http://www.sendmail.org.
VENDOR_DEFAULT=VENDOR_SUN	Selects Sun as the default vendor.

The following table lists generic flags that are not used to compile the version of `sendmail` that is delivered with the Solaris 9 release.

TABLE 25–4 Generic Flags Not Used in the Solaris Version of `sendmail`

Flag	Description
SASL	Simple Authentication and Security Layer (RFC 2554)
STARTTLS	Transaction Level Security (RFC 2487)

To see a list of the flags used to compile `sendmail`, use the following command.

```
% /usr/lib/sendmail -bt -d0.10 < /dev/null
```

Note – The preceding command does not list the flags that are specific to Sun.

Alternative `sendmail` Commands

The Solaris release does not include all of the command synonyms that are provided in the generic release from Berkeley. This table includes a complete list of the command aliases, whether they are included in the Solaris release, and how to generate the same behavior by using `sendmail`.

TABLE 25–5 Alternate `sendmail` Commands

Alternate Name	In the Solaris Release?	Options With `sendmail`
`hoststat`	No	`sendmail -bh`
`mailq`	Yes	`sendmail -bp`
`newaliases`	Yes	`sendmail -bi`
`purgestat`	No	`sendmail -bH`
`smtpd`	No	`sendmail -bd`

Versions of the Configuration File

The Solaris 9 version of `sendmail` includes a configuration option that enables you to define the version of the `sendmail.cf` file. This option enables older configuration files to be used with the current version of `sendmail`. You can set the version level to values between `0` and `9`. You can also define the vendor. Either Berkeley or Sun are valid vendor options. If a version level is specified but no vendor is defined, `Sun` is used as the default vendor setting. The following table lists some of the valid options.

TABLE 25–6 Configuration File Version Values

Field	Description
`V7/Sun`	Setting that was used for Version 8.8 of `sendmail`.
`V8/Sun`	Setting that was used for Version 8.9 of `sendmail`. This setting was included in the Solaris 8 release.
`V9/Sun`	Setting that was used for versions 8.10 and 8.11 of `sendmail`.
`V10/Sun`	Setting that is used for versions 8.12 of `sendmail`. Version 8.12 is the default for the Solaris 9 release.

Note – You are urged not to use `V1/Sun`. For more information, refer to http://www.sendmail.org/vendor/sun/differences.html#4.

For task information, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24.

Software and Hardware Components of Mail Services

This section describes the software and hardware components of a mail system.

Software Components

Each mail service includes at least one of each of the following software components.

This section also describes these software components.

Mail User Agent

The *mail user agent* is the program that acts as the interface between the user and mail transfer agent. The `sendmail` program is a mail transfer agent. The Solaris operating environment supplies the following mail user agents.

- `/usr/bin/mail`
- `/usr/bin/mailx`
- `$OPENWINHOME/bin/mailtool`
- `/usr/dt/bin/dtmail`

Mail Transfer Agent

The *mail transfer agent* is responsible for the routing of mail messages and the resolution of mail addresses. This agent is also known as a mail *transport* agent. The transfer agent for the Solaris operating environment is `sendmail`. The transfer agent performs these functions.

- Accepts messages from the mail user agent
- Resolves destination addresses
- Selects a proper delivery agent to deliver the mail
- Receives incoming mail from other mail transfer agents

Local Delivery Agent

A *local delivery agent* is a program that implements a mail delivery protocol. The following local delivery agents are provided with the Solaris operating environment.

- The UUCP local delivery agent, which uses `uux` to deliver mail
- The local delivery agent, which is `mail.local` in the standard Solaris release

Chapter 26 provides information on these related topics.

- "New Delivery Agent Flags" on page 410
- "New Equates for Delivery Agents" on page 410
- "Changes to `mail.local`" on page 415

Mailers

Mailer is a `sendmail`-specific term. A *mailer* is used by `sendmail` to identify a specific instance of a customized local delivery agent or a customized mail transfer agent. You need to specify at least one mailer in your `sendmail.cf` file. For task information, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24. This section provides a brief description of two types of mailers.

- "Simple Mail Transport Protocol (SMTP) Mailers" on page 352
- "UNIX-to-UNIX Copy Program (UUCP) Mailers" on page 353

For additional information about mailers, see http://www.sendmail.org/m4/readme.html or `/usr/lib/mail/README`.

Simple Mail Transport Protocol (SMTP) Mailers

SMTP is the standard mail protocol that is used on the Internet. This protocol defines these mailers.

`smtp` Provides regular (historic-style) SMTP transfers to other servers

`esmtp` Provides extended SMTP transfers to other servers

`smtp8` Provides SMTP transfers to other servers without converting 8–bit data to MIME

`dsmtp` Provides on-demand delivery by using the `F=%` mailer flag. Refer to "Changes to the `MAILER()` Declaration" on page 409 and "New Delivery

Agent Flags" on page 410 in Chapter 26.

UNIX-to-UNIX Copy Program (UUCP) Mailers

If possible, avoid using UUCP. For an explanation, refer to http://www.sendmail.org/m4/uucp.html or do a search in `/usr/lib/mail/README` on this string, `USING UUCP MAILERS`.

UUCP defines these mailers.

`uucp-old` Names in the `$=U` class are sent to `uucp-old`. `uucp` is the obsolete name for this mailer. The `uucp-old` mailer uses an exclamation-point address in the headers.

`uucp-new` Names in the `$=Y` class are sent to `uucp-new`. Use this mailer when you know that the receiving UUCP mailer can manage multiple recipients in one transfer. `suucp` is the obsolete name for this mailer. The `uucp-new` mailer also uses an exclamation-point address in the headers.

If `MAILER(smtp)` is also specified in your configuration, two more mailers are defined.

`uucp-dom` This mailer uses domain-style addresses and, basically, applies the SMTP rewriting rules.

`uucp-uudom` Names in the `$=Z` class are sent to `uucp-uudom`. `uucp-uudom` and `uucp-dom` use the same header address format, domain-style addresses.

Note – Because the `smtp` mailer modifies the UUCP mailer, always put `MAILER(smtp)` before `MAILER(uucp)` in your `.mc` file.

Mail Addresses

The *mail address* contains the name of the recipient and the system to which the mail message is delivered. When you administer a small mail system that does not use a name service, addressing mail is easy. The login names uniquely identify the users. When, however, you are administering a mail system that has more than one system with mailboxes, one or more domains, or when you have a UUCP (or other) mail connection to the outside world, mail addressing becomes more complex. The information in the following sections can help you understand the parts and complexities of a mail address.

- "Route–Independent Mail Addresses" on page 355

Domains and Subdomains

Email addressing uses domains. A *domain* is a directory structure for network address naming. A domain can have one or more *subdomains*. The domain and subdomains of an address can be compared to the hierarchy of a file system. Just as a subdirectory is considered to be inside the directory above it, each subdomain in a mail address is considered to be inside the location to its right.

The following table shows some top-level domains.

TABLE 25–7 Top-Level Domains

Domain	Description
`com`	Commercial sites
`edu`	Educational sites
`gov`	United States government installations
`mil`	United States military installations
`net`	Networking organizations
`org`	Other nonprofit organizations

Domains are case insensitive. You can use uppercase, lowercase, or mixed-case letters in the domain part of an address without making any difference.

For more information about domains, refer to "Introduction to DNS" in *System Administration Guide: Naming and Directory Services*.

Name Service Domain Name and Mail Domain Name

When you are working with name service domain names and mail domain names, remember the following.

- By default, the `sendmail` program strips the first component from the NIS or NIS+ domain name to form the mail domain name. For example, if an NIS+ domain name were `bldg5.example.com`, its mail domain name would be `example.com`.
- Although mail domain addresses are case insensitive, the NIS or NIS+ domain name is not. For the best results, use lowercase characters when setting up the mail and NIS or NIS+ domain names.
- The DNS domain name and the mail domain name must be identical.

For more information, refer to “Interactions of `sendmail` With Name Services” on page 379.

Typical Format for Mail Addresses

Typically, a mail address has the following format. For further details, refer to “Route–Independent Mail Addresses” on page 355.

user@subdomain. subdomain2.subdomain1.top-level-domain

The part of the address to the left of the `@` sign is the local address. The local address can contain the following.

- Information about routing with another mail transport (for example, `bob::vmsvax@gateway` or `smallberries%mill.uucp@gateway`)
- An alias (for example, `iggy.ignatz`)

Note – The receiving mailer is responsible for determining what the local part of the address means. For information about mailers, refer to “Mailers” on page 352.

The part of the address to the right of the `@` sign shows the domain levels, which is where the local address resides. A dot separates each subdomain. The domain part of the address can be an organization, a physical area, or a geographic region. Furthermore, the order of domain information is hierarchical—the more local the subdomain, the closer it is to the `@` sign.

Route–Independent Mail Addresses

Mail addresses can be route independent. Route-independent addressing requires the sender of an email message to specify the name of the recipient and the final destination. A high-speed network, such as the Internet, uses route-independent addresses. Route-independent addresses can have this format.

user@host . domain

Route-independent addresses for UUCP connections can have this address format.

host . domain ! *user*

The increased popularity of the domain-hierarchical naming scheme for computers is making route-independent addresses more common. In fact, the most common route-independent address omits the host name and relies on the domain name service to properly identify the final destination of the email message.

user@domain

Route-independent addresses are read by searching for the @ sign and then reading the domain hierarchy from the right (the highest level) to the left (the most specific part of the address to the right of the @ sign).

Mailbox Files

A *mailbox* is a file that is the final destination for email messages. The name of the mailbox can be the user name or the identity of a specific function, like the postmaster. Mailboxes are in the `/var/mail/`*username* file, which can exist either on the user's local system or on a remote mail server. In either instance, the mailbox is on the system to which the mail is delivered.

Mail should always be delivered to a local file system so that the user agent can pull mail from the mail spool and store it readily in the local mailbox. Do not use NFS-mounted file systems as the destination for a user's mailbox. Specifically, do not direct mail to a mail client that is mounting the `/var/mail` file system from a remote server. Mail for the user, in this instance, should be addressed to the mail server and not to the client host name. NFS-mounted file systems can cause problems with mail delivery and handling.

The `/etc/mail/aliases` file and name services such as NIS and NIS+ provide mechanisms for creating aliases for electronic mail addresses, so that users do not need to know the precise local name of a user's mailbox.

The following table shows some common naming conventions for special-purpose mailboxes.

TABLE 25–8 Conventions for the Format of Mailbox Names

Format	Description
username	User names are frequently the same as mailbox names.
Firstname.*Lastname* *Firstname*_*Lastname* *Firstinitial*.*Lastname* *Firstinitial*_*Lastname*	User names can be identified as full names with a dot (or an underscore) that separates the first and last names, or by a first initial with a dot (or an underscore) that separates the initial and the last name.
`postmaster`	Users can address questions and report problems with the mail system to the `postmaster` mailbox. Each site and domain should have a `postmaster` mailbox.
`MAILER-DAEMON`	`sendmail` automatically routes any mail that is addressed to the `MAILER-DAEMON` to the postmaster.
aliasname`-request`	Names that end in `-request` are administrative addresses for distribution lists. This address should redirect mail to the person who maintains the distribution list.
`owner-`*aliasname*	Names that begin with `owner-` are administrative addresses for distribution lists. This address should redirect mail to the person who handles mail errors.

TABLE 25–8 Conventions for the Format of Mailbox Names *(Continued)*

Format	Description
`owner-owner`	This alias is used when there is no `owner-`*aliasname* alias for errors to be returned to. This address should redirect mail to the person who handles mail errors and should be defined on any system that maintains a large number of aliases.
local`%`*domain*	The percent sign (`%`) marks a local address that is expanded when the message arrives at its destination. Most mail systems interpret mailbox names with `%` characters as full mail addresses. The `%` is replaced with an `@`, and the mail is redirected accordingly. Although many people use the `%` convention, it is not a formal standard. This convention is referred to as the "percent hack." This feature is often used to help debug mail problems.

Starting with `sendmail` version 8, the envelope sender for mail that is sent to a group alias has been changed to the address that is expanded from the owner alias, if an owner alias exists. This change enables any mail errors to be sent to the alias owner, rather than being returned to the sender. With this change, users notice that mail that was sent to an alias looks as if it came from the alias owner, when delivered. The following alias format helps with some of the problems that are associated with this change.

```
mygroup: :include:/pathname/mygroup.list
owner-mygroup: mygroup-request
mygroup-request: sandys, ignatz
```

In this example, the `mygroup` alias is the actual mail alias for the group. The `owner-mygroup` alias receives error messages. The `mygroup-request` alias should be used for administrative requests. This structure means that in mail sent to the `mygroup` alias, the envelope sender changes to `mygroup-request`.

Mail Aliases

An *alias* is an alternate name. For email, you can use aliases to assign a mailbox location or to define mailing lists. For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24. Also, you can refer to "Mail Alias Files" on page 373 in this chapter.

For large sites, the mail alias typically defines the location of a mailbox. Providing a mail alias is like providing a room number as part of the address for an individual at a large corporation that occupies multiple rooms. If you do not provide the room number, the mail is delivered to a central address. Without a room number, extra effort is required to determine where within the building the mail is to be delivered, and the possibility of an error increases. For example, if two people who are named Kevin Smith are in the same building, only one of them might get mail. To correct the problem, each Kevin Smith should have a room number added to his address.

Use domains and location-independent addresses as much as possible when you create mailing lists. To enhance portability and flexibility of alias files, make your alias

entries in mailing lists as generic and system independent as possible. For example, if you have a user who is named `ignatz` on system `mars`, in domain `example.com`, create the alias `ignatz@example` instead of `ignatz@mars`. If user `ignatz` changes the name of his system but remains within the `example` domain, you do not need to update alias files to reflect the change in system name.

When you create alias entries, type one alias per line. You should have only one entry that contains the user's system name. For example, you could create the following entries for user `ignatz`.

```
ignatz: iggy.ignatz
iggyi: iggy.ignatz
iggy.ignatz: ignatz@mars
```

You can create an alias for local names or domains. For example, an alias entry for user `fred`, who has a mailbox on the system `mars` and is in the domain `planets`, could have this entry in the NIS+ aliases table.

```
fred: fred@planets
```

When you create mail lists that include users outside your domain, create the alias with the user name and the domain name. For example, if you have a user named `smallberries` on system `privet`, in domain `example.com`, create the alias as `smallberries@example.com`. The email address of the sender is now automatically translated to a fully qualified domain name when mail goes outside the user's domain.

The following list describes methods for creating and administering mail alias files.

- You can create mail aliases for global use in the NIS+ `mail_aliases` table, the NIS `aliases` map, or in local `/etc/mail/aliases` files. You can also create and administer mailing lists that use the same alias files.
- Depending on the configuration of your mail services, you can administer aliases by using the NIS or NIS+ name service to maintain a global `aliases` database or by updating all the local `/etc/mail/aliases` files to keep them synchronized.
- Users can also create and use aliases. They can create aliases either in their local `~/.mailrc` file, which only they can use, or in their local `/etc/mail/aliases` file, which anyone can use. Users cannot normally create or administer NIS or NIS+ alias files.

Hardware Components

You can provide the three required elements of mail configuration in the same system or have separate systems provide these elements.

- "Mail Host" on page 359
- "Mail Server" on page 359

- "Mail Client" on page 360

When users are to communicate with networks outside your domain, you must also add a fourth element, a mail gateway. For more information, refer to "Mail Gateway" on page 360. The following sections describe each hardware component.

Mail Host

A *mail host* is the machine that you designate as the main mail machine on your network. A mail host is the machine to which other systems at the site forward mail that they cannot deliver. You designate a system as a mail host in the `hosts` database by adding the word `mailhost` to the right of the IP address in the local `/etc/hosts` file or in the hosts file in the name service. You must also use the `main.cf` file as the mail configuration file on the mail host system. For detailed task information, refer to "How to Set Up a Mail Host" on page 313 in Chapter 24.

A good candidate for a mail host is a system on the local area network that also has a modem for setting up PPP or UUCP links over telephone lines. Another good candidate is a system that is configured as a router from your network to the Internet global network. For more information, refer to "Configuring Routers" in *System Administration Guide: IP Services*. If none of the systems on your local network has a modem, designate one as the mail host.

Some sites use standalone machines that are not networked in a time-sharing configuration. That is, the standalone machine serves terminals that are attached to its serial ports. You can set up electronic mail for this configuration by designating the standalone system as the mail host of a one-system network. "Overview of the Hardware Components" on page 301 in Chapter 23 provides a figure that shows a typical email configuration.

Mail Server

A *mailbox* is a single file that contains email for a particular user. Mail is delivered to the system where the user's mailbox resides, which can be on a local machine or a remote server. A *mail server* is any system that maintains user mailboxes in its `/var/mail` directory. For task information, refer to "How to Set Up a Mail Server" on page 309 in Chapter 24.

The mail server routes all mail from a client. When a client sends mail, the mail server puts it in a queue for delivery. After the mail is in the queue, a user can reboot or turn off the client without losing those mail messages. When the recipient gets mail from a client, the path in the `From` line of the message contains the name of the mail server. If the recipient responds, the response goes to the user's mailbox. Good candidates for mail servers are systems that provide a home directory for users or systems that are backed up regularly.

If the mail server is not the user's local system, users in configurations that use NFS software can mount the `/var/mail` directory by using the `/etc/vfstab` file (if they have `root` access) or by using the automounter. If NFS support is not available, users can log in to the server to read their mail.

If users on your network send other types of mail, such as audio files or files from desktop publishing systems, you need to allocate more space on the mail server for mailboxes.

One advantage to establishing a mail server for all mailboxes is that it simplifies backups. Backups can be difficult to do when mail is spread over many systems. The disadvantage of storing many mailboxes on one server is that the server can be a single point of failure for many users. However, the advantages of providing good backups usually make the risk worthwhile.

Mail Client

A *mail client* is any system that receives mail on a mail server and does not have a local `/var/mail` directory. This type of configuration is known as remote mode. Remote mode is enabled by default in `/etc/mail/subsidiary.cf`.

You must check that the mail client has the appropriate entry in the `/etc/vfstab` file and a mount point to mount the mailbox from the mail server. Also, ensure that the alias for the client is directed to the mail server's host name, not to the client's name. For task information, refer to "How to Set Up a Mail Client" on page 311 in Chapter 24.

Mail Gateway

The *mail gateway* is a machine that handles connections between networks that run different communications protocols or communications between different networks that use the same protocol. For example, a mail gateway might connect a TCP/IP network to a network that runs the Systems Network Architecture (SNA) protocol suite.

The simplest mail gateway to set up is one that connects two networks that use the same protocol or mailer. This system handles mail with an address for which `sendmail` cannot find a recipient in your domain. If a mail gateway exists, `sendmail` uses it for sending and receiving mail outside your domain.

You can set up a mail gateway between two networks that use unmatched mailers, as shown in the next figure. To support this configuration, you must customize the `sendmail.cf` file on the mail gateway system, which can be a difficult and time-consuming process.

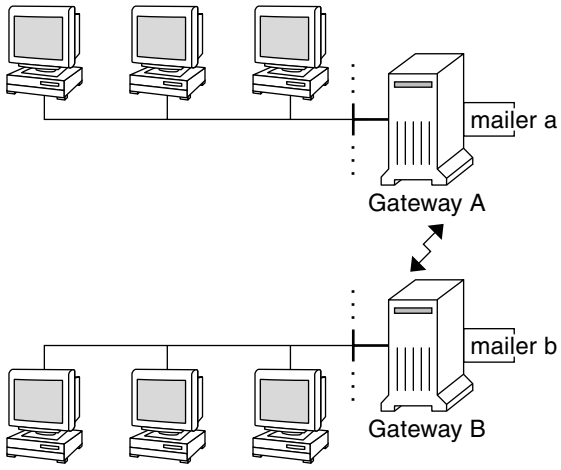


FIGURE 25–1 Gateway Between Different Communications Protocols

If you have to set up a mail gateway, you should find a gateway configuration file that is close to what you need and modify it to conform to your situation.

If you have a machine that provides connections to the Internet, you can configure that machine as the mail gateway. Carefully consider your site's security needs before you configure a mail gateway. You might need to create a firewall gateway between your corporate network and the outside world, and set up that gateway as the mail gateway. For task information, refer to "How to Set Up a Mail Gateway" on page 314 in Chapter 24.

Mail Service Programs and Files

Mail services include many programs and daemons that interact with each other. This section introduces the files, programs, terms, and concepts that are related to administering electronic mail.

Contents of the `/usr/bin` Directory

The following table shows the contents of the `/usr/bin` directory, which is used for mail services.

Name	Type	Description
`aliasadm`	File	A program to manipulate the NIS+ aliases map.
`mail`	File	A user agent.
`mailcompat`	File	A filter to store mail in SunOS 4.1 mailbox format.
`mailq`	Link	A link to `/usr/lib/sendmail`. Used to list the mail queue.
`mailstats`	File	A program that is used to read mail statistics that are stored in the `/etc/mail/sendmail.st` file (if present).
`mailx`	File	A user agent.
`mconnect`	File	A program that connects to the mailer for address verification and debugging.
`praliases`	File	A command to "uncompile" the alias database. Refer to the uncompile information that is provided in the man page for `praliases`(1).
`rmail`	Link	A link to `/usr/bin/mail`. Command that is often used to permit only the sending of mail.
`vacation`	File	A command to set up an automatic reply to mail.

Contents of the `/etc/mail` Directory

The following table shows the contents of the `/etc/mail` directory.

Name	Type	Description
`Mail.rc`	File	Default settings for the `mailtool` user agent.

Name	Type	Description
aliases	File	Mail-forwarding information.
aliases.db	File	Default binary form of mail-forwarding information (created by running newaliases).
aliases.dir	File	Binary form of mail-forwarding information (created by running newaliases). Can still be used, but is no longer used by default in the Solaris 9 release.
aliases.pag	File	Binary form of mail-forwarding information (created by running newaliases). Can still be used, but is no longer used by default in the Solaris 9 release.
mailx.rc	File	Default settings for the mailx user agent.
main.cf	File	Sample configuration file for main systems.
relay-domains	File	List of all domains for which relaying is allowed. By default, only the local domain is allowed.
sendmail.cf	File	Configuration file for mail routing.
submit.cf	File	New configuration file for the mail submission program (MSP). For more information, refer to "New Configuration File, submit.cf" on page 386.
local-host-names	File	Optional file that you can create if the number of aliases for the mail host is too long.
helpfile	File	Help file that is used by the SMTP HELP command.
sendmail.pid	File	File that lists the PID of the listening daemon and is now in /var/run.
sendmail.st	File	sendmail statistics file. If this file is present, sendmail logs the amount of traffic through each mailer.
subsidiary.cf	File	Sample configuration file for subsidiary systems.
trusted-users	File	File that lists the users (one per line) who can be trusted to perform certain mail operations. By default, only root is in this file. Certain mail operations, when performed by untrusted users, result in the following warning, X-Authentication-Warning: header being added to a message.

Contents of the /usr/lib Directory

Table 25–9 shows the contents of the /usr/lib directory, which is used for mail services.

TABLE 25–9 Contents of the `/usr/lib` Directory

Name	Type	Description
`mail.local`	File	Mailer that delivers mail to mailboxes.
`sendmail`	File	Routing program, also known as the mail transfer agent.
`smrsh`	File	Shell program (**s**end**m**ail **r**estricted **sh**ell) that uses the "`|program`" syntax of `sendmail` to restrict programs that `sendmail` can run to those in the `/var/adm/sm.bin` directory. Refer to the `smrsh`(1M) man page for recommendations on what to include in `/var/adm/sm.bin`. To enable, include this `m4` command, `FEATURE('smrsh')`, in your `mc` file.

Contents of the `/usr/lib/mail` Directory

Within the `/usr/lib` directory is a subdirectory, `mail`, that contains all of the necessary files to build a `sendmail.cf` file. The contents of `mail` are shown in Table 25–10.

TABLE 25–10 Contents of the `/usr/lib/mail` Directory Used for Mail Services

Name	Type	Description
`README`	File	Describes the configuration files.
`cf`	Directory	Provides site-dependent and site-independent descriptions of hosts.
`cf/main.mc`	File	Previously named `cf/main-v7sun.mc`. Is the main configuration file.
`cf/makefile`	File	Provides rules for building new configuration files.
`cf/submit.mc`	File	Is the configuration file for the mail submission program (MSP), which is used to submit messages.
`cf/subsidiary.mc`	File	Previously named `cf/subsidiary-v7sun.mc`. Is the configuration file for hosts that NFS-mount `/var/mail` from another host.
`domain`	Directory	Provides site-dependent subdomain descriptions.

TABLE 25–10 Contents of the `/usr/lib/mail` Directory Used for Mail Services *(Continued)*

Name	Type	Description
`domain/generic.m4`	File	Is the generic domain file from Berkeley.
`domain/solaris-antispam.m4`	File	Is the domain file with changes that make `sendmail` function like previous Solaris versions, except that relaying is disabled completely, sender addresses with no host name are rejected, and unresolvable domains are rejected.
`domain/solaris-generic.m4`	File	Is the default domain file with changes that make `sendmail` function like previous Solaris versions.
`feature`	Directory	Contains definitions of specific features for particular hosts (see `README` for a full description of the features).
`m4`	Directory	Contains site-independent include files.
`mailer`	Directory	Contains definitions of mailers, which include `local`, `smtp`, and `uucp`.
`ostype`	Directory	Describes various operating system environments.
`ostype/solaris2.m4`	File	Defines default local mailer as `mail.local`.
`ostype/solaris2.ml.m4`	File	Defines default local mailer as `mail.local`.
`ostype/solaris2.pre5.m4`	File	Defines local mailer as `mail`.
`ostype/solaris8.m4`	File	Defines local mailer as `mail.local` (in LMTP mode), enables IPv6, specifies `/var/run` as the directory for the `sendmail.pid` file.
`sh`	Directory	Contains shell scripts that are used by the `m4` build process and migration aids.
`sh/check-permissions`	File	Checks permissions of `:include:` aliases and `.forward` files and their parent directory path for correct permissions.
`sh/check-hostname`	File	Verifies that `sendmail` is able to determine the fully qualified host name.

Other Files Used for Mail Services

Several other files and directories are used for mail services, as shown in Table 25–11.

TABLE 25–11 Other Files Used for Mail Services

Name	Type	Description
`sendmailvars.org_dir`	Table	NIS+ version of `sendmailvars` file.
`/etc/default/sendmail`	File	Lists the environment variables for the startup script for `sendmail`.
`/etc/shells`	File	Lists the valid login shells.
`/usr/sbin/editmap`	File	Queries and edits single records in database maps for `sendmail`.
`/usr/sbin/in.comsat`	File	Mail notification daemon.
`/usr/sbin/makemap`	File	Builds binary forms of keyed maps.
`/usr/sbin/newaliases`	Link	A link to `/usr/lib/sendmail`. Used to create the binary form of the alias database. Previously in `/usr/bin`.
`/usr/sbin/syslogd`	File	Error message logger, used by `sendmail`.
`/usr/sbin/etrn`	File	Perl script for starting the client-side remote mail queue.
`/usr/dt/bin/dtmail`	File	CDE mail user agent.
`/var/mail/mailbox1`, `/var/mail/mailbox2`	File	Mailboxes for delivered mail.
`/var/spool/clientmqueue`	Directory	Storage for mail that is delivered by the client daemon.
`/var/spool/mqueue`	Directory	Storage for mail that is delivered by the master daemon.
`$OPENWINHOME/bin/mailtool`	File	Window–based mail user agent.
`/var/run/sendmail.pid`	File	File that lists the PID of the listening daemon.

Interactions of Mail Programs

Mail services are provided by a combination of the following programs, which interact as shown in the simplified illustration in Figure 25–2.

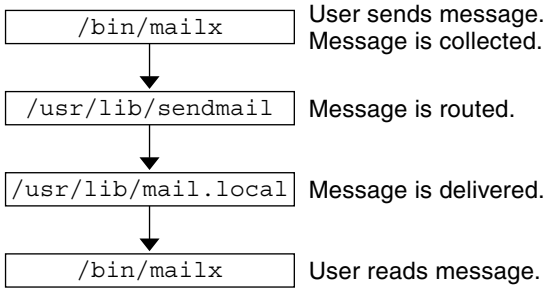


FIGURE 25–2 Interactions of Mail Programs

For a more detailed illustration, refer to Figure 25–4 in "`sendmail` Features" on page 370.

The following is a description of the interactions of mail programs.

1. Users send messages by using programs such as `mailx` or `mailtool`. See the man pages for `mailx`(1) or `mailtool`(1) for information about these programs.
2. The message is collected by the program that generated it and is passed to the `sendmail` daemon.
3. The `sendmail` daemon *parses* the addresses (divides them into identifiable segments) in the message. The daemon uses information from the configuration file, `/etc/mail/sendmail.cf`, to determine network name syntax, aliases, forwarding information, and network topology. By using this information, `sendmail` determines the route a message must follow to get to a recipient.
4. The `sendmail` daemon passes the message to the appropriate system.
5. The `/usr/lib/mail.local` program on the local system delivers the mail to the mailbox in the `/var/mail/`*username* directory of the recipient of the message.
6. The recipient is notified that mail has arrived and retrieves it by using `mail`, `mailx`, `mailtool`, or a similar program.

`sendmail` Program

The following list describes some of the capabilities of the `sendmail` program.

- `sendmail` can use different types of communications protocols, such as TCP/IP and UUCP.

- `sendmail` implements an SMTP server, message queueing, and mailing lists.
- `sendmail` controls name interpretation by using a pattern-matching system that can work with the following naming conventions.
  - Domain-based naming convention. The domain technique separates the issue of physical versus logical naming. For more information on domains, refer to "Mail Addresses" on page 353.
  - Improvised techniques, such as providing network names that appear local to hosts on other networks.
  - Arbitrary (older) naming syntaxes.
  - Disparate naming schemes.

The Solaris operating environment uses the `sendmail` program as a mail router. The following list describes some of its functions.

- `sendmail` is responsible for receiving and delivering email messages.
- `sendmail` is an interface between mail–reading programs like `mail`, `mailx`, and `mailtool`, and mail–transport programs like `uucp`.
- `sendmail` controls email messages that users send.
  - By evaluating the recipients' addresses
  - By choosing an appropriate delivery program
  - By rewriting the addresses in a format that the delivery agent can handle
  - By reformatting the mail headers as required
  - By finally passing the transformed message to the mail program for delivery

For more information about the `sendmail` program, refer to the following topics.

- "`sendmail` and Its Rerouting Mechanisms" on page 368
- "`sendmail` Features" on page 370
- "`sendmail` Configuration File" on page 371

`sendmail` and Its Rerouting Mechanisms

The `sendmail` program supports three mechanisms for mail rerouting. The mechanism you choose depends on the type of change that is involved.

- A server change
- A domain-wide change
- A change for one user

Additionally, the rerouting mechanism you choose can affect the level of administration that is required. Consider the following options.

1. One rerouting mechanism is *aliasing*.

Aliasing can map names to addresses on a server-wide basis or a name service-wide basis, depending on the type of file that you use.

Consider the following advantages and disadvantages for name service aliasing.

- The use of a name service (such as NIS or NIS+) alias file permits mail rerouting changes to be administered from a single source. However, name service aliasing can create lag time when the rerouting change is propagated.
- Name service administration is usually restricted to a select group of system administrators. A normal user would not administer this file.

Consider the following advantages and disadvantages for using a server alias file.

- By using a server alias file, rerouting can be managed by anyone who can become `root` on the designated server.
- Server aliasing should create little or no lag time when the rerouting change is propagated.
- The change only affects the local server, which might be acceptable if most of the mail is sent to one server. However, if you need to propagate this change to many mail servers, use a name service.
- A normal user would not administer this change.

For more information, refer to "Mail Alias Files" on page 373 in this chapter. For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

2. The next mechanism is *forwarding*.

   This mechanism permits users to administer mail rerouting. Local users can reroute their incoming mail to the following.

   - Another mailbox
   - A different mailer
   - Another mail host

   This mechanism is supported through the use of `.forward` files. For more information about these files, refer to "`.forward` Files" on page 376 in this chapter. For a task map, refer to "Administering `.forward` Files (Task Map)" on page 335 in Chapter 24.

3. The last rerouting mechanism is *inclusion*.

   This mechanism allows users to maintain alias lists, instead of requiring `root` access. To provide this feature, the `root` user must create an appropriate entry in the alias file on the server. After this entry is created, the user can reroute mail as necessary. For more information on inclusion, refer to "`/etc/mail/aliases` File" on page 373 in this chapter. For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

Figure 25–3 shows how `sendmail` uses aliases. Programs that read mail, such as `/usr/bin/mailx`, can have aliases of their own, which are expanded before the

message reaches `sendmail`. The aliases for `sendmail` can come from a number of name service sources (local files, NIS or NIS+). The order of the lookup is determined by the `nsswitch.conf` file. Refer to the `nsswitch.conf`(4) man page.

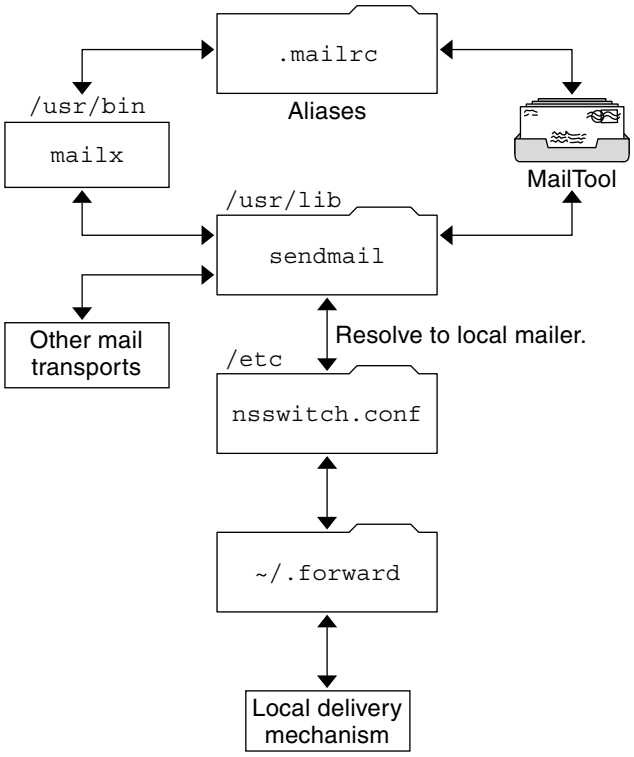


FIGURE 25–3 How `sendmail` Uses Aliases

`sendmail` Features

The `sendmail` program provides the following features.

- `sendmail` is reliable. It is designed to correctly deliver every message. No message should ever become completely lost.
- `sendmail` uses existing software for delivery whenever possible.
- `sendmail` can be configured to handle complex environments, including multiple connections to a single network type, such as UUCP or Ethernet. `sendmail` checks the contents of an address as well as its syntax to determine which mailer to use.
- `sendmail` uses configuration files to control mail configuration instead of requiring that configuration information be compiled into the code.

- Users can maintain their own mailing lists. In addition, individuals can specify their own forwarding mechanism without modifying the domain-wide alias file, typically located in the domain-wide aliases that are maintained by NIS or NIS+.
- Each user can specify a custom mailer to process incoming mail, which can provide functions like returning a message that reads: "I am on vacation." See the `vacation`(1) man page for more information.
- `sendmail` batches addresses to a single host to reduce network traffic.

Figure 25–4 shows how `sendmail` interacts with the other programs in the mail system.

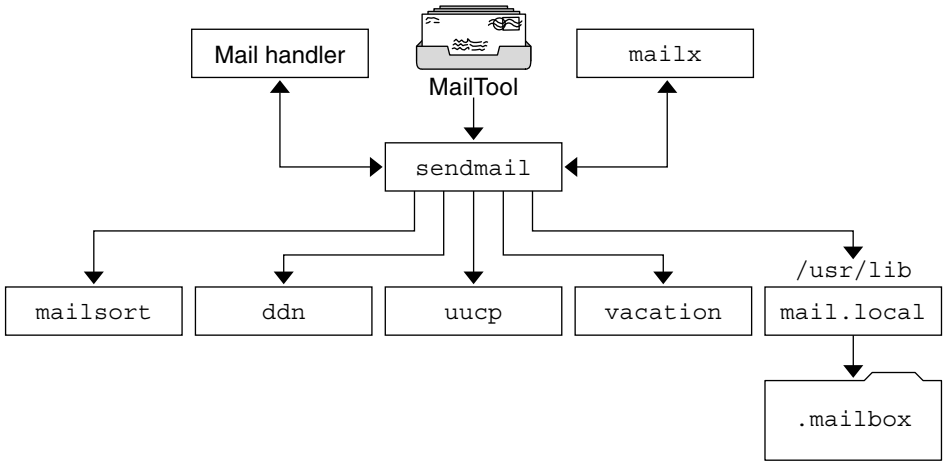


FIGURE 25–4 Interaction of `sendmail` With Other Mail Programs

As shown in Figure 25–4, the user interacts with a mail-generating and mail-sending program. When the mail is submitted, the mail-generating program calls `sendmail`, which routes the message to the correct mailers. Because some of the senders might be network servers and some of the mailers might be network clients, you can use `sendmail` as an Internet mail gateway. See "Interactions of Mail Programs" on page 367 for a more detailed description of the process.

`sendmail` Configuration File

A *configuration file* controls the way that `sendmail` performs its functions. The configuration file determines the choice of delivery agents, address rewriting rules, and the format of the mail header.

The `sendmail` program uses the information from the `/etc/mail/sendmail.cf` file to perform its functions. Each system has a default `sendmail.cf` file that is installed in the `/etc/mail` directory. You do not need to edit or change the default

configuration file for mail servers or mail clients. The only systems that require a customized configuration file are mail hosts and mail gateways.

The Solaris operating environment provides three default configuration files in the `/etc/mail` directory.

1. A configuration file named `main.cf` for the system (or systems) you designate as the mail host or a mail gateway
2. A configuration file named `subsidiary.cf`, which is a duplicate copy of the default `sendmail.cf` file
3. A configuration file named `submit.cf`, which is used to run `sendmail` in mail submission program mode, instead of daemon mode. For more information, refer to "New Configuration File, `submit.cf`" on page 386.

The configuration file you use on a system depends on the role of the system in your mail service.

- For mail clients or mail servers, you do not need to do anything to set up or edit the default configuration file.
- To set up a mail host or mail gateway, copy the `main.cf` file and rename it to `sendmail.cf` in the `/etc/mail` directory. Then reconfigure the `sendmail` configuration file to set the relay mailer and relay host parameters that are needed for your mail configuration. For task information, refer to "Setting Up Mail Services (Task Map)" on page 308 or "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24.

The following list describes some configuration parameters you can change, depending on the requirements of your site.

- Time values, which specify the following information.
  - Read timeouts. Refer to "Changes to the `Timeout` Option" on page 401.
  - Length of time a message remains undelivered in the queue before it is returned to the sender. Refer to "New Queue Features" on page 411. For a task map, refer to "Administering the Queue Directories (Task Map)" on page 332.
- Delivery modes, which specify how quickly mail is delivered.
- Load limits, which increase efficiency during busy periods by not attempting to deliver large messages, messages to many recipients, and messages to sites that have been down for a long time.
- Log level, which specifies the kinds of problems that are logged.

Mail Alias Files

You can use any of the following files, maps, or tables to maintain aliases.

Your method of maintaining aliases depends on who uses the alias and who needs to be able to change the alias. Each type of alias has unique format requirements.

If you are looking for task information, refer to "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

`.mailrc` Aliases

Aliases that are listed in a `.mailrc` file are accessible only by the user who owns the file. This restriction enables users to establish an alias file they control and that is usable only by its owner. Aliases in a `.mailrc` file adhere to the following format.

`alias` *aliasname value value value* `...`

aliasname is the name that the user uses when sending mail, and *value* is a valid email address.

If a user establishes a personal alias for `scott` that does not match the email address for `scott` in the name service, mail is routed to the wrong person when people try to reply to mail that is generated by this user. The only workaround is to use any of the other aliasing mechanisms.

`/etc/mail/aliases` File

Any alias that is established in the `/etc/mail/aliases` file can be used by any user who knows the name of the alias and the host name of the system that contains the file. Distribution list formats in a local `/etc/mail/aliases` file adhere to the following format.

aliasname`:` *value,value,value* `...`

aliasname is the name that the user uses when sending mail to this alias, and *value* is a valid email address.

If your network is not running a name service, the `/etc/mail/aliases` file of each system should contain entries for all mail clients. You can either edit the file on each system or edit the file on one system and copy it to each of the other systems.

The aliases in the /etc/mail/aliases file are stored in text form. When you edit the /etc/mail/aliases file, you need to run the newaliases program to recompile the database and make the aliases available in binary form to the sendmail program. For task information, refer to "How to Set Up a Local Mail Alias File" on page 327 in Chapter 24. Otherwise, you can use the Administration Tool's Database Manager to administer the mail aliases that are stored in the local /etc files.

You can create aliases for only local names—a current host name or no host name. For example, an alias entry for user ignatz who has a mailbox on the system saturn would have the following entry in the /etc/mail/aliases file.

```
ignatz: ignatz@saturn
```

You should create an administrative account for each mail server. You create such an account by assigning a mailbox on the mail server to root and by adding an entry for root to the /etc/mail/aliases file. For example, if the system saturn is a mailbox server, add the entry root: sysadmin@saturn to the /etc/mail/aliases file.

Normally, only the root user can edit this file. However, when you use the Administration Tool, all users in group 14, which is the sysadmin group, can change the local file. Another option is to create the following entry.

aliasname: :include:*/path/aliasfile*

aliasname is the name that the user uses when sending mail, and */path/aliasfile* is the full path to the file that contains the alias list. The alias file should include email entries, one entry on each line, and no other notations.

user1@host1
user2@host2

You can define additional mail files in /etc/mail/aliases to keep a log or a backup copy. The following entry stores all mail that is sent to *aliasname* in *filename*.

aliasname: /home/backup/*filename*

You can also route the mail to another process. The following example stores a copy of the mail message in *filename* and prints a copy.

aliasname: "|tee -a /home/backup/*filename* |lp"

For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

NIS aliases Map

All users in a local domain can use the entries that are in the NIS aliases map, because the sendmail program can use the NIS aliases map instead of the local

/etc/mail/aliases files to determine mailing addresses. For more information, refer to the nsswitch.conf(4) man page.

Aliases in the NIS aliases map adhere to the following format.

aliasname: *value,value,value* ...

aliasname is the name that the user uses when sending mail, and *value* is a valid email address.

The NIS aliases map should contain entries for all mail clients. In general, only the root user on the NIS master can change these entries. This type of alias might not be a good choice for aliases that are constantly changing, but such aliases can be useful if the aliases point to another alias file, as in the following syntax example.

aliasname: *aliasname@host*

aliasname is the name that the user uses when sending mail, and *host* is the host name for the server that contains an /etc/mail/alias file.

For task information, refer to "How to Set Up an NIS mail.aliases Map" on page 326 in Chapter 24.

NIS+ mail_aliases Table

The NIS+ mail_aliases table contains the names by which a system or person is known in the local domain. The sendmail program can use the NIS+ mail_aliases table, instead of the local /etc/mail/aliases files, to determine mailing addresses. Refer to the aliasadm(1M) and nsswitch.conf(4) man pages for more information.

Aliases in the NIS+ mail_aliases table adhere to the following format:

alias: *expansion* # ["*options* " # "*comments*"]

Table 25–12 describes the four columns that are in an NIS+ mail_aliases table.

TABLE 25–12 Columns in the NIS+ mail_aliases Table

Column	Description
alias	The name of the alias
expansion	The value of the alias or a list of aliases as it would appear in a sendmail /etc/mail/aliases file
options	The column that is reserved for future use
comments	The column for comments about an individual alias

The NIS+ `mail_aliases` table should contain entries for all mail clients. You can list, create, modify, and delete entries in the NIS+ `aliases` table with the `aliasadm` command. To use the `aliasadm` command, you must be a member of the NIS+ group that owns the `aliases` table. For task information, refer to "How to Manage Alias Entries in an NIS+ `mail_aliases` Table" on page 321 in Chapter 24. Alternately, you can use the Administration Tool's Database Manager to administer the NIS+ mail aliases.

Note – If you are creating a new NIS+ `aliases` table, you must initialize the table before you create the entries. If the table exists, no initialization is needed.

`.forward` Files

Users can create a `.forward` file in their home directories that `sendmail`, along with other programs, can use to redirect mail or send mail. Refer to the following topics.

For a task map, refer to "Administering `.forward` Files (Task Map)" on page 335 in Chapter 24.

Situations to Avoid

The following list describes some situations that you can avoid or easily fix.

- If mail is not being delivered to the expected address, check the user's `.forward` file. The user might have put the `.forward` file in the home directory of `host1`, which forwards mail to `user@host2`. When the mail arrives at `host2`, `sendmail` looks up `user` in the NIS or NIS+ aliases and sends the message back to `user@host1`, which results in a loop and more bounced mail.
- To avoid security problems, never put `.forward` files in the `root` and `bin` accounts. If necessary, forward the mail by using the `aliases` file instead.

Controls for `.forward` files

For the `.forward` files to be an effective part of mail delivery, ensure that the following controls (mostly permissions settings) are correctly applied.

- The `.forward` file must be writable only by the owner of the file. This restriction prevents other users from breaking security.

- The paths that lead to the home directory must be owned and writable by `root` only. For example, if a `.forward` file is in `/export/home/terry`, `/export` and `/export/home` must be owned and writable by `root` only.
- The actual home directory should be writable only by the user.
- The `.forward` file cannot be a symbolic link, and it cannot have more than one hard link.

`.forward.`*hostname* File

You can create a `.forward.`*hostname* file to redirect mail that is sent to a specific host. For example, if a user's alias has changed from `sandy@phoenix.example.com` to `sandy@example.com`, place a `.forward.phoenix` file in the home directory for `sandy`.

```
% cat .forward.phoenix
sandy@example.com
"|/usr/bin/vacation sandy"
% cat .vacation.msg
From: sandy@example.com (via the vacation program)
Subject: my alias has changed

My alias has changed to sandy@example.com.
Please use this alias in the future.
The mail that I just received from you
has been forwarded to my new address.

Sandy
```

In this example, mail can be forwarded to the correct place while the sender is notified of the alias change. Because the `vacation` program permits only one message file, you can forward only one message at a time. However, if the message is not host specific, one vacation message file can be used by `.forward` files for many hosts.

`.forward+`*detail* File

Another extension to the forwarding mechanism is the `.forward+`*detail* file. The *detail* string can be any sequence of characters except operator characters. The operator characters are `.:%&!^[]+`. By using this type of file, you can determine if someone else is using your email address without your knowledge. For instance, if a user tells someone to use the email address `sandy+test1@example.com`, the user would be able to identify any future mail that was delivered to this alias. By default, any mail that is sent to the `sandy+test1@example.com` alias is checked against the alias and the `.forward+`*detail* files. If no matches are made, the mail falls back to delivery to `sandy@example.com`, but the user is able to see a change in the `To:` mail header.

/etc/default/sendmail File

This file is used to store startup options for `sendmail` so that they are not removed when a host is upgraded. The following variables can be used:

CLIENTOPTIONS="*string*"
: Selects additional options to be used with the client daemon, which looks in the client-only queue (`/var/spool/clientmqueue`) and acts as a client queue runner. No syntax checking is done, so be careful when making changes to this variable.

CLIENTQUEUEINTERVAL=#
: Similar to the `QUEUEINTERVAL` option, `CLIENTQUEUEINTERVAL` sets the time interval for mail queue runs. However, the `CLIENTQUEUEINTERVAL` option controls the functions of the client daemon, instead of the master daemon. Typically, the master daemon is able to deliver all messages to the SMTP port. However, if the message load is too high or the master daemon is not running, then messages go into the client-only queue, `/var/spool/clientmqueue`. The client daemon, which looks in the client-only queue, then acts as a client queue processor.

ETRN_HOSTS="*string*"
: Enables an SMTP client and server to interact immediately without waiting for the periodic queue run intervals. The server can immediately deliver the portion of its queue that goes to the specified hosts. For more information, refer to the `etrn`(1M) man page.

MODE=-bd
: Selects the mode to start `sendmail` with. Use the `-bd` option or leave it undefined.

OPTIONS=*string*
: Selects additional options to be used with the master daemon. No syntax checking is done, so be careful when making changes to this variable.

QUEUEINTERVAL=#
: Sets the interval for mail queue runs on the master daemon. # can be a positive integer that is followed by either `s` for seconds, `m` for minutes, `h` for hours, `d` for days, or `w` for weeks. The syntax is checked before `sendmail` is started. If the interval is negative or if the entry does not end with an appropriate letter, the interval is ignored and `sendmail` starts with a queue interval of 15 minutes.

QUEUEOPTIONS=p
: Enables one persistent queue runner that sleeps between queue run intervals, instead of a new queue runner for each queue run interval. You can set this option to `p`, which is the only setting available. Otherwise, this option is not set.

Mail Addresses and Mail Routing

The path a mail message follows during delivery depends on the setup of the client system and the topology of the mail domain. Each additional level of mail hosts or mail domains can add another alias resolution, but the routing process is basically the same on most hosts.

You can set up a client system to receive mail locally. Receiving mail locally is known as running `sendmail` in local mode. Local mode is the default for all mail servers and some clients. On a mail server or a mail client in local mode, a mail message is routed the following way.

Note – The following example assumes that you are using the default rule set in the `sendmail.cf` file.

---

1. **Expand the mail alias, if possible, and restart the local routing process.**

   The mail address is expanded by looking up the mail alias in the name service and substituting the new value, if one is found. This new alias is then checked again.

2. **If the mail is local, deliver it to `/usr/lib/mail.local`.**

   The mail is delivered to a local mailbox.

3. **If the mail address includes a host in this mail domain, deliver the mail to that host.**

4. **If the address does not include a host in this domain, forward the mail to the mail host.**

   The mail host uses the same routing process as the mail server, but the mail host can receive mail that is addressed to the domain name as well as to the host name.

Interactions of `sendmail` With Name Services

This section describes domain names as they apply to `sendmail` and name services, as well as the rules for effective use of name services, and the specific interactions of `sendmail` with name services. For details, refer to the following topics.

- "`sendmail.cf` and Mail Domains" on page 380
- "`sendmail` and Name Services" on page 380

- "Interactions of NIS and `sendmail`" on page 382
- "Interactions of `sendmail` With NIS and DNS" on page 382
- "Interactions of NIS+ and `sendmail`" on page 383
- "Interactions of `sendmail` With NIS+ and DNS" on page 384

If you are looking for related task information, refer to "How to Use DNS With `sendmail`" on page 316 or "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

`sendmail.cf` and Mail Domains

The standard `sendmail.cf` file uses mail domains to determine whether mail is delivered directly or through a mail host. Intradomain mail is delivered through a direct SMTP connection, while interdomain mail is forwarded to a mail host.

In a secure network, only a few selected hosts are authorized to generate packets that are targeted to external destinations. Even if a host has the IP address of the remote host that is external to the mail domain, the establishment of an SMTP connection is not guaranteed. The standard `sendmail.cf` assumes the following.

- The current host is not authorized to send packets directly to a host outside the mail domain.
- The mail host is capable of forwarding the mail to an authorized host that can transmit packets directly to an external host. In fact, the mail host can itself be an authorized host.

Given these assumptions, the mail host is responsible for delivering or forwarding interdomain mail.

`sendmail` and Name Services

`sendmail` imposes various requirements on name services. To improve your understanding of these requirements, this section first describes the relationship of mail domains to name service domains. Then it describes the various requirements. Refer to the following.

- "Mail Domains and Name Service Domains" on page 381
- "Host Name Service Data" on page 381
- Man pages for `in.named`(1M), `nis+`(1), `nisaddent`(1M), and `nsswitch.conf`(4)

Mail Domains and Name Service Domains

The mail domain name must be a suffix of the name service domain. For example, if the domain name of the name service is `A.B.C.D`, the mail domain name could be one of the following.

- `A.B.C.D`
- `B.C.D`
- `C.D`
- `D`

When first established, the mail domain name is often identical to the name service domain. As the network grows, the name service domain can be divided into smaller pieces to make the name service more manageable. However, the mail domain often remains undivided to provide consistent aliasing.

Host Name Service Data

This section describes the requirements that `sendmail` imposes on name services.

A host table or map in a name service must be set up to support three types of `gethostbyname()` queries.

- `mailhost` – Some name service configurations satisfy this requirement automatically.
- Full host name (for example, `smith.admin.acme.com`) – Many name service configurations satisfy this requirement.
- Short host name (for example, `smith`) – `sendmail` must connect to the mail host in order to forward external mail. To determine if a mail address is within the current mail domain, `gethostbyname()` is invoked with the full host name. If the entry is found, the address is considered internal.

  NIS, NIS+, and DNS support `gethostbyname()` with a short host name as an argument, so this requirement is automatically satisfied.

Two additional rules about the host name service need to be followed to establish efficient `sendmail` services within a name service.

- `gethostbyname()` with full host name argument and short host name argument should yield consistent results. For example, `gethostbyname(smith.admin.acme.com)` should return the same result as `gethostbyname(smith)`, as long as both functions are called from the mail domain `admin.acme.com`.
- For all name service domains under a common mail domain, `gethostbyname()` with a short host name should yield the same result. For example, if the mail domain `smith.admin.acme.com` is given, `gethostbyname(smith)` should return the same result when the call comes from either the `ebb.admin.acme.com`

domain or the `esg.admin.acme.com` domain. The mail domain name is usually shorter than the name service domain, which gives this requirement special implications for various name services.

For more information about the `gethostbyname()` function, refer to the `gethostbyname`(3NSL) man page.

Interactions of NIS and `sendmail`

The following list describes the interactions of `sendmail` and NIS and provides some guidance.

- **Mail domain name** – If you are setting up NIS as the primary name service, `sendmail` automatically strips the first component of the NIS domain name and uses the result as the mail domain name. For example, `ebs.admin.acme.com` becomes `admin.acme.com`.
- **Mail host name** – You must have a `mailhost` entry in the NIS host map.
- **Full host names** – The normal NIS setup does not "understand" the full host name. Rather than trying to make NIS understand the full host name, turn off this requirement from the `sendmail` side by editing the `sendmail.cf` file and replacing all occurrences of `%l` with `%y`. This change turns off `sendmail`'s interdomain mail detection. If the target host can be resolved to an IP address, a direct SMTP delivery is attempted. Ensure that your NIS host map does not contain any host entry that is external to the current mail domain. Otherwise, you need to further customize the `sendmail.cf` file.
- **Matching full and short host names** – Follow the previous instructions on how to turn off `gethostbyname()` for a full host name.
- **Multiple NIS domains in one mail domain** – All NIS host maps under a common mail domain should have the same set of host entries. For example, the host map in the `ebs.admin.acme.com` domain should be the same as the host map in the `esg.admin.acme.com`. Otherwise, one address might work in one NIS domain, but fail in the other NIS domain.

For task information, refer to "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

Interactions of `sendmail` With NIS and DNS

The following list describes the interactions of `sendmail` with NIS and DNS and provides some guidance.

- **Mail domain name** – If you are setting up NIS as the primary name service, `sendmail` automatically strips the first component of the NIS domain name and

uses the result as the mail domain name. For example, `ebs.admin.acme.com` becomes `admin.acme.com`.

- **Mail host name** – When the DNS forwarding feature is turned on, queries that NIS cannot resolve are forwarded to DNS, so you do not need a `mailhost` entry in the NIS host map.
- **Full host names** – Although NIS does not "understand" full host names, DNS does. This requirement is satisfied when you follow the regular procedure for setting up NIS and DNS.
- **Matching full and short host names** – For every host entry in the NIS host table, you must have a corresponding host entry in DNS.
- **Multiple NIS domains in one mail domain** – All NIS host maps under a common mail domain should have the same set of host entries. For example, the host map in the `ebs.admin.acme.com` domain should be the same as the host map in the `esg.admin.acme.com` domain. Otherwise, one address might work in one NIS domain, but fail in the other NIS domain.

For task information, refer to "How to Use DNS With `sendmail`" on page 316 and "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

Interactions of NIS+ and `sendmail`

The following list describes the interactions of `sendmail` with NIS+ and provides some guidance.

- **Mail domain name** – If you are setting up NIS+ as your primary name service, `sendmail` can check the mail domain from the NIS+ `sendmailvars` table, a two–column NIS+ table with one key column and one value column. To set up your mail domain, you must add one entry to this table. This entry should have the key column set to the literal string `maildomain` and the value column set to your mail domain name (for example, `admin.acme.com`). Although NIS+ allows any string in the `sendmailvars` table, the suffix rule still applies for the mail system to work correctly. You can use `nistbladm` to add the `maildomain` entry to the `sendmailvars` table. Notice in the following example that the mail domain is a suffix of the NIS+ domain.

```
nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
```

- **Mailhost host name** – You must have a `mailhost` entry in the NIS+ hosts table.
- **Full host names** – NIS+ "understands" the full host name. Following the regular NIS+ setup procedure satisfies this requirement.
- **Matching full and short host names** – To satisfy this requirement, you can duplicate the entries in the host table, or you can enter all host entries in the user name service domains into a master host table at mail domain level.

- **Multiple NIS domains in one mail domain** – To satisfy this requirement, you can duplicate the entries in all the host tables, or you can type all host entries in the user name service domains into a master host table at mail domain level. Because you are merging (logical or physical) multiple host tables into one host table, the same host name cannot be reused in the multiple name service domain that shares a common mail domain.

For task information, refer to "Administering Mail Alias Files (Task Map)" on page 320 in Chapter 24.

Interactions of `sendmail` With NIS+ and DNS

The following list describes the interactions of `sendmail` with NIS+ and DNS and provides some guidance.

- **Mail domain name** — If you are setting up NIS+ as your primary name service, `sendmail` can check the mail domain from the NIS+ `sendmailvars` table, a two–column NIS+ table with one key column and one value column. To set up your mail domain, you must add one entry to this table. This entry should have the key column set to the literal string `maildomain` and the value column set to the your mail domain name (for example, `admin.acme.com`). Although NIS+ allows any string in the `sendmailvars` table, the suffix rule still applies for the mail system to work correctly. You can use `nistbladm` to add the `maildomain` entry to the `sendmailvars` table. Notice in the following example that the mail domain is a suffix of the NIS+ domain.

```
nistbladm -A key="maildomain" value=<mail domain> sendmailvars.org_dir.<NIS+ domain>
```

- **Mailhost host name** — If your network uses both NIS+ and DNS as the source for the host database, you can put the `mailhost` entry in either the NIS+ or DNS host table. Ensure that your users include NIS+ and DNS as the source for the host database in the `/etc/nsswitch.conf` file.
- **Full host names** — Both NIS+ and DNS "understand" full host names. Following the regular NIS+ and DNS setup procedures satisfies this requirement.
- **Matching full and short host names** — For every host entry in the NIS+ host table, you must have a corresponding host entry in DNS.
- **Multiple NIS domains in one mail domain** — To satisfy this requirement, you can duplicate the entries in all the host tables, or you can type all host entries in the user name service domains into a master host table at the mail domain level.

For task information, refer to "Administering Mail Alias Files (Task Map)" on page 320 and "How to Use DNS With `sendmail`" on page 316 in Chapter 24.

CHAPTER 26

What's New With Mail Services (Reference)

Chapter 23 provides an introduction to the components of mail services and a description of a typical mail configuration. Chapter 24 explains how to set up and administer an electronic mail system with standard configuration files. Chapter 25 describes in greater detail the components of mail services. Chapter 25 also describes the mail service programs and files, the mail routing process, and the interactions of `sendmail` with name services. This chapter describes the new features that are included in version 8.12 of `sendmail`, the version that is in this Solaris 9 release. You can also read about changes to `mail.local`, `mailstats`, and `makemap`. This chapter also describes a new maintenance utility, `editmap`. The following list can help you in your search on a specific topic.

For details not covered in this chapter, you can look in the man pages for `sendmail`(1M), `mail.local`(1M), `mailstats`(1), `makemap`(1M), and `editmap`(1M).

Changes to `sendmail`

This section contains information on the following topics.

New Configuration File, submit.cf

Version 8.12 of sendmail includes an additional configuration file, /etc/mail/submit.cf. This new file, submit.cf, is used to run sendmail in mail submission program mode instead of daemon mode. Mail submission program mode, unlike daemon mode, does not require root privilege, so this new paradigm provides better security.

See the following list of functions for submit.cf:

- sendmail uses submit.cf to run in mail submission program (MSP) mode, which submits email messages and can be invoked by programs (such as, mailx), as well as by users. Refer to the description of the -Ac option and the -Am option in "New or Deprecated Command-Line Options" on page 388.
- submit.cf is used in the following operating modes:

  -bm Is the default operating mode

  -bs Uses standard input to run SMTP

  -bt Is the test mode that is used to resolve addresses
- sendmail, when using submit.cf, does not run as an SMTP daemon.
- sendmail, when using submit.cf, uses /var/spool/clientmqueue, the client-only mail queue, which holds messages that were not delivered to the sendmail daemon. Messages in the client-only queue are delivered by the client "daemon," which is really acting as a client queue-runner.
- By default, sendmail uses submit.cf periodically to run the MSP queue (otherwise, known as the client-only queue), /var/spool/clientmqueue.

```
/usr/lib/sendmail -Ac -q15m
```

Note the following:

- submit.cf is provided automatically when you install or upgrade to the Solaris 9 operating environment.
- submit.cf requires no planning or preliminary procedures prior to the installation of the Solaris 9 operating environment.

- Unless you specify a configuration file, `sendmail` automatically uses `submit.cf` as required. Basically, `sendmail` knows which tasks are appropriate for `submit.cf` and which tasks are appropriate for `sendmail.cf`.
- `submit.cf` is not to be modified.

Functions That Distinguish `sendmail.cf` From `submit.cf`

The `sendmail.cf` configuration file is for the daemon mode. When using this file, `sendmail` is acting as a mail transfer agent (MTA), which is started by `root`.

```
/usr/lib/sendmail -L sm-mta -bd -q1h
```

See the following list of other distinguishing functions for `sendmail.cf`:

- By default, `sendmail.cf` accepts SMTP connections on ports 25 and 587.
- By default, `sendmail.cf` runs the main queue, `/var/spool/mqueue`.

Functional Changes in `sendmail`

With the addition of `submit.cf`, the following functional changes have occurred:

- In version 8.12 of `sendmail`, only `root` can run the mail queue. For further details, refer to the changes that are described in the `mailq`(1) man page. For new task information, refer to "Administering the Queue Directories (Task Map)" on page 332.
- Because the mail submission program mode runs without `root` privilege, which might prevent `sendmail` from having access to certain files (such as, the `.forward` files), the `-bv` option for `sendmail` could give the user misleading output. No workaround is available.
- Prior to `sendmail` version 8.12, if you were not running the `sendmail` daemon (that is, running in daemon mode), you would only prevent the delivery of inbound mail. Now, in `sendmail` version 8.12, if you are not running the `sendmail` daemon with the default configuration, you also prevent the delivery of outbound mail. The client queue-runner (also known as the mail submission program) must be able to submit mail to the daemon on the local SMTP port. If the client queue-runner tries to open an SMTP session with the local host and the daemon is not listening on the SMTP port, the mail remains in the queue. The default configuration **does** run a daemon, so this problem does not occur if you are using the default configuration. However, if you have disabled your daemon, refer to "Managing Mail Delivery by Using an Alternate Configuration (Task)" on page 319 for a way to resolve this problem.

New or Deprecated Command-Line Options

The following table describes new command-line options for `sendmail`. Other command-line options are described in the `sendmail`(1M) man page.

TABLE 26–1 New Command-Line Options for `sendmail`

Option	Description
`-Ac`	Indicates that you want to use the configuration file, `submit.cf`, even if the operation mode does not indicate an initial mail submission. For more information about `submit.cf`, refer to "New Configuration File, `submit.cf`" on page 386.
`-Am`	Indicates that you want to use the configuration file, `sendmail.cf`, even if the operation mode indicates an initial mail submission. For more information, refer to "New Configuration File, `submit.cf`" on page 386.
`-bP`	Indicates that you are printing the number of entries in each queue.
`-G`	Indicates that the message that is being submitted from the command line is for relaying, not for initial submission. The message is rejected if the addresses are not fully qualified. No canonicalization is done. As is noted in the Release Notes that are part of the `sendmail` distribution on ftp://ftp.sendmail.org, improperly formed messages might be rejected in future releases.
`-L` *tag*	Sets the identifier that is used for syslog messages to the supplied *tag*.
`-q[!]I` *substring*	Processes only jobs that contain this *substring* of one of the recipients. When `!` is added, the option processes only jobs that do not have this *substring* of one of the recipients.
`-q[!]R` *substring*	Processes only jobs that contain this *substring* of the queue ID. When `!` is added, the option processes only jobs that do not have this *substring* of the queue ID.
`-q[!]S` *substring*	Processes only jobs that contain this *substring* of the sender. When `!` is added, the option processes only jobs that do not have this *substring* of the sender.
`-qf`	Processes saved messages in the queue once, without using the `fork` system call, and runs the process in the foreground. Refer to the `fork`(2) man page.
`-qG`*name*	Processes only the messages in the *name* queue group.
`-qp`*time*	Processes saved messages in the queue at a specific interval of time with a single child that is forked for each queue. The child sleeps between queue runs. This new option is similar to the `-q`*time*, which periodically forks a child to process the queue.
`-U`	As is noted in the Release Notes that are part of the `sendmail` distribution on ftp://ftp.sendmail.org, this option is not available in version 8.12. Mail user agents should use the `-G` argument.

New and Revised Configuration File Options and Related Topics

This section contains a table of new and revised configuration file options and information on the following related topics.

- "Deprecated and Unsupported Configuration File Options for `sendmail`" on page 396
- "New `ClientPortOptions` Option" on page 397
- "Changes to `DaemonPortOptions` Option" on page 398
- "Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options" on page 400
- "Changes to the `PrivacyOptions` Option" on page 400
- "Changes to the `Timeout` Option" on page 401

When you declare these options, use one of the following syntaxes.

```
O OptionName=argument            # for the configuration file
-OOptionName=argument            # for the command line
define(`m4Name',argument)        # for m4 configuration
```

If you need to build a new `sendmail.cf` file, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24.

The following table describes new and revised options for `sendmail`.

TABLE 26–2 New and Revised Options for `sendmail`

Option	Description
`BadRcptThrottle`	`m4` name: `confBAD_RCPT_THROTTLE` Argument: *number* The new option limits the rate that recipients in the SMTP envelope are accepted after a threshold number of recipients has been rejected.
`ClientPortOption`	For details, see "New `ClientPortOptions` Option" on page 397.
`ConnectionRateThrottle`	`m4` name: `confCONNECTION_RATE_THROTTLE` Argument: *number* The option `ConnectionRateThrottle` now limits the number of connections per second to each daemon, not the total number of connections.

TABLE 26–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`ControlSocketName`	`m4` name: `confCONTROL_SOCKET_NAME`
	Argument: *filename*. The recommended socket name is `/var/spool/mqueue/.smcontrol`. For security, this UNIX domain socket must be in a directory that is accessible only by `root`.
	When it is set, this new option creates a control socket for daemon management. This option enables an external program to control and query the status of the running `sendmail` daemon by way of a named socket. The socket is similar to the `ctlinnd` interface to the `INN` news server. If this option is not set, no control socket is available.
`DaemonPortOptions`	For details, see "Changes to `DaemonPortOptions` Option" on page 398.
`DataFileBufferSize`	`m4` name: `confDF_BUFFER_SIZE`
	Argument: *number*
	The new option controls the maximum size (in bytes) of a memory-buffered data (`df`) file before a disk-based file is used. The default is `4096` bytes. You should not have to change the default for the Solaris operating environment.
`DeadLetterDrop`	`m4` name: `confDEAD_LETTER_DROP`
	Argument: *filename*
	The new option, which you should not need to set, defines the location of the system-wide `dead.letter` file, which was formerly hard-coded to `/usr/tmp/dead.letter`.
`DelayLA`	`m4` name: `confDELAY_LA`
	Argument: *number*
	If this new option is set to a value greater than zero, the option does the following:
	Delays connections by one second when the load averages exceed a specified value
	Delays the execution of most SMTP commands by one second
	Otherwise, if the option is not set, the default value, which is zero, does not change the behavior of `sendmail`.

TABLE 26–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`DeliverByMin`	`m4` name: `confDELIVER_BY_MIN`
	Argument: *time*
	The new option enables a client to specify a minimum amount of time for an email message to be delivered, as specified in RFC 2852, Deliver By SMTP Service Extension.
	If the time is set to zero, no time is listed.
	If the time is set to less than zero, the extension is not offered.
	If the time is set to greater than zero, the extension is listed as the minimum time for the `EHLO` keyword, `DELIVERBY`.
`DirectSubmissionModifiers`	`m4` name: `confDIRECT_SUBMISSION_MODIFIERS`
	Argument: *modifiers*
	The new option defines `${daemon_flags}` for direct (command-line) submissions. If this option is not set, the value of `${daemon_flags}` is either `CC f`, if the option `-G` is used, or `c u`.
`DontBlameSendmail`	You can use the following new arguments.
	The argument, `NonRootSafeAddr`, has been added. When `sendmail` does not have enough privileges to run a `.forward` program or deliver to a file as the owner of that file, addresses are marked unsafe. Furthermore, if `RunAsUser` is set, users cannot use programs or deliver to files in their `.forward` programs. Use `NonRootSafeAddr` to resolve these problems.
`DoubleBounceAddress`	`m4` name: `confDOUBLE_BOUNCE_ADDRESS`
	Argument: *address*. The default is `postmaster`.
	If an error occurs when `sendmail` is sending an error message, `sendmail` sends the "double-bounced" error message to the address that is specified by the argument to this option.
`FallBackMXhost`	`m4` name: `confFALLBACK_MX`
	Argument: fully qualified domain name.
	This option now includes MX record lookups. To use the old behavior of no MX record lookups, you must put the name in square brackets.

TABLE 26–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`FastSplit`	`m4` name: `confFAST_SPLIT`
	Argument: *number*. The default value is one.
	This new option does the following:
	If the option is set to a value greater than zero, the initial MX lookups on addresses are suppressed when they are sorted, which might result in faster envelope splitting.
	If the mail is submitted from the command line, the value can limit the number of processes that are used to deliver the envelopes.
	If more envelopes are created, they are put in the queue and must be resolved with a queue run.
`LDAPDefaultSpec`	`m4` name: `confLDAP_DEFAULT_SPEC`
	Argument: Class switch with appropriate definition (for example, `-h`*host*, `-p`*port*, `-d`*bind DN*).
	The new option allows a default map specification for LDAP maps. The assigned default settings are used for all LDAP maps unless other individual map specifications are made with the `K` command. Set this option before defining any LDAP maps.
`MailboxDatabase`	`m4` name: `confMAILBOX_DATABASE`
	Argument: `pw`, which uses `getpwnam()`, is the default value. No other values are supported.
	The new option specifies the type of mailbox database that is used to look up local recipients.
`MaxHeadersLength`	`m4` name: `confMAX_HEADERS_LENGTH`
	Argument: *number*
	This option specifies a maximum length for the sum of all headers and can be used to prevent a denial-of-service attack. The default is 32768. A warning is issued if a value less than 16384 is used. You should not have to change the default value for the Solaris operating environment.
`MaxMimeHeaderLength`	`m4` name: `confMAX_MIME_HEADER_LENGTH`
	Argument: *number*
	This option sets the maximum length of certain MIME header field values to *x* number of characters. Also, for parameters within headers, you can specify a maximum length of *y*. The combined values look like *x*/*y*. If *y* is not specified, half of *x* is used. If no values are set, the default is `0`, which means no checks are made. This option is intended to protect mail user agents from buffer-overflow attacks. The suggested values are in the range of 256/128 to 1024/256. A warning is issued if values less than 128/40 are used.

TABLE 26–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`MaxQueueChildren`	`m4` name: `confMAX_QUEUE_CHILDREN` Argument: *number* This new option limits the number of concurrently active queue runner processes to the number that is specified in the argument. The option helps to limit the system resources that are used when the queue is processed. When the total number of queue runners for multiple queue groups exceeds the defined argument, the remaining queue groups are run later.
`MaxRecipientsPerMessage`	`m4` name: `confMAX_RCPTS_PER_MESSAGE` Argument: *number* If it is set, this option allows no more than the specified number of recipients in an SMTP envelope. The minimum argument is `100`. You can still declare this option from both the command line and the configuration file. However, normal users can now set it from the command line to enable the override of messages that are submitted through `sendmail -bs`. In this instance, `sendmail` does not relinquish its `root` privileges.
`MaxRunnersPerQueue`	`m4` name: `confMAX_RUNNERS_PER_QUEUE` Argument: *number*. The default is one. Consider your resources carefully and do not set this value too high. This new option specifies the maximum number of queue runners per queue group. The queue runners work in parallel on a queue group's messages, which is useful when the processing of a message might delay the processing of subsequent messages.
`NiceQueueRun`	`m4` name: `confNICE_QUEUE_RUN` Argument: *number* This new option sets the priority of queue runners. Refer to the `nice`(1) man page.
`PidFile`	`m4` name: `confPID_file` Argument: See "Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options" on page 400. This new option defines the location of the `pid` file. The file name is macro-expanded before it is opened. The default is `/var/run/sendmail.pid`.
`PrivacyOptions`	For details, see "Changes to the `PrivacyOptions` Option" on page 400.

TABLE 26–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`ProcessTitlePrefix`	`m4` name: `confPROCESS_TITLE_PREFIX` Argument: See "Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options" on page 400. The new option specifies a prefix string for the process title that is shown in `/usr/ucb/ps auxww` listings. The string is macro processed. You should not have to make any changes for the Solaris operating environment.
`QueueFileMode`	`m4` name: `confQUEUE_FILE_MODE` Argument: *number* This new option provides the default permissions in octal for queue files. If this option is not set, `sendmail` uses `0600`. However, if the option's real and effective user ID is different, `sendmail` uses `0644`.
`QueueLA`	`m4` name: `confQUEUE_LA` Argument: *number* The default value has changed from eight to eight times the number of processors online when the system starts. For single-processor machines, this change has no effect. Changing this value overrides the default and prevents the number of processors from being considered. Therefore, the effect of any value changes should be well understood.
`QueueSortOrder`	`m4` name: `confQUEUE_SORT_ORDER` This option sets the algorithm that is used for sorting the queue. The default value is `priority`, which sorts the queue by message priority. Note the following changes. The `host` argument now reverses the host name before sorting, which means domains are grouped to run through the queue together. This improvement provides better opportunities for use of the connection cache, if available. The new `filename` argument sorts the queue by file name, which avoids the opening and reading of each queue file when preparing to run the queue. The new `modification` argument sorts the queue by time of modification, starting with the oldest entries of the `qf` file. The new `random` argument sorts the queue randomly, which avoids contention, if several queue runners have manually been started. For more information, refer to `QueueSortOrder` in the `sendmail`(1M) man page.

TABLE 26–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`RefuseLA`	`m4` name: `confREFUSE_LA` Argument: *number* The default value has changed from 12 to 12 times the number of processors online when the system starts. For single-processor machines, this change has no effect. A change of this value overrides the default and prevents the number of processors from being considered. Therefore, the effect of any value changes should be well understood.
`ResolverOptions`	Two changes have been made. When attempting to canonify a host name, some name servers that are down return a temporary failure message, `SERVFAIL`, for IPv6 `T_AAAA` lookups. You can use this new argument, `WorkAroundBrokenAAAA`, to avoid this behavior. Also, the `RES_USE_INET6` argument is controlled by a new flag, `use_inet6`. For more information, refer to the `resolver`(3RESOLV) man page.
`RrtImpliesDsn`	`m4` name: `confRRT_IMPLIES_DSN` Argument: `true` or `false` If the new option is set, a "Return-Receipt-To:" header causes the request of a delivery status notification (DSN), which is sent to the envelope sender, not to the address that is specified in the header.
`SendMimeErrors`	`m4` name: `confMIME_FORMAT_ERRORS` Argument: `true` or `false` The default is now `true`.
`SharedMemoryKey`	`m4` name: `confSHARED_MEMORY_KEY` Argument: *number* This new option permits you to use shared memory, if it is available, to store free space for queue file systems. This option minimizes the number of system calls to check for available space.
`SuperSafe`	`m4` name: `confSAFE_QUEUE` Argument: `true`, `false`, or `interactive`. The default and recommended value is `true`. Avoid using `false`. If this option is set to `true`, the queue file is always instantiated, even if you are attempting immediate delivery. You can use the `interactive` value together with `DeliveryMode=i` to skip some synchronization calls that are doubled in the code execution path for this mode.
`Timeout`	For details, see "Changes to the `Timeout` Option" on page 401.

TABLE 26–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`TrustedUser`	`m4` name: `confTRUSTED_USER` Argument: *user name* or *user numeric ID* The new option enables you to specify a user name (instead of `root`) to own important files. If this option is set, generated alias databases and the control socket—if configured—are automatically owned by this user. This option requires `HASFCHOWN`. For information about `HASFCHOWN`, see "Flags Used and Not Used to Compile `sendmail`" on page 348. Only `TrustedUser`, `root`, and class `t` (`$=t`) users can rebuild the alias map.
`UseMSP`	`m4` name: `confUSE_MSP` Argument: `true` or `false`. The default is `false`. This new option permits group writable queue files, if the group is the same as that of a set-group-id `sendmail` binary. In `submit.cf`, this option must be set to `true`.
`XscriptFileBufferSize`	`m4` name: `confXF_BUFFER_SIZE` Argument: *number* The new option controls the maximum size (in bytes) of a memory-buffered transcript (`xf`) file before a disk-based file is used. The default is `4096` bytes. You should not have to change this default for the Solaris operating environment.

Deprecated and Unsupported Configuration File Options for `sendmail`

Refer to the following table for a list of deprecated configuration file options. The table includes the `AutoRebuildAliases` option, which is not in version 8.12 of `sendmail`.

TABLE 26–3 Deprecated and Unsupported Configuration File Options for `sendmail`

Option	Description
`AutoRebuildAliases`	Because a denial-of-service attack could occur if this option is set, this option is not in version 8.12 of `sendmail`. Refer to the Release Notes that are part of the `sendmail` distribution at ftp://ftp.sendmail.org. A user could `kill` the `sendmail` process while the aliases file is being rebuilt and leave the file in an inconsistent state. Furthermore, because `AutoRebuildAliases` is not available, `newaliases` must be run manually now in order for changes to `/etc/mail/aliases` to take effect. Also, you must remember that because `sendmail` is no longer `setuid root`, only `root` can run `newaliases`.
`MeToo`	This option, which now defaults to `True`, has been deprecated. Refer to the Release Notes that are part of the `sendmail` distribution at ftp://ftp.sendmail.org.
`UnsafeGroupWrites`	This option is deprecated. If required, you should now use the `GroupWritableForwardFileSafe` and `GroupWritableIncludeFileSafe` arguments for the `DontBlameSendmail` option.
`UseErrorsTo`	This option is deprecated. Furthermore, because this option violates RFC 1123, you should avoid using it.

New `ClientPortOptions` Option

The new `ClientPortOptions` option is for outgoing connections and is similar to the `DaemonPortOptions` option. This option sets the client SMTP options, which are a sequence of *key=value* pairs. To declare this option, use one of the following syntaxes. For formatting purposes, the example includes two pairs. However, you can apply one or more pairs.

```
O ClientPortOptions=pair,pair              # for the configuration file
-OClientPortOptions=pair,pair              # for the command line
define('confCLIENT_OPTIONS','pair,pair')   # for m4 configuration
```

If you need to build a new `sendmail.cf` file, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24.

The following table describes the new keys for this option.

TABLE 26–4 New Keys for `ClientPortOptions`

Key	Description
`Addr`	Specifies the address mask. The value can be a numeric address in dot notation or a network name. If the pair is omitted, the default is `INADDR_ANY`, which accepts connections from any network.
`Family`	Specifies the address family. The key's default is `inet` for `AF_INET`. Other values are `inet6` for `AF_INET6`, `iso` for `AF_ISO`, `ns` for `AF_NS`, and `x.25` for `AF_CCITT`.
`Listen`	Specifies the size of the listen queue. The key defaults to `10`. You should not have to change this default for the Solaris operating environment.
`Port`	Specifies the name and number of the listening port. The key defaults to `smtp`.
`RcvBufSize`	Specifies the size of the TCP/IP send buffer. The key has no default value, which means that no size specifications are automatically made. If the option is set to a value greater than zero, that value is used. You should not have to limit the size of this buffer for the Solaris operating environment.
`Modifier`	Specifies flags for `sendmail`: The `h` flag uses the name that corresponds to the outgoing interface address for the `HELO` or `EHLO` commands, whether it was chosen by the connection parameter or by the default. The `A` flag disables `AUTH`. This flag can also be used with the `Modifier` key for `DaemonPortOptions`. Refer to "Changes to `DaemonPortOptions` Option" on page 398. The `S` flag turns off the use of or the offer to use `STARTTLS` when email is being delivered or received.

Changes to `DaemonPortOptions` Option

The following tables describe the new features.

- Table 26–5
- Table 26–6

To declare this option, use one of the following syntaxes. In the example, *pair* refers to *key=value*. For formatting purposes, the example includes two pairs. However, you can apply one or more pairs.

```
O DaemonPortOptions=pair,pair              # for the configuration file
-ODaemonPortOptions=pair,pair              # for the command line
define('confDAEMON_OPTIONS','pair,pair')   # for m4 configuration
```

Note – To avoid security risks, `sendmail` relinquishes its root permissions when you set this option from the command line.

If you need to build a new `sendmail.cf` file, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24.

The following table describes new and revised keys for the `DaemonPortOptions` option.

TABLE 26–5 New and Revised Keys for `DaemonPortOptions`

Key	Description
`Name`	A new key that specifies a user-definable name for `sendmail`. This key is used for error messages and for logging. The default is `MTA`.
`Modifier`	A new key that specifies values for `sendmail` that can be listed in a sequence without delimiters. For a list of values, see Table 26–6.
`Family`	Unless a `Family` is specified in a `DaemonPortOptions` option, `inet` is now the only default. If IPv6 users also want to listen on IPv6 interfaces, they can configure additional sockets into `sendmail.cf` by adding a `Family=inet6` setting to a `DaemonPortOptions` option.

The following table describes the values for the new `Modifier` key.

TABLE 26–6 Values for the New `Modifier` Key

Value	Description
`A`	Disables `AUTH` by overriding the `Modifier` value of `a`.
	Can be used with the `Modifier` key for `ClientPortOptions`. Refer to "New `ClientPortOptions` Option" on page 397.
`C`	Does not perform host-name canonification.
`E`	Disallows the `ETRN` command.
`O`	Ignores the socket if a failure should occur.
`S`	Turns off the use or the offer to use `STARTTLS` when email is being delivered or received.
	Can be used with the `Modifier` key for `ClientPortOptions`.
`a`	Requires authentication.
`b`	Binds to the interface that receives the mail.

TABLE 26–6 Values for the New `Modifier` Key *(Continued)*

Value	Description
`c`	Performs host-name canonification. Use this value only in configuration file declarations.
`f`	Requires fully qualified host names. Use this value only in configuration file declarations.
`h`	Uses the interface's name for the outgoing `HELO` command.
`u`	Allows unqualified addresses. Use this value only in configuration file declarations.

Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options

The following table describes additional macro-processed arguments for the `PidFile` and `ProcessTitlePrefix` options. For more information about these options, see Table 26–2.

TABLE 26–7 Arguments for the `PidFile` and `ProcessTitlePrefix` Options

Macro	Description
`${daemon_addr}`	Provides daemon address (for example, 0.0.0.0)
`${daemon_family}`	Provides daemon family (for example, `inet`, and `inet6`)
`${daemon_info}`	Provides daemon information (for example, SMTP+queueing@00:30:00)
`${daemon_name}`	Provides daemon name (for example, MSA)
`${daemon_port}`	Provides daemon port (for example, 25)
`${queue_interval}`	Provides queue run interval (for example, 00:30:00)

Changes to the `PrivacyOptions` Option

New and revised arguments for `PrivacyOptions` (`popt`) are described in the following table. You can declare this option from the command line without `sendmail` relinquishing its `root` privilege. To declare this `sendmail` option, use one of the following syntaxes.

```
O PrivacyOptions=argument                # for the configuration file
-OPrivacyOptions=argument                # for the command line
define('confPRIVACY_FLAGS','argument')   # for m4 configuration
```

If you need to build a new `sendmail.cf` file, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317, in Chapter 24.

The following table provides descriptions of new and revised arguments for the `PrivacyOptions` option.

TABLE 26–8 New and Revised Arguments for `PrivacyOptions`

Argument	Description
`goaway`	This argument no longer accepts the following flags: `noetrn`, `restrictmailq`, `restrictqrun`, `restrictexpand`, `nobodyreturn`, and `noreceipts`.
`nobodyreturn`	This argument instructs `sendmail` not to include the body of the original message in delivery status notifications.
`noreceipts`	When this argument is set, delivery status notification (DSN) is not announced.
`restrictexpand`	This argument instructs `sendmail` to drop privileges when the `-bv` option is given by users who are neither `root` nor `TrustedUser`. The users cannot read private aliases, `.forward` files, or `:include:` files. This argument also overrides the `-v` command-line option.

Changes to the `Timeout` Option

The following table provides information about the changes to the `Timeout` option. Specifically, this `sendmail` option has some new keywords and a new value for `ident`. In the Solaris operating environment, you should not need to change the default values for the keywords that are listed in the table. However, if you choose to make a change, use the *keyword=value* syntax. The *value* is a time interval. Refer to the following examples.

```
O Timeout.keyword=value    # for the configuration file
-OTimeout.keyword=value    # for the command line
define('m4_name', value) # for m4 configuration
```

If you need to build a new `sendmail.cf` file, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24.

Note – To avoid security risks, `sendmail` relinquishes its root permissions when you set this option from the command line.

TABLE 26–9 New and Revised Settings for `Timeout`

Keyword	Default Value	Description
`aconnect`	`0`	`m4` name: `confTO_ACONNECT` Limits the total time to wait for all connections to succeed for a single delivery attempt. The maximum value is unspecified.
`control`	`2m`	`m4` name: `confTO_CONTROL` Limits the total time that is dedicated to completing a control socket request.
`ident`	`5s`	`m4` name: `confTO_IDENT` Defaults to 5 seconds—instead of 30 seconds—to prevent the common delays that are associated with mailing to a site that drops `IDENT` packets. No maximum value is specified.
`lhlo`	`2m`	`m4` name: `confTO_LHLO` Limits the time to wait for a reply from an LMTP `LHLO` command. No maximum value is specified.
`queuereturn`	`5d`	`m4` name: `confTO_QUEUERETURN` Includes the value `now`, which immediately bounces entries from the queue without a delivery attempt.
`resolver.retrans`	*varies*	`m4` name: `confTO_RESOLVER_RETRANS` Sets the resolver's retransmission time interval (in seconds), which applies to `resolver.retrans.first` and `resolver.retrans.normal`.
`resolver.retrans.first`	*varies*	`m4` name: `confTO_RESOLVER_RETRANS_FIRST` Sets the resolver's retransmission time interval (in seconds) for the first attempt to deliver a message.
`resolver.retrans.normal`	*varies*	`m4` name: `confTO_RESOLVER_RETRANS_NORMAL` Sets the resolver's retransmission time interval (in seconds) for all resolver lookups, except the first delivery attempt.
`resolver.retry`	*varies*	`m4` name: `confTO_RESOLVER_RETRY` Sets the number of times to retransmit a resolver query, which applies to `Timeout.resolver.retry.first` and `Timeout.resolver.retry.normal` .

TABLE 26–9 New and Revised Settings for `Timeout` *(Continued)*

Keyword	Default Value	Description
`resolver.retry.first`	*varies*	`m4` name: `confTO_RESOLVER_RETRY_FIRST` Sets the number of times to retransmit a resolver query for the first attempt to deliver a message.
`resolver.retry.normal`	*varies*	`m4` name: `confTO_RESOLVER_RETRY_NORMAL` Sets the number of times to retransmit a resolver query for all resolver lookups, except the first delivery attempt.

New Defined Macros for `sendmail`

The following table describes new macros that are reserved for use by the `sendmail` program. The macros' values are assigned internally. For more information, refer to the `sendmail`(1M) man page.

TABLE 26–10 Defined Macros for `sendmail`

Macro	Description
`${addr_type}`	Identifies the current address as an envelope sender or a recipient address.
`${client_resolve}`	Holds the result of the resolve call for `${client_name}`: `OK`, `FAIL`, `FORGED`, or `TEMP`.
`${deliveryMode}`	Specifies the current delivery mode `sendmail` is using instead of the value of the `DeliveryMode` option.
`${dsn_notify}`, `${dsn_envid}`, `${dsn_ret}`	Holds the corresponding DSN parameter values.
`${if_addr}`	Provides the interface's address for the incoming connection if the interface does not belong to the loopback net. This macro especially useful for virtual hosting.

TABLE 26–10 Defined Macros for `sendmail` *(Continued)*

Macro	Description
`${if_addr_out}`, `${if_name_out}`, `${if_family_out}`	Avoids the reuse of `${if_addr}`. Holds the following values, respectively. The address of the interface for the outgoing connection The host name of the interface for the outgoing connection The family of the interface for the outgoing connection
`${if_name}`	Provides the interface's host name for the incoming connection and is especially useful for virtual hosting.
`${load_avg}`	Checks and reports the current average number of jobs in the run queue.
`${msg_size}`	Holds the value of the message size (`SIZE=parameter`) in an ESMTP dialogue before the message has been collected. Thereafter, the macro holds the message size as computed by `sendmail` and is used in `check_compat`. For information about `check_compat`, refer to Table 26–14.
`${nrcpts}`	Holds the number of validated recipients.
`${ntries}`	Holds the number of delivery attempts.
`${rcpt_mailer}`, `${rcpt_host}`, `${rcpt_addr}`, `${mail_mailer}`, `${mail_host}`, `${mail_addr}`	Holds the results of parsing the `RCPT` and `MAIL` arguments—that is, the resolved right-hand side (RHS) triplet from the mail delivery agent (`$#`*mailer*), the host (`$@`*host*), and the user (`$:`*addr*).

New Macros Used to Build the `sendmail` Configuration File

In this section, you can find the following.

TABLE 26–11 New Macros Used to Build the `sendmail` Configuration File

Macro	Description
`LOCAL_MAILER_EOL`	Overrides the default end-of-line string for the local mailer.
`LOCAL_MAILER_FLAGS`	Adds `Return-Path:` header by default.
`MAIL_SETTINGS_DIR`	Contains the path (including the trailing slash) for the mail settings directory.
`MODIFY_MAILER_FLAGS`	Improves the `*_MAILER_FLAGS`. This macro sets, adds, or deletes flags.
`RELAY_MAILER_FLAGS`	Defines additional flags for the relay mailer.

New `MAX` Macros

Use the following new macros to configure the maximum number of commands that can be received before `sendmail` slows its delivery. You can set these `MAX` macros at compile time. The maximum values in the following table also represent the current default values.

TABLE 26–12 New `MAX` Macros

Macro	Maximum Value	Commands Checked by Each Macro
`MAXBADCOMMANDS`	25	Unknown commands
`MAXNOOPCOMMANDS`	20	`NOOP`, `VERB`, `ONEX`, `XUSR`
`MAXHELOCOMMANDS`	3	`HELO`, `EHLO`
`MAXVRFYCOMMANDS`	6	`VRFY`, `EXPN`
`MAXETRNCOMMANDS`	8	`ETRN`

Note – You can disable a macro's check by setting the macro's value to zero.

New and Revised `m4` Configuration Macros for `sendmail`

This section contains a table of new and revised `m4` configuration macros for `sendmail`. Use the following syntax to declare these macros.

symbolic_name(‘*value*’)

If you need to build a new sendmail.cf file, refer to “Building the sendmail.cf Configuration File (Task)” on page 317 in Chapter 24.

TABLE 26–13 New and Revised m4 Configuration Macros for sendmail

m4 **Macro**	**Description**
FEATURE()	For details, refer to “Changes to the FEATURE() Declaration” on page 406.
LOCAL_DOMAIN()	This macro adds entries to class w ($=w).
MASQUERADE_EXCEPTION()	A new macro that defines hosts or subdomains that cannot be masqueraded.
SMART_HOST()	This macro can now be used for bracketed addresses, such as user@[*host*].
VIRTUSER_DOMAIN() or VIRTUSER_DOMAIN_FILE()	When these macros are used, include $={VirtHost} in $=R. As a reminder, $=R is the set of host names that are allowed to relay.

Changes to the FEATURE() Declaration

Refer to the following tables for information about the specific changes to the FEATURE() declarations.

- Table 26–14
- Table 26–15

To use the new and revised FEATURE names, use the following syntax.

FEATURE(‘*name*’, ‘*argument*’)

If you need to build a new sendmail.cf file, refer to “Building the sendmail.cf Configuration File (Task)” on page 317, in Chapter 24.

TABLE 26–14 New and Revised FEATURE() Declarations

Name of FEATURE()	**Description**
compat_check	Argument: Refer to the example in the following paragraph. This new FEATURE() enables you to look for a key in the access map that consists of the sender address and the recipient address and is delimited by the following string, <@>. *sender@sdomain*<@>*recipient@rdomain* is an example.

TABLE 26–14 New and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
delay_checks	Argument: friend, which enables a spam-friend test, or hater, which enables spam-hater test. A new FEATURE() that delays all checks. By using FEATURE('delay_checks'), the rule sets check_mail and check_relay are not called when a client connects or issues a MAIL command, respectively. Instead, these rule sets are called by the check_rcpt rule set. For details, refer to the /usr/lib/mail/README file.
dnsbl	Argument: This FEATURE() accepts a maximum of two arguments: ■ DNS server name ■ Rejection message A new FEATURE() that you can include multiple times to check the return values for DNS lookups. Note that this FEATURE() enables you to specify the behavior of temporary lookup failures.
enhdnsbl	Argument: domain name A new FEATURE() that is an enhanced version of dnsbl, which enables you to check the return values for DNS lookups. For more information, refer to /usr/lib/mail/README.
generics_entire_domain	Argument: None A new FEATURE() that you can also use to apply genericstable to subdomains of $=G.
ldap_routing	Argument: For details, refer to the "Release Notes" in http://www.sendmail.org. A new FEATURE() that implements LDAP address routing.
local_lmtp	Argument: Path name of an LMTP-capable mailer. The default is mail.local, which is LMTP capable in this Solaris release. A FEATURE() that now sets the delivery status notification (DSN) diagnostic-code type for the local mailer to the proper value of SMTP.
local_no_masquerade	Argument: None A new FEATURE() that you can use to avoid masquerading for the local mailer.
lookupdotdomain	Argument: None A new FEATURE() that you can also use to look up the *.domain* in the access map.

TABLE 26–14 New and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
nocanonify	Argument: canonify_hosts or nothing
	A FEATURE() that now includes the following features.
	Enables a list of domains, as specified by CANONIFY_DOMAIN or CANONIFY_DOMAIN_FILE, to be passed to the $[and $] operators for canonification.
	Enables addresses that have only a host name, such as <user@host>, to be canonified, if canonify_hosts is specified as its parameter.
	Adds a trailing dot to addresses with more than one component.
no_default_msa	Argument: None
	A new FEATURE() that turns off sendmail's default setting from m4–generated configuration files to "listen" on several different ports, an implementation of RFC 2476.
nouucp	Argument: reject, which does not allow the ! token, or nospecial, which does allow the ! token.
	A FEATURE() that determines whether or not to allow the ! token in the local part of an address.
nullclient	Argument: None
	A FEATURE() that now provides the full rule sets of a normal configuration, allowing anti-spam checks to be performed.
preserve_local_plus_detail	Argument: None
	A new FEATURE() that enables you to preserve the +detail portion of the address when sendmail passes the address to the local delivery agent.
preserve_luser_host	Argument: None
	A new FEATURE() that enables you to preserve the name of the recipient host, if LUSER_RELAY is used.
queuegroup	Argument: None
	A new FEATURE() that enables you to select a queue group that is based on the full email address or on the domain of the recipient.
relay_mail_from	Argument: The *domain* is an optional argument.
	A new FEATURE() that allows relaying if the mail sender is listed as a RELAY in the access map and is tagged with the From: header line. If the optional *domain* argument is given, the domain portion of the mail sender is also checked.

TABLE 26–14 New and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
virtuser_entire_domain	Argument: None A FEATURE() that you can now use to apply $={VirtHost}, a new class for matching virtusertable entries that can be populated by VIRTUSER_DOMAIN or VIRTUSER_DOMAIN_FILE. FEATURE('virtuser_entire_domain') can also apply the class $={VirtHost} to entire subdomains.

The following FEATURE() declarations are no longer supported.

TABLE 26–15 Unsupported FEATURE() Declarations

Name of FEATURE()	Replacement
rbl	FEATURE('dnsbl') and FEATURE('enhdnsbl') replace this FEATURE() that has been removed.
remote_mode	MASQUERADE_AS('$S') replaces FEATURE('remote_mode') in /usr/lib/mail/cf/subsidiary.mc. $S is the SMART_HOST value in sendmail.cf.
sun_reverse_alias_files	FEATURE('genericstable').
sun_reverse_alias_nis	FEATURE('genericstable').
sun_reverse_alias_nisplus	FEATURE('genericstable').

Changes to the MAILER() Declaration

The MAILER() declaration specifies support for delivery agents. To declare a delivery agent, use the following syntax.

MAILER('*symbolic_name*')

Note the following changes.

- In this new version of sendmail, the MAILER('smtp') declaration now includes an additional mailer, dsmtp, which provides on-demand delivery by using the F=% mailer flag. The dsmtp mailer definition uses the new DSMTP_MAILER_ARGS, which defaults to IPC $h.
- Numbers for rule sets that are used by MAILERs have been removed. You now have no required order for listing your MAILERs except for MAILER('uucp'), which must follow MAILER('smtp') if uucp-dom and uucp-uudom are used.

For more information about mailers, refer to “Mailers” on page 352. If you need to build a new `sendmail.cf` file, refer to “Building the `sendmail.cf` Configuration File (Task)” on page 317 in Chapter 24.

New Delivery Agent Flags

The following table describes new delivery agent flags, which by default are not set. These single-character flags are Boolean. You can set or unset a flag by including or excluding it in the `F=` statement of your configuration file, as shown in the following example.

```
Mlocal,     P=/usr/lib/mail.local, F=lsDFMAw5:/|@qSXfmnz9, S=10/30, R=20/40,
Mprog,      P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/,
Msmtp,      P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990,
Mesmtp,     P=[IPC], F=mDFMuXa, S=11/31, R=21, E=\r\n, L=990,
Msmtp8,     P=[IPC], F=mDFMuX8, S=11/31, R=21, E=\r\n, L=990,
Mrelay,     P=[IPC], F=mDFMuXa8, S=11/31, R=61, E=\r\n, L=2040,
```

TABLE 26–16 New Mailer Flags

Flag	Description
`%`	Mailers that use this flag do not attempt delivery to the initial recipient of a message or to queue runs unless the queued message is selected by using an `ETRN` request or one of the following queue options: `-qI`, `-qR` , or `-qS` .
`1`	This flag disables the ability of the mailer to send null characters (for example, `\0`).
`2`	This flag disables the use of ESMTP and requires that SMTP be used, instead.
`6`	This flag enables mailers to strip headers to 7 bit.

New Equates for Delivery Agents

The following table describes new equates that you can use with the `M` delivery agent definition command. The following syntax shows you how to append new equates or new arguments to the equates that already exist in the configuration file.

Magent_name, *equate*, *equate*, ...

The following example includes the new `W=` equate, which specifies the maximum time to wait for the mailer to return after all data has been sent.

```
Msmtp, P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990, W=2m
```

When you modify the definition of a value for `m4` configuration, use the syntax that is provided in the following example.

```
define('SMTP_MAILER_MAXMSGS', '1000')
```

The preceding example places a limit of 1000 on the number of messages that are delivered per connection on an `smtp` mailer.

If you need to build a new `sendmail.cf` file, refer to "Building the `sendmail.cf` Configuration File (Task)" on page 317 in Chapter 24.

Note – Typically, you modify the equate definitions in the `mailer` directory only when you fine-tune.

TABLE 26–17 New Equates for Delivery Agents

Equate	Description
`/=`	Argument: Path to a directory Specifies a directory to apply `chroot()` to before the mailer program is executed
`m=`	Argument: Any of the following `m4` values that have previously been defined with the `define()` routine `SMTP_MAILER_MAXMSGS`, for the `smtp` mailer `LOCAL_MAILER_MAXMSGS`, for the `local` mailer `RELAY_MAILER_MAXMSGS`, for the `relay` mailer Limits the number of messages that are delivered per connection on an `smtp`, `local`, or `relay` mailer
`W=`	Argument: An increment of time Specifies the maximum time to wait for the return of the mailer after all data has been sent

New Queue Features

The following list provides details about new queue features.

- This release supports multiple queue directories. To use multiple queues, supply a `QueueDirectory` option value in the configuration file that ends with an asterisk (*), as is shown in the following example.

  ```
  O QueueDirectory=/var/spool/mqueue/q*
  ```

  The option value, `/var/spool/mqueue/q*`, uses all of the directories (or symbolic links to directories) that begin with "`q`" as queue directories. Do not change the queue directory structure while `sendmail` is running. Queue runs create a separate process for running each queue unless the verbose flag (`-v`) is used on a non-daemon queue run. The new items are randomly assigned to a queue.

- The new queue file-naming system uses file names that are guaranteed to be unique for 60 years. This system allows queue IDs to be assigned without complex file-system locking and simplifies the movement of queued items between queues.
- In version 8.12 of `sendmail`, only `root` can run the mail queue. For further details, refer to the changes that are described in the `mailq`(1) man page. For new task information, refer to "Administering the Queue Directories (Task Map)" on page 332.
- To accommodate envelope splitting, queue file names are now 15 characters long, rather than 14–characters long. File systems with a 14 character name limit are no longer supported.

For task information, refer to "Administering the Queue Directories (Task Map)" on page 332.

Changes for LDAP in `sendmail`

The following list describes changes in the use of the Lightweight Directory Access Protocol (LDAP) with `sendmail`.

- `LDAPROUTE_EQUIVALENT()` and `LDAPROUTE_EQUIVALENT_FILE()` permit you to specify equivalent host names, which are replaced by the masquerade domain name for LDAP routing lookups. For more information, refer to `/usr/lib/mail/README`.
- As noted in the Release Notes that are part of the `sendmail` distribution at ftp://ftp.sendmail.org, the LDAPX map has been renamed to LDAP. Use the following syntax for LDAP.

  `Kldap ldap` *options*
- This release supports the return of multiple values for a single LDAP lookup. Place the values to be returned in a comma-separated string with the `-v` option, as is shown.

  `Kldap ldap -v"`*mail*`,`*more_mail*`"`
- If no LDAP attributes are specified in an LDAP map declaration, all attributes that are found in the match are returned.
- This version of `sendmail` prevents commas in quoted key and value strings in the specifications of the LDAP alias file from breaking up a single entry into multiple entries.
- This version of `sendmail` has a new option for LDAP maps. The option, –V*separator* enables you to specify a separator, so that a lookup can return both an attribute and a value that are separated by the relevant *separator*.
- Instead of using the `%s` token to parse an LDAP filter specification, you can also use the new token, `%0`, to encode the key buffer. The `%0` token applies a literal meaning to LDAP special characters.

The following example shows how these tokens differ for a lookup on "*."

TABLE 26–18 Comparison of Tokens

LDAP Map Specification	Specification Equivalent	Result
`-k"uid=%s"`	`-k"uid=*"`	Matches any record with a user attribute
`-k"uid=%0"`	`-k"uid=\2A"`	Matches a user with the name "*"

The following table describes new LDAP map flags.

TABLE 26–19 New LDAP Map Flags

Flag	Description
`-1`	Requires a single match to be returned. If more than one match is returned, the results are the equivalent of no records being found.
`-r` *never* | *always* | *search* | *find*	Sets the LDAP alias dereference option.
`-Z` *size*	Limits the number of matches to return.

New Built-in Mailer Feature

The old `[TCP]` built-in mailer is not available. Use the `P=[IPC]` built-in mailer instead. The interprocess communications `([IPC])` built-in mailer now enables delivery to a UNIX domain socket on systems that support it. You can use this mailer with `LMTP` delivery agents that listen on a named socket. An example mailer might resemble the following.

```
Mexecmail, P=[IPC], F=lsDFMmnqSXzA5@/:|, E=\r\n,
S=10, R=20/40, T=DNS/RFC822/X-Unix, A=FILE /var/run/lmtpd
```

The first mailer argument in the `[IPC]` mailer is now checked for a legitimate value. The following table provides possible values for the first mailer argument.

TABLE 26–20 Possible Values for the First Mailer Argument

Value	Description
`A=FILE`	Use for UNIX domain socket delivery
`A=TCP`	Use for TCP/IP connections
`A=IPC`	Is no longer available as a first mailer argument

New Rule Sets

The following table lists the new rule sets and describes what they do.

TABLE 26–21 New Rule Sets

Set	Description
`check_eoh`	Correlates information that is gathered between headers and checks for missing headers. This rule set is used with the macro storage map and is called after all of the headers have been collected.
`check_etrn`	Uses the `ETRN` command (such as `check_rcpt` uses `RCPT`).
`check_expn`	Uses the `EXPN` command (such as `check_rcpt` uses `RCPT`).
`check_vrfy`	Uses the `VRFY` command (such as `check_rcpt` uses `RCPT`).

The following list describes new rule set features.

- Numbered rule sets are also named, but they can still be accessed by their numbers.
- The `H` header configuration file command allows for a default rule set to be specified for header checks. This rule set is called only if the individual header has not been assigned its own rule set.
- Comments in rule sets (that is, text within parentheses) are not removed if the configuration file version is nine or greater. For example, the following rule matches the input `token (1)`, but does not match the input `token`.

```
R$+ (1)         $@ 1
```

- `sendmail` accepts the SMTP `RSET` command even when it rejects commands because of TCP wrappers or the `check_relay` rule set.
- You receive a warning if you set the `OperatorChars` option multiple times. Also, do not set `OperatorChars` after the rule sets are defined.
- The name of the rule set, as well as its lines, are ignored if an invalid rule set is declared. The rule set lines are not added to `S0`.

Changes to Files

Note the following changes.

- The `helpfile` is now located in `/etc/mail/helpfile`. The old name (`/etc/mail/sendmail.hf`) has a symbolic link that points to the new name.
- The `trusted-users` file is now located in `/etc/mail/trusted-users`. During an upgrade, if the old name (`/etc/mail/sendmail.ct`) is detected, but not the new name, a hard link from the old name to the new name is created. Otherwise,

nothing is done. The default content is `root`.

- The `local-host-names` file is now located in `/etc/mail/local-host-names`. During an upgrade, if the old name (`/etc/mail/sendmail.cw`) is detected, but not the new name, a hard link from the old name to the new name is created. Otherwise, nothing is done. The default content is zero length.
- The new name for `/usr/lib/mail/cf/main-v7sun.mc` is `/usr/lib/mail/cf/main.mc`.
- The new name for `/usr/lib/mail/cf/subsidiary-v7sun.mc` is `/usr/lib/mail/cf/subsidiary.mc`.

IPv6 Addresses in Configuration

In version 8.12 of `sendmail`, IPv6 addresses that are used in configuration should be prefixed with the `IPv6:` tag to identify the address properly. If you are not identifying an IPv6 address, a prefix tag is not used. To see an example in a procedure, refer to "How to Set Up a Mail Host" on page 313.

Changes to `mail.local`

The following table describes the new command-line options for the `mail.local` program, which is used by `sendmail` as a delivery agent for local mail.

TABLE 26–22 New Command-Line Options for `mail.local`

Option	Description
`-7`	Prevents the Local Mail Transfer Protocol (LMTP) mode from advertising 8BITMIME support in the `LHLO` response
`-b`	Causes a permanent error instead of a temporary error if a mailbox exceeds its quota

`mail.local` is the default for LMTP mode. However, for this release, if you choose to use `mail.local` as the local delivery agent without being in LMTP mode, you need to do one of the following to set the `S` flag.

Use the following syntax for the configuration file.

```
MODIFY_MAILER_FLAGS('LOCAL', '+S')      # for the configuration file
```

Alternately, perform the following two steps for `m4` configuration.

```
define('MODIFY_MAILER_FLAGS', 'S')dnl   # first step
MAILER(local)dnl                        # second step
```

Note – `MODIFY_MAILER_FLAGS` is a new macro that is used to build the configuration file. For details, refer to "New Macros Used to Build the `sendmail` Configuration File" on page 404.

For a complete review, refer to the `mail.local`(1M) man page.

Changes to `mailstats`

The `mailstats` program, which provides statistics on mailer usage, comes with the `sendmail` program. The following table describes new options in `mailstats`.

TABLE 26–23 New `mailstats` Options

Option	Description
`-C` *filename*	Specifies a `sendmail` configuration file
`-p`	Provides clear statistics in a program-readable mode
`-P`	Also provides clear statistics in a program-readable mode, but this option does not truncate the statistics file

For more information, refer to the `mailstats`(1) man page.

Changes to `makemap`

The `makemap` command creates keyed database files for `sendmail`. The following table describes new `makemap` options. When you declare options, use the following syntax.

`makemap` *options class filename*

When you use the preceding syntax, remember the following.

- *options* are preceded by a dash (for example, `-dN`).
- *class* refers to the type of database (for example, `btree`, `dbm`, or `hash`).
- *filename* refers to the full path (or relative name) for the database file.

TABLE 26–24 New `makemap` Options

Option	Description
`-C`	Uses the specified `sendmail` configuration file for finding the `TrustedUser` option
`-c`	Uses the specified `hash` and `btree` cache size
`-e`	Allows an empty value from the right-hand side (RHS)
`-l`	Lists supported map types
`-t`	Specifies a different delimiter, instead of white space
`-u`	Dumps (unmaps) the contents of the database to standard output

Note – If `makemap` is running as root, the ownership of the generated maps is automatically changed to the `TrustedUser`, as specified in the `sendmail` configuration file. For more information about the `TrustedUser` option, refer to Table 26–2.

For more information, refer to the `makemap`(1M) man page.

New Command, `editmap`

Use the new maintenance command, `editmap`, to query and edit single records in keyed database maps for `sendmail`. From the command line, use the following syntax.

`editmap` *options maptype mapname key* `"`*value*`"`

- *options* are preceded by a dash (for example, `-Nf`). The man page provides a list of options and explains how each option functions.
- *maptype* refers to the type of database. `editmap` can use `btree`, `dbm`, and `hash`.
- *mapname* refers to the full path (or relative name) for the database file.
- *key* refers to a single or multitoken string that you can use for searches.

- "*value*" refers to the string that appears to the right of the key in a keyed database file. In the following example, `man` is the key and `man@example.com` is the assigned value for that key.

  ```
  man    man@host.com
  ```

For a detailed description and a list of options, refer to the `editmap`(1M) man page.

Other Changes and Features of Interest

The following list describes other changes and features of interest.

- As noted in RFC 2476, `sendmail` now listens for submissions on port 587.
- As was noted in the Release Notes that are part of the `sendmail` distribution at ftp://ftp.sendmail.org, the `XUSR` SMTP command is deprecated. Mail user agents should begin using RFC 2476 `Message Submission` for initial user message submission.
- The `Content-Length:` header is no longer provided in messages that are piped to programs with any version of the Sun configuration files. However, this header is still provided for appended messages and ordinary mailbox deliveries that use any version of the Sun configuration files.
- `sendmail` now accepts connections when disk space is low, but in such situations it allows only `ETRN` commands.
- Entries in the alias file can be continued by putting a backslash directly before the new line.
- The timeout for sending a message by way of SMTP has been changed to check for delivery progress every five minutes. This change detects an inability to send information more quickly and reduces the number of processes that are waiting to time out.
- You can now copy the contents of a class to another class by using the syntax of the following example.

  `C{`*Dest*`}  $={`*Source*`}`

  In the preceding example, all items in class `$={`*Source*`}` are copied into class `$={`*Dest*`}`.
- The maps are no longer optional by default. Also, if a problem occurs with a map, you receive an error message.
- Canonification is no longer attempted for any host or domain in class `P` (`$=P`).
- The `=` equate is not included in an option expansion if no value is associated with the option.

- Route addresses are stripped. For example, `<@a,@b,@c:user@d>` is converted to `<user@d>`.

CHAPTER 27

Modem–Related Network Services Topics

Chapter 28	Provides overview information for PPP
Chapter 29	Provides planning information for PPP
Chapter 30	Provides step-by-step instructions for setting up a dial-up PPP link
Chapter 31	Provides step-by-step instructions for setting up a leased line PPP link
Chapter 32	Provides step-by-step instructions for setting up authentication on a PPP link
Chapter 33	Provides step-by-step instructions for creating PPPoE tunnels to support PPP links over DSL equipment.
Chapter 34	Provides instructions for PPP link maintenance and problem solving
Chapter 35	Provides detailed reference information for working with PPP
Chapter 36	Provides step-by-step instructions for migrating from the earlier Solaris PPP (`asppp`) to Solaris PPP 4.0
Chapter 37	Provides background information on UUCP
Chapter 38	Provides step-by-step instructions for setting up and troubleshooting UUCP
Chapter 39	Provides reference material on UUCP database files, UUCP configuration files, UUCP shell scripts, and UUCP troubleshooting information

CHAPTER 28

Solaris PPP 4.0 (Overview)

Solaris PPP 4.0 enables two computers in different physical locations to communicate with each other using Point-to-Point Protocol (PPP) over a variety of media. The Solaris 9 operating environment includes Solaris PPP 4.0 as part of the base installation.

This chapter introduces Solaris PPP 4.0. Topics that are discussed include:

Solaris PPP 4.0 Basics

Solaris PPP 4.0 implements the Point-to-Point Protocol (PPP) data link protocol, a member of the TCP/IP protocol suite. PPP describes how data is transmitted between two endpoint machines, over communications media such as telephone lines.

Since the early 1990s, PPP has been a widely used Internet standard for sending datagrams over a communications link. The PPP standard is described in RFC 1661 by the Point-to-Point Working Group of the Internet Engineering Task Force (IETF). PPP is commonly used when remote machines, such as home computers, call an Internet service provider (ISP) or a corporate server that is configured to receive incoming calls.

Solaris PPP 4.0 is based on the publicly available Australian National University (ANU) PPP–2.4 PPP and implements the PPP standard. Both asynchronous and synchronous PPP links are supported.

Solaris PPP 4.0 Compatibility

Various versions of standard PPP are available and in wide use throughout the Internet community. ANU PPP-2.4 is a popular choice for Linux and all three major BSD variants (FreeBSD, OpenBSD, and NetBSD), and Tru64 UNIX.

Solaris PPP 4.0 brings the highly configurable features of ANU PPP-2.4 to machines that run the Solaris operating environment. Machines that run Solaris PPP 4.0 can easily set up PPP links, not only to machines that run ANU PPP-2.4, but to any machine that runs an implementation of standard PPP.

Some non-ANU-based PPP implementations that successfully interoperate with Solaris PPP 4.0 include the following:

- Solaris PPP, also known as `asppp`, available with the Solaris 2.4 through Solaris 8 operating environments
- Solstice™ PPP 3.0.1
- Windows 98 DUN
- Cisco IOS 12.0 (synchronous)

Which Version of Solaris PPP to Use

Solaris PPP 4.0 is the PPP implementation supported by the Solaris 9 operating environment.The Solaris 9 operating environment does not include the earlier asynchronous PPP (`asppp`) software. Asynchronous PPP configuration is discussed in the Solaris 8 System Administrators' Collection at `http://www.docs.sun.com`.

Why Use Solaris PPP 4.0?

If you currently use `asppp`, consider migrating to Solaris PPP 4.0. Note the following differences between the two Solaris PPP technologies:

- **Transfer modes**

  `asppp` supports asynchronous communications only. Solaris PPP 4.0 supports both asynchronous and synchronous communications.

- **Configuration process**

  Setting up `asppp` requires configuring the `asppp.cf` configuration file, three UUCP files, and the `ifconfig` command. Moreover, you have to preconfigure interfaces for all users who might log in to a machine.

  Setting up Solaris PPP 4.0 requires defining options for the PPP configuration files, or issuing the `pppd` command with options, or a combination of both. Solaris PPP dynamically creates and removes interfaces. You do not have to directly configure PPP interfaces for each user.

- **Solaris PPP 4.0 features not available from asppp**
  - MS-CHAPv1 and MS-CHAPv2 authentication
  - PPP over Ethernet (PPPoE) , to support ADSL bridges
  - PAM authentication
  - Plug-in modules
  - IPv6 addressing
  - Data compression that uses Deflate or BSD compress

Solaris PPP 4.0 Upgrade Path

If you are converting an existing `asppp` configuration to Solaris PPP 4.0, you can use the translation script that is provided with this release. For complete instructions, refer to "How to Convert From `asppp` to Solaris PPP 4.0" on page 567.

Where to Go for More Information

Many resources with information regarding PPP cam be found in print and online. The following subsections give some suggestions.

Trade Books

For more information about widely used PPP implementations, including ANU PPP, refer to the following books:

- Carlson, James. *PPP Design, Implementation, and Debugging*. 2nd. Addison-Wesley, 2000.
- Sun, Andrew. *Using and Managing PPP*. O' Reilly & Associates, 1999.

Web Sites

Go to the following web sites for general information about PPP:

- For a list of frequently asked questions (FAQ) and other information regarding `pppd`, go to the following site that is provided by the Internet Engineering group at Sun Microsystems, `http://playground.sun.com/pppd`.
- For ANU PPP information, go to the PPP repository of Australian National University, `http://.pserver.samba.org/cgi-bin/cvsweb/ppp/`.
- For technical information, FAQs, discussions about Solaris system administration, and earlier versions of PPP, go to Sun Microsystem's system administrators' resource, `http://www.sun.com/bigadmin/home/index.html`.

- For modem configuration and advice on many different implementations of PPP, refer to Stokely Consulting's Web Project Management & Software Development web site, `http://www.stokely.com/unix.serial.port.resource/ppp.slip.htm`.

Requests for Comments (RFCs)

Some useful Internet RFCs about PPP include the following:

- 1661 and 1662, which describe the major features of the PPP protocol
- 1334, which describes authentication protocols, such as Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP)
- 2516, an informational RFC which describes PPP over Ethernet (PPPoE)

To obtain copies of PPP RFCs, specify the number of the RFC on the IETF RFC web page at `http://www.ietf.org/rfc.html` .

Man Pages

For technical details about the Solaris PPP 4.0 implementation, refer to the following man pages:

- `pppd`(1M)
- `chat`(1M)
- `pppstats`(1M)
- `pppoec`(1M)
- `pppoed`(1M)
- `sppptun`(1M)
- `snoop`(1M)

You can find the PPP-related man pages in the *Solaris 9 Beta Reference Manual Collection* or through the `man` command.

PPP Configurations and Terminology

This section introduces PPP configurations and terms that are used in this guide.

Solaris PPP 4.0 supports a number of configurations.

- Switched access, or *dial-up*, configurations

- Hardwired, or *leased-line* configurations

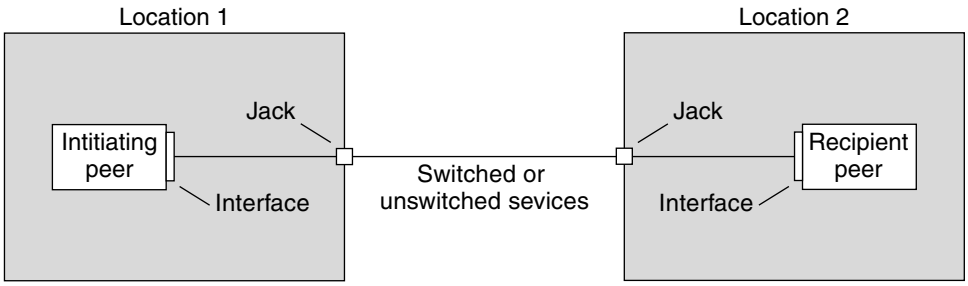


FIGURE 28–1 Parts of the PPP Link

The previous figure shows a basic PPP link. The link has the following parts:

- Two machines, usually in separate physical locations, called *peers*. A peer could be a personal computer, engineering workstation, large server, or even a commercial router, depending on a site's requirements.
- Serial interface on each peer. On Solaris machines, this interface could be `cua`, `hih`, or other interface, depending on whether you configure asynchronous or synchronous PPP.
- Physical link, such as a serial cable, a modem connection, or a leased line from a network provider, such as a T1 or T3 line.

Dial-up PPP Overview

The most commonly used PPP configuration is the *dial-up link*. In a dial-up link, the local peer *dials up* the remote peer to establish the connection and run PPP. In the dial-up process, the local peer calls the remote peer's telephone number to initiate the link.

A common dial-up scenario includes a computer in a user's home that calls a peer at an ISP, configured to receive incoming calls. Another dial-up scenario is a corporate site where a local machine in one building uses a PPP link to transmit data to a peer in another building.

In this guide, the local peer that initiates the dial-up connection is referred to as the *dial-out machine*. The peer that receives the incoming call is referred to as the *dial-in server*, although this machine is simply the target peer of the dial-out machine.

PPP is not a client-server protocol. Some PPP documents use the terms "client" and "server" to refer to telephone call establishment. A dial-in server is not a true server like a file server or name server. Dial-in server is a widely used PPP term because

dial-in machines often “serve” network accessibility to more than one dial-out machine. Nevertheless, the dial-in server is simply the target peer of the dial-out machine.

Parts of the Dial-up PPP Link

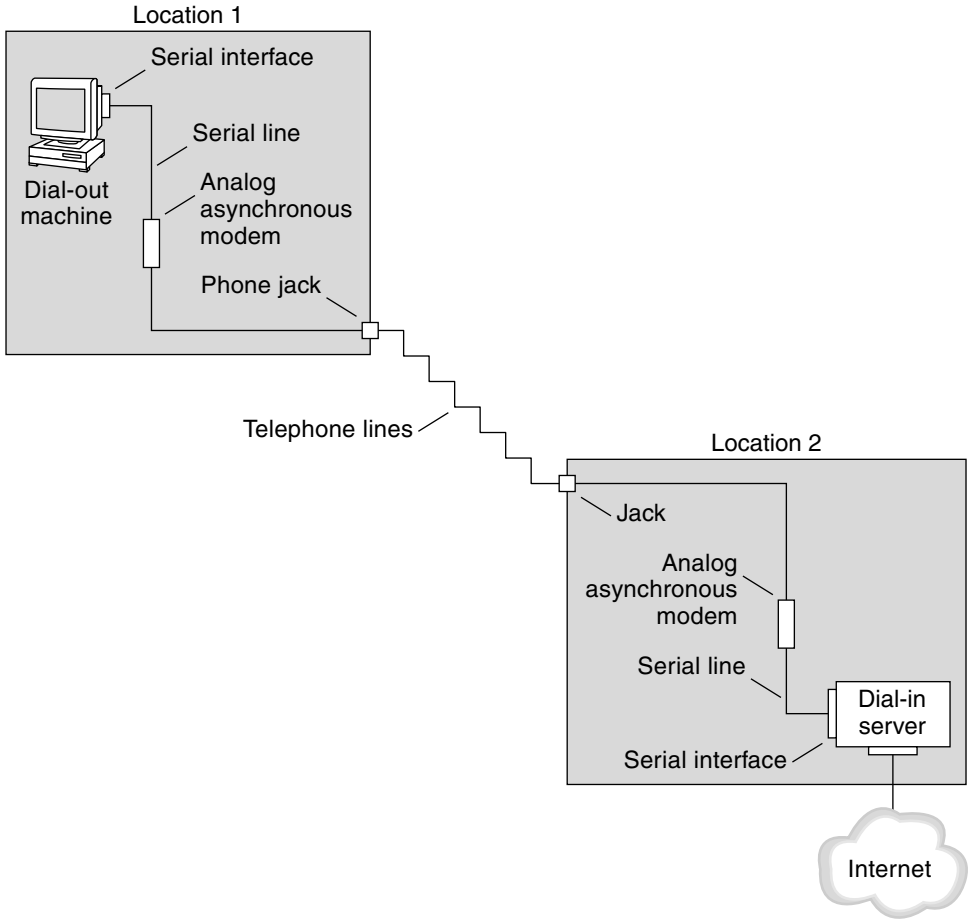


FIGURE 28–2 Basic Analog Dial-up PPP Link

The configuration for the dial-out (Location 1) side of the link is composed of the following elements::

- Dial-out machine, typically a personal computer or workstation in an individual’s home.

- Serial interface on the dial-out machine. `/dev/cua/a` or `/dev/cua/b` is the standard serial interface for outgoing calls on machines running Solaris software.
- Asynchronous modem or ISDN terminal adapter (TA) connected to a telephone jack.
- Telephone lines and services of a telephone company.

The configuration for the dial-in side (Location 2) of the link is composed of the following elements:

- Telephone jack or similar connector, connected to the telephone network
- Asynchronous modem or ISDN TA
- Serial interface on the dial-in server, either `ttya` or `ttyb` for incoming calls
- Dial-in server, which is connected to a network, such as a corporate intranet, or, in the instance of an ISP, the global Internet

Using ISDN Terminal Adapters With a Dial-out Machine

External ISDN TAs have faster speeds than modems, but you configure them in basically the same way. The major difference in configuring an ISDN TA is in the chat script, which requires commands specific to the TA's manufacturer. Refer to "Chat Script for External ISDN TA" on page 540 for information on chat scripts for ISDN TAs.

What Happens During Dial-up Communications

PPP configuration files on both the dial-out and dial-in peers contain instructions for setting up the link. The following process occurs as the dial-up link is initiated:

1. User or process on the dial-out machine runs the `pppd` command to start the link.
2. Dial-out machine reads its PPP configuration files and sends instructions over the serial line to its modem, including the phone number of the dial-in server.
3. Modem dials the phone number and establishes a telephone connection with the modem on the dial-in server.

   If necessary, the dial-out machine sends commands to the dial-in server to invoke PPP on the server.
4. Modem attached to the dial-in server begins link negotiation with the modem on the dial-out machine.

   The series of text strings that the dial-out machine sends to the modem and dial-in server are contained in a file called a *chat script*.
5. When modem-to-modem negotiation completes, the modem on the dial-out machine reports "CONNECT."

6. PPP on both peers enters *Establish* phase, where Link Control Protocol (LCP) negotiates basic link parameters and the use of authentication.
7. If necessary, the peers authenticate each other.
8. PPP's Network Control Protocols (NCPs) negotiate the use of network protocols, such as IPv4 or IPv6

The user on the dial-out machine can then run `rlogin`, `telnet`, or a similar command to a host on a network reachable from the dial-in server.

Leased-Line PPP Overview

A hardwired, *leased-line* PPP configuration involves two peers that are connected by a link that consists of a switched or unswitched digital service leased from a provider. Solaris PPP 4.0 works over any full-duplex, point-to-point leased-line medium. Typically, a company rents a hardwired link from a network provider to connect to an ISP or other remote site.

Comparison of Dial-Up and Leased-Line Links

Both dial-up and leased-line links involve two peers connected by a communications medium. The next table summarizes the differences between the link types.

Leased Line	Dial-up Line
Always connected unless a system administrator or power failure takes it down	Initiated on demand, when a user tries to call a remote peer
Uses synchronous communications	Uses asynchronous communications
Rented from a provider	Uses existing telephone lines
Requires synchronous units	Uses less costly modems
Requires specialized interfaces	Uses standard serial interfaces that are included on most computers.

Parts of a Leased-Line PPP Link

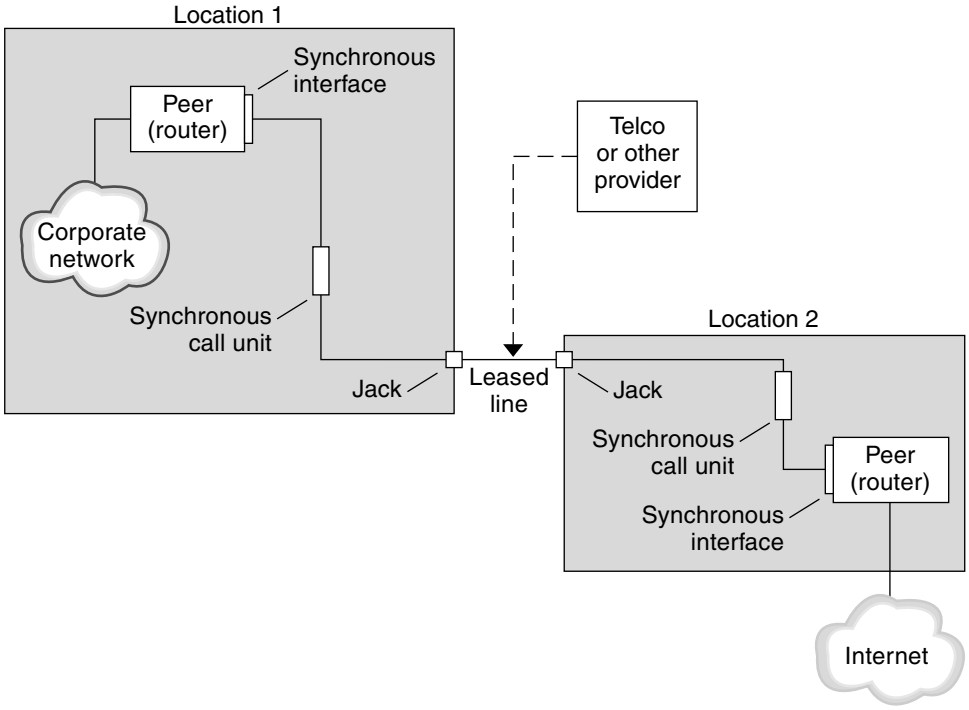


FIGURE 28–3 Basic Leased-Line Configuration

The parts of the leased-line link include:

- **Two peers**, one at each end of the link. Each peer might be a workstation or server. Often the peer functions as a router between its network or the Internet, and the opposite peer.
- **Synchronous interface on each peer**. Some machines running Solaris software require you to purchase a synchronous interface card, such as HSI/S, to connect to a leased line. Other machines, such as Sun Ultras, have built-in synchronous interfaces.
- **CSU/DSU synchronous digital unit on each peer**, which connects the synchronous port to the leased line.

  A CSU might be built-in to the DSU, or owned by you, or leased from a provider, ,depending on your locale. The DSU gives the Solaris machine a standard synchronous serial interface. With Frame Relay, the Frame Relay Access Device (FRAD) performs the serial interface adaptation.
- **Leased line**, providing switched or unswitched digital services. Some examples are SONET/SDH, Frame Relay PVC, and T1.

Note – SONET is called an *octet synchronous* link. PPP uses a framing mechanism similar to asynchronous framing over a SONET line. PPP does not use the expected bit-synchronous protocol.

What Happens During Leased-Line Communications

On most types of leased lines, peers do not actually dial each other. Rather, a company purchases a leased-line service to explicitly connect between two fixed locations. Sometimes the two peers at either end of the leased line are at different physical locations of the same company. Another scenario is a company that sets up a router on a leased line that is connected to an ISP.

Leased lines are less commonly used than dial-up links, though the hardwired links are easier to set up. Hardwired links do not require chat scripts. Authentication is often not used because both peers are known to each other when a line is leased. After the two peers initiate PPP over the link, it stays active unless the leased line fails or either peer explicitly terminates the link.

A peer on a leased line that runs Solaris PPP 4.0 uses most of the same configuration files that define a dial-up link.

The following process occurs to initiate communication over the leased line:

1. Each peer machine runs the `pppd` command as part of the booting process or other administrative script.
2. The peers read their PPP configuration files.
3. The peers negotiate communications parameters.
4. An IP link is established.

PPP Authentication

Authentication is the process of verifying that a user is who he or she claims to be. The classic UNIX login sequence is a simple form of authentication:

1. The `login` command prompts the user for a name and password.
2. `login` then attempts to authenticate the user by looking up the typed user name and password in the password database.
3. If the database contains the user name and password, then the user is *authenticated* and given access to the system. If the database does not contain the user name and password, the user is denied access to the system.

By default, Solaris PPP 4.0 does not demand authentication on machines that do not have a default route specified. Thus, a local machine without a default route does not authenticate remote callers. Conversely, if a machine does have a default route defined, by default it does authenticate remote callers.

If necessary, you can use PPP authentication protocols to verify the identity of callers who are trying to set up a PPP link to your machine. Conversely, you must configure PPP authentication information for your local machine if it needs to call peers that must authenticate callers.

Authenticators and Authenticatees

The calling machine on a PPP link is considered the *authenticatee* because it must prove its identity to the remote peer. The peer is considered the *authenticator*. It looks up the caller's identity in the appropriate PPP files for the security protocol and authenticates (or does not) authenticate the caller.

You typically configure PPP authentication for a dial-up link. When the call begins, the dial-out machine is the authenticatee. The dial-in server is the authenticator. The server has a database in the form of a *secrets* file, which lists all users who are granted permission to set up a PPP link to the server. Think of these users as *trusted callers*.

Some dial-out machines require remote peers to provide authentication information when responding to the dial-out machine's call. Then their roles are reversed: the remote peer becomes the authenticatee and the dial-out machine the authenticator.

Note – PPP 4.0 does not prevent authentication by leased-line peers, but it is not often used. The nature of leased-line contracts usually means that both participants on the ends of the line are known to each other and often are trusted. However, because PPP authentication is not that difficult to administer, you should seriously consider implementing authentication for leased lines.

PPP Authentication Protocols

The PPP authentication protocols are Password Authentication Protocol (PAP) and Challenge-Handshake Authentication Protocol (CHAP). Each protocol uses a *secrets* database that contains identification information, or *security credentials*, for each caller that is permitted to link to the local machine. For a detailed explanation of PAP, go to "Password Authentication Protocol (PAP)" on page 544. For a CHAP explanation, go to "Challenge-Handshake Authentication Protocol (CHAP)" on page 547.

Why Use PPP Authentication?

Providing authentication on a PPP link is optional. Moreover, though authentication does verify that a peer is to be trusted, it does not provide confidentiality of data. (For confidentiality, use encryption software, such as IPsec, PGP, ssh, and SSL.)

Note – Solaris PPP 4.0 does not implement the PPP Encryption Control Protocol (ECP) described in RFC 1968.

Consider implementing PPP authentication in the following cases:

- Your company accepts incoming calls from users over the public, switched telephone network.
- Your corporate security policy requires remote users to provide authentication credentials when accessing your network through a corporate firewall or when engaging in secure transactions.
- You want to authenticate callers against a standard UNIX password database (`/etc/passwd`, NIS, NIS+, LDAP, or PAM). Use PAP authentication for this scenario.
- Your company's dial-in servers also provide the network's Internet connection. Use PAP authentication for this scenario.
- The serial line is less secure than the password database on the machine or networks at either end of the link. Use CHAP authentication for this scenario.

Support for DSL Users Through PPPoE

Many network providers and individuals who are working at home use Digital Subscriber Line (DSL) technology to provide fast network access. To support DSL users, Solaris PPP 4.0 includes the PPP over Ethernet (PPPoE) feature. PPPoE technology enables multiple hosts to run PPP sessions over one Ethernet link to one or more destinations.

If one of the following factors apply to your situation, you should use PPPoE:

- You support DSL users, possibly including yourself. Your DSL service provider might require users to configure a PPPoE tunnel to receive services over the DSL line.
- Your site is an ISP that intends to offer PPPoE to customers.

This section introduces terms that are associated with PPPoE and an overview of a basic PPPoE topology.

PPPoE Overview

PPPoE is a proprietary protocol from RedBack Networks. PPPoE is a discovery protocol, rather than another version of standard PPP. In a PPPoE scenario, a machine that initiates PPP communications first must locate, or *discover*, a peer that runs PPPoE. The PPPoE protocol uses Ethernet broadcast packets to locate the peer.

After the discovery process, PPPoE sets up an Ethernet-based tunnel from the initiating host, or *PPPoE client*, to the peer, the *PPPoE access server*.*Tunneling* is the practice of running one protocol on top of another protocol that normally occupies a higher or the same position on the TCP/IP protocol stack. Using PPPoE, Solaris PPP 4.0 tunnels PPP over Ethernet IEEE 802.2, both of which are data link protocols. The resulting PPP connection behaves like a dedicated link between the PPPoE client and the access server. For detailed information about PPPoE, see "Creating PPPoE Tunnels for DSL Support" on page 552.

Parts of a PPPoE Configuration

Three participants are involved in a PPPoE configuration: a consumer, a telephone company, and a service provider, as the following figure shows.

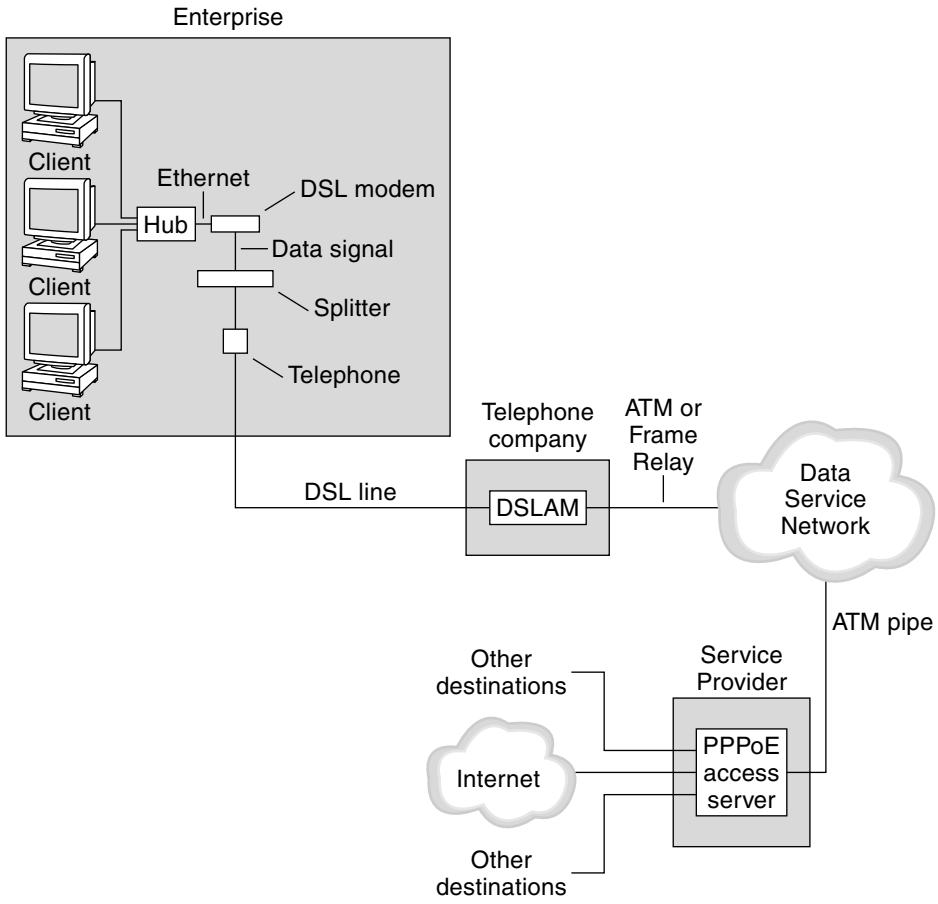


FIGURE 28–4 Participants in a PPPoE Tunnel

The PPPoE Consumer

As system administrator, you might assist consumers with their PPPoE configurations. One common type of PPPoE consumer is an individual who needs to run PPPoE over a DSL line. Another PPPoE consumer is a company that purchases a DSL line through which employees can run PPPoE tunnels, as illustrated in the previous figure.

The main reason for a corporate consumer to use PPPoE is to offer PPP communications through a high-speed DSL device to a number of hosts. Often a single PPPoE client has an individual *DSL modem*. Or, a group of clients that are connected to a hub might share a DSL modem that is also connected to the hub by an Ethernet line.

Note – DSL devices are technically bridges, not modems. However, because common practice is to refer to these devices as modems, this guide uses the term "DSL modem."

When PPPoE runs, it runs PPP over a tunnel on the Ethernet line that is connected to the DSL modem. That line is connected to a splitter, which, in turn connects to a telephone line.

PPPoE at a Telephone Company

The telephone company is the middle layer of the PPPoE scenario. The telephone company splits the signal that is received over the phone line by using a device that is called a *Digital Subscriber Line Access Multiplexer (DSLAM)*. The DSLAM breaks out the signals onto separate wires, analog wires for telephone service and digital wires for PPPoE. From the DSLAM, the digital wires extend the tunnel over an ATM data network to the ISP.

PPPoE at a Service Provider

The ISP receives the PPPoE transmission from the ATM data network over a bridge. At the ISP, an access server that runs PPPoE functions as the peer for the PPP link. The access server is very similar in function to the dial-in server that was introduced in Figure 28–2, but the access server does not use modems. It converts the individual PPPoE sessions into regular IP traffic, for example Internet access.

If you are a system administrator for an ISP, you might be responsible for configuring and maintaining an access server.

Security on a PPPoE Tunnel

The PPPoE tunnel is inherently insecure. You can use PAP or CHAP to provide user authentication for the PPP link that is running over the tunnel.

CHAPTER 29

Planning for the PPP Link (Tasks)

Setting up a PPP link involves a set of discrete tasks, including planning tasks and other activities that are not related to PPP. This chapter explains how to plan for the most common PPP links, for authentication, and for PPPoE.

The task chapters that follow Chapter 29 use sample configurations to illustrate how to set up a particular link. These sample configurations are introduced in this chapter.

Topics that are covered include the following:

Overall PPP Planning (Task Map)

PPP requires planning tasks before you actually can set up the link. Moreover, if you want to use a PPPoE tunneling, you first have to set up the PPP link and then provide tunneling. The following task map lists the large planning tasks discussed in this chapter. You might need to use only the general task for the link type to be configured. Or you might require the task for the link, authentication, and perhaps PPPoE.

TABLE 29–1 Task Map for PPP Planning

Task	Description	For Instructions
Plan for a dial-up PPP link	Gather information required to set up a dial-out machine or a dial-in server.	"Planning a Dial-up PPP Link" on page 440

TABLE 29–1 Task Map for PPP Planning *(Continued)*

Task	Description	For Instructions
Plan for a leased-line link	Gather information required to set up a client on a leased line.	"Planning a Leased-Line Link" on page 444
Plan for authentication on the PPP link	Gather information required to configure PAP or CHAP authentication on the PPP link.	"Planning for Authentication on a Link" on page 447
Plan for a PPPoE tunnel	Gather information required to set up a PPPoE tunnel over which a PPP link can run.	"Planning for DSL Support Over a PPPoE Tunnel" on page 452

Planning a Dial-up PPP Link

Dial-up links are the most commonly used PPP links. This section includes the following information:

- Planning information for a dial-up link
- Explanation of the sample link to be used in Chapter 30.

Typically, you only configure the machine at one end of the dial-up PPP link, the dial-out machine or the dial-in server. For an introduction to dial-up PPP, refer back to "Dial-up PPP Overview" on page 427

Before You Set Up the Dial-out Machine

Before you configure a dial-out machine, gather the information that is listed in the following table.

Note – The planning information in this section does not include information to be gathered about authentication or PPPoE. For details on authentication planning, refer to "Planning for Authentication on a Link" on page 447. For PPPoE planning, refer to "Planning for DSL Support Over a PPPoE Tunnel" on page 452

TABLE 29–2 Information for a Dial-out Machine

Information	Action
Maximum modem speed	Refer to documentation provided by the modem manufacturer.

TABLE 29–2 Information for a Dial-out Machine *(Continued)*

Information	Action
Modem connection commands (AT commands)	Refer to documentation provided by the modem manufacturer.
Name to use for dial-in server at the other end of the link	Create any name that helps you identify the dial-in server.
Login sequence required by dial-in server	Contact the dial-in server's administrator or ISP documentation, if dial-in server is at the ISP.

Before setting up a dial-in server, gather the information listed in the next table.

Before You Set Up the Dial-in Server

Before you configure a dial-in server, gather the information that is listed in the following table.

Note – The planning information in this section does not include information to be gathered about authentication or PPPoE. For details on authentication planning, refer to "Planning for Authentication on a Link" on page 447. For PPPoE planning, refer to "Planning for DSL Support Over a PPPoE Tunnel" on page 452

TABLE 29–3 Information for a Dial-in Server

Information	Action
Maximum modem speed	Refer to documentation provided by the modem manufacturer.
User names of people who are permitted to call the dial-in server	Obtain the names of the prospective users before you set up their home directories, as discussed in "How to Configure Users of the Dial-in Server" on page 467.
Dedicated IP address for PPP communications	Obtain an address from the individual at your company responsible for delegating IP addresses.

Example— Configuration for Dial-up PPP

The tasks to be introduced in Chapter 30 carry out a small company's requirement to let employees work at home a few days a week. Some employees require the Solaris operating environment on their home machines. They also need to log in remotely to their work machines on the corporate intranet.

The tasks set up a basic dial-up link with the following features:

- The *dial-out* machines are at the houses of employees who need to call the corporate intranet.
- The *dial-in* server is a machine on the corporate intranet that is configured to receive incoming calls from employees.
- UNIX-style login is used to authenticate the dial-out machine. Stronger Solaris PPP 4.0 authentication methods are not required by the company's security policy.

The next figure shows the link that is set up in Chapter 30.

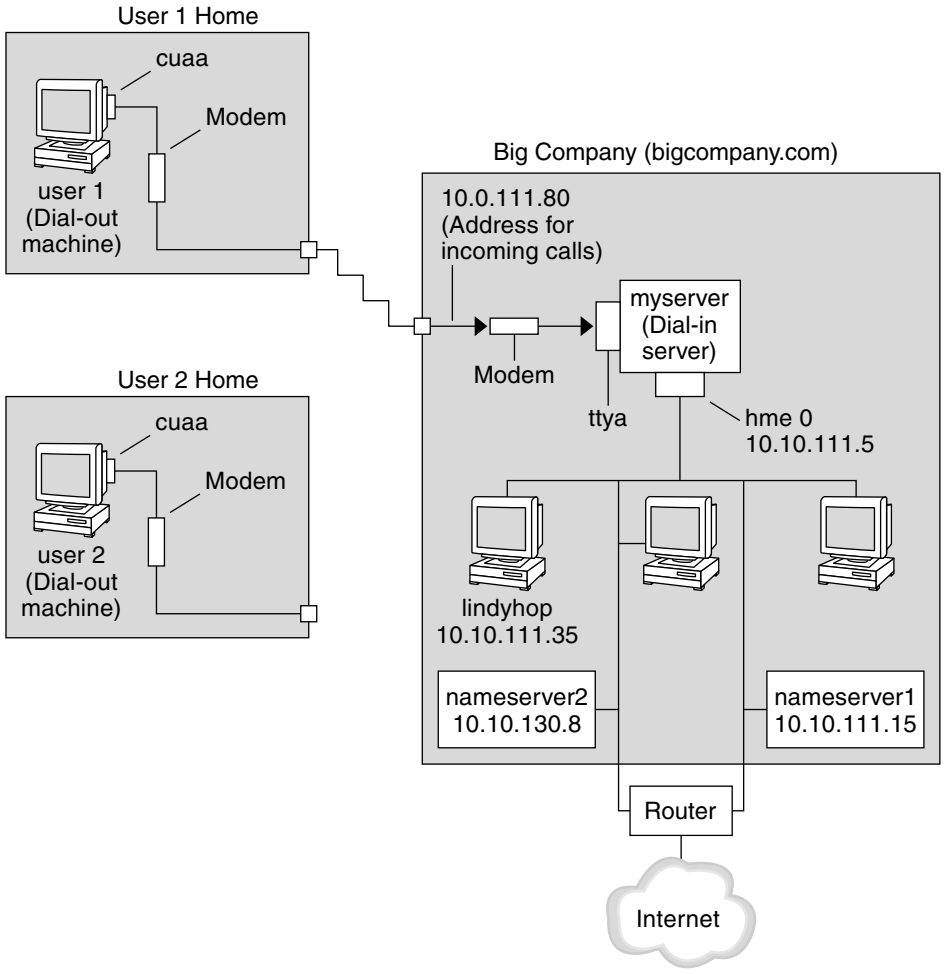


FIGURE 29–1 Sample Dial-up Link

In this figure, a remote host dials out through its modem over telephone lines to Big Company's intranet. Another host is configured to dial out to Big Company but currently is inactive. One at a time, the calls from remote users are answered by the modem that is attached to the dial-in server at Big Company. A PPP connection is established between the peers. Then the dial-out machine can remotely log in to a host machine on the intranet.

Where to Go For More Information About Dial-up PPP

Task	For Information
Set up a dial-out machine	Table 30–2
Set up a dial-in machine	Table 30–4
Get an overview of dial-up links	"Dial-up PPP Overview" on page 427
Get detailed information about PPP files and commands	"Using PPP Options in Files and on the Command Line" on page 521

Planning a Leased-Line Link

Setting up a leased-line link involves configuring the peer at one end of a switched or unswitched service leased from a provider.

This section includes the following information:

- Planning information for a leased-line link
- Explanation of the sample link shown in Figure 29–2

For an introduction to leased-line links, refer to "Leased-Line PPP Overview" on page 430. For tasks for setting up the leased line, see Chapter 31.

Before You Set Up the Leased-Line Link

When your company rents a leased-line link from a network provider, you typically configure only the system at your end of the link. The peer at the other end of the link is maintained by another administrator. This individual might be a system administrator at a remote location in your company or a system administrator at an ISP.

Hardware That Is Needed for a Leased-Line Link

In addition to the link media itself, your end of the link requires the following hardware:

- Synchronous interface for your system

- Synchronous unit (CSU/DSU)
- Your system

Some network providers include a router, synchronous interface, and a CSU/DSU as part of the customer premises equipment (CPE). However, necessary equipment varies according to provider and any governmental restrictions in your locale. The network provider can give you information about the unit needed, if it is not provided with the leased line.

Information To Be Gathered for the Leased-Line Link

Before you configure the peer at your end of a leased line, you might need to gather the following items or information that is listed in the next table.

TABLE 29–4 Planning for a Leased Line Link

Information	Action
Device name of the interface	Refer to the Interface card documentation .
Configuration instructions for the synchronous interface card	Refer to the Interface card documentation. You need this information to configure the HSI/S interface. You might not need to configure other types of interface cards.
[Optional] IP address of the remote peer	Refer to the service provider documentation or contact the system administrator of the remote peer. This information is needed only if the IP address is not negotiated between the two peers.
[Optional] Name of the remote peer	Refer to the service provider documentation or contact the system administrator of the remote peer.
[Optional] Speed of the link	Refer to the service provider documentation or contact the system administrator of the remote peer.
[Optional] Compression used by the remote peer	Refer to the service provider documentation or contact the system administrator of the remote peer.

Example—Configuration for a Leased-Line Link

The tasks in Chapter 31 show how to implement the goal of a medium-sized organization called LocalCorp to provide Internet access for its employees. Currently, the employees' computers are connected on a private, corporate intranet.

LocalCorp requires speedy transactions and access to the many resources on the Internet. The organization signs a contract with Far ISP, a service provider, that allows LocalCorp to set up its own leased line to Far ISP. Then, LocalCorp leases a T1 line

from Phone East, a telephone company. Phone East puts in the leased line between LocalCorp and Far ISP and provides to Local Corp a CSU/DSU that is already configured.

The tasks set up a leased-line link with the following characteristics:

- LocalCorp has set up a system as a gateway router, which forwards packets over the leased line to hosts on the Internet.
- Far ISP also has set up a peer as a router to which leased lines from customers are attached.

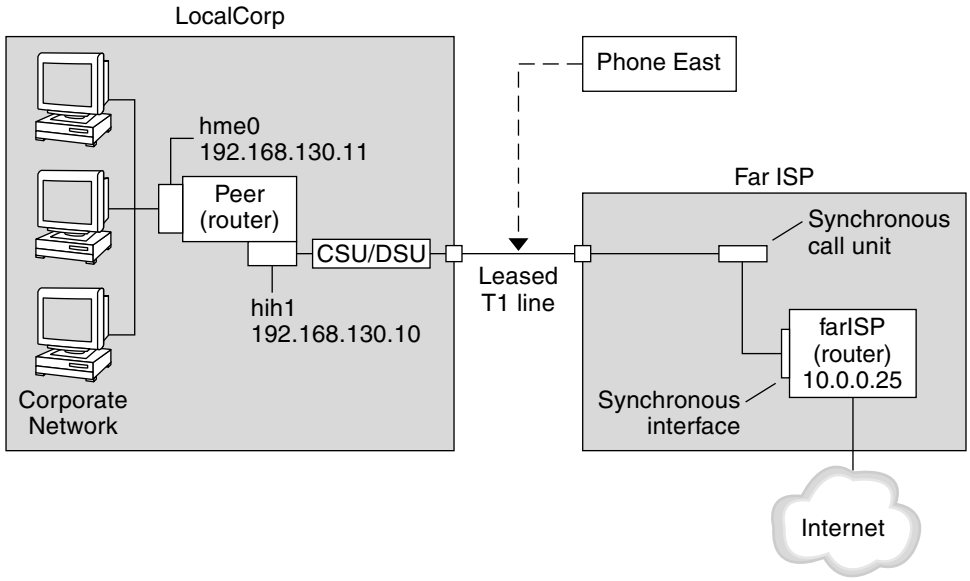


FIGURE 29–2 Sample Leased-Line Configuration

In the figure, the machine that is set up for PPP at LocalCorp is a router with a connection to the corporate intranet through its `hme0` interface. The second connection is through the machine's HSI/S interface (hih1) to the CSU/DSU digital unit. The CSU/DSU then connects to the installed leased line. The link between LocalCorp and Far ISP is initiated after the administrator at LocalCorp configures the HSI/S interface and PPP files, and then types the `/etc/init.d/pppd start` command.

Where to Get More Information About Leased Lines

Task	For Information
Set up a client on a leased line	Chapter 31
Get an overview of leased lines	"Leased-Line PPP Overview" on page 430

Planning for Authentication on a Link

This section contains planning information for providing authentication on the PPP link. Chapter 32 contains tasks for implementing PPP authentication at your site.

PPP offers two types of authentication, PAP, which is described in detail in "Password Authentication Protocol (PAP)" on page 544 and CHAP, which is described in "Challenge-Handshake Authentication Protocol (CHAP)" on page 547 .

Before you set up authentication on a link, you must choose which authentication protocol best suits your site's security policy. Then you set up the secrets file and PPP configuration files for the dial-in machines, or callers' dial-out machines, or both. For information on choosing the appropriate authentication protocol for your site, see "Why Use PPP Authentication?" on page 434.

This section includes the following information:

- Planning information for both PAP and CHAP authentication
- Explanations of the sample authentication scenarios shown in Figure 29–3 and Figure 29–4

For tasks for setting up authentication, see Chapter 32.

Before You Set Up PPP Authentication

Setting up authentication at your site should be an integral part of your overall PPP strategy. Before implementing authentication, you should assemble the hardware, configure the software, and test the link to see if it works.

TABLE 29–5 Prerequisites Before Configuring Authentication

Information	Source for Instructions
Tasks for configuring a dial-up link	Chapter 30.
Tasks for testing the link	Chapter 34.
Security requirements for your site	Your corporate security policy. If you do not have one, setting up PPP authentication gives you an opportunity to create a security policy.
Suggestions about whether to use PAP or CHAP at your site	"Why Use PPP Authentication?" on page 434. For more detailed information about these protocols, refer to "Authenticating Callers on a Link" on page 543.

Example—PPP Authentication Configurations

This section contains the sample authentication scenarios to be used in the procedures in Chapter 32.

Example—Configuration Using PAP Authentication

The tasks in "Configuring PAP Authentication " on page 480 show how to set up PAP authentication over the PPP link. The procedures use as an example a PAP scenario created for the fictitious "Big Company" that was introduced in "Example—Configuration for Dial-up PPP" on page 441.

Big Company wants to enable its users to work from home. The system administrators want a secure solution for the serial lines to the dial-in server. UNIX-style login that uses the NIS password databases has served BigCompany's network well in the past. The system administrators want a UNIX-like authentication scheme for calls that come in to the network over the PPP link. So they implement the following scenario that uses PAP authentication.

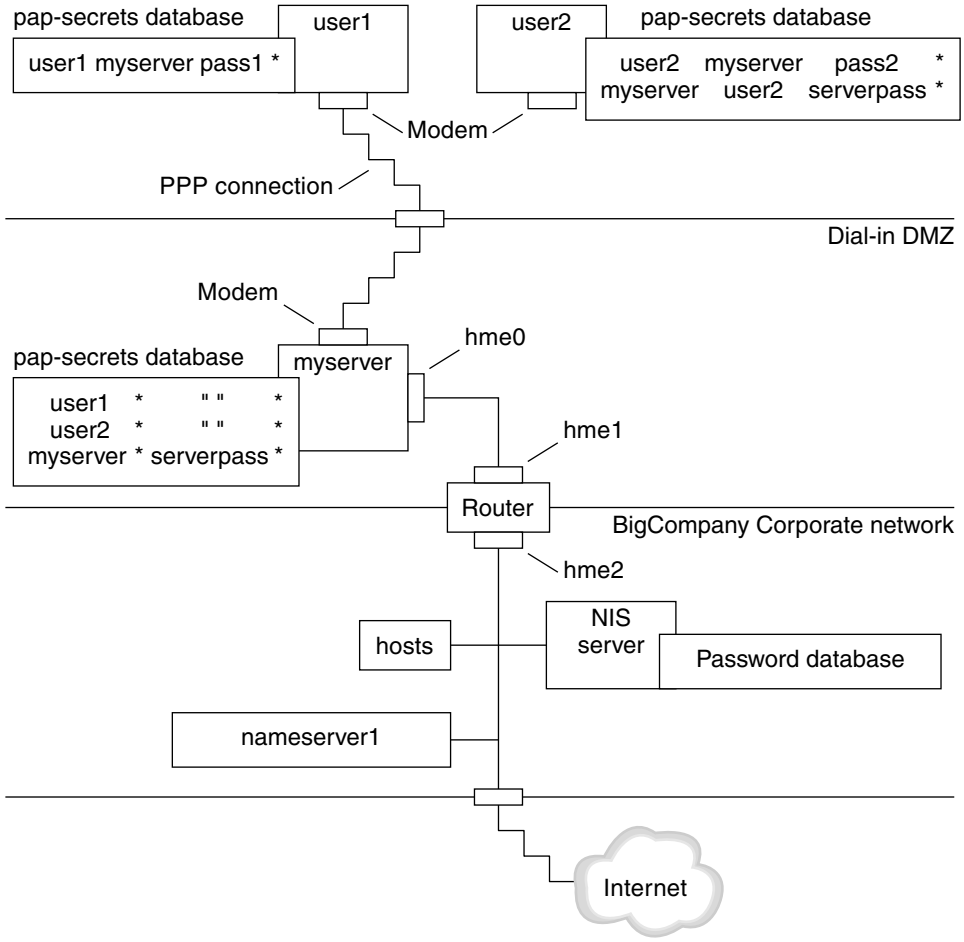


FIGURE 29–3 Example—PAP Authentication Scenario (Working From Home)

The system administrators create a dedicated dial-in DMZ that is separated from the rest of the corporate network by a router. The term DMZ comes from the military term de-militarized zone. The DMZ is an isolated network set up for security purposes. It typically contains resources that a company offers to the public, such as web servers, anonymous FTP servers, databases, and modem servers. Network designers often place the DMZ between a firewall and a company's Internet connection.

The only occupants of the DMZ pictured in Figure 29–3 are the dial-in server `myserver` and the router. The dial-in server requires callers to provide PAP credentials (including user names and passwords) when setting up the link. Furthermore, the dial-in server uses the `login` option of PAP. Therefore, the callers' PAP user names and passwords must correspond exactly to their UNIX user names and passwords that already are in the dial-in server's password database.

After the PPP link is established, the caller's packets are forwarded to the router. The router forwards the transmission to its destination on the corporate network or Internet.

Example—Configuration Using CHAP Authentication

The tasks in "Configuring CHAP Authentication" on page 487 show how to set up CHAP authentication. The procedures use as an example a CHAP scenario to be created for the fictitious LocalCorp that was introduced in "Example—Configuration for a Leased-Line Link" on page 445.

LocalCorp provides connectivity to the Internet over a leased line to an ISP. Because it generates heavy network traffic, the Technical Support department within LocalCorp requires its own, isolated private network. The department's field technicians travel extensively and need to access the Technical Support network from remote locations for problem-solving information. To protect sensitive information that is stored on the private network's database, remote callers must be authenticated before they are granted permission to log in.

Therefore, the system administrators implement the following CHAP authentication scenario for a dial-up PPP configuration.

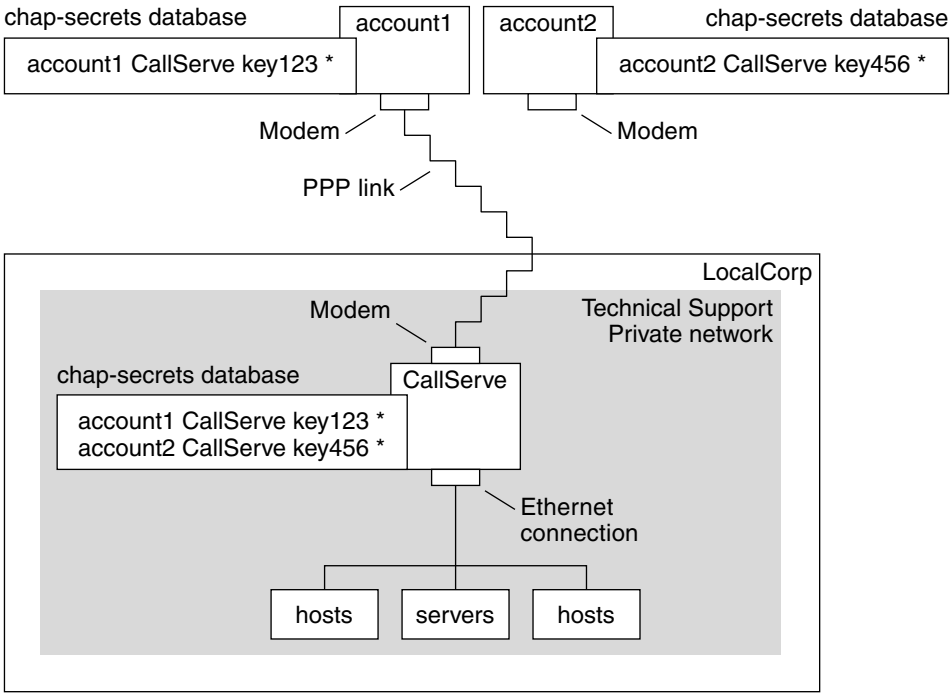


FIGURE 29–4 Example—CHAP Authentication Scenario (Calling a Private Network)

The only link from the Technical Support department network to the outside world is the serial line to the dial-in server's end of the PPP link. The system administrators configure the laptop computer of each field service representative for PPP with CHAP security, including a CHAP secret. The chap-secrets database on the dial-in server contains the CHAP credentials for all machines that are allowed to call in to the Technical Support network.

Where to Get More Information About Authentication

Task	For Instructions
Set up PAP authentication	"Configuring PAP Authentication " on page 480

Task	For Instructions
Set up CHAP authentication	"Configuring CHAP Authentication" on page 487
Learn details about PPP authentication	"Authenticating Callers on a Link" on page 543 and the `pppd`(1M) man page

Planning for DSL Support Over a PPPoE Tunnel

Some DSL providers require you to set up PPPoE tunneling for your site in order to run PPP over the providers' DSL lines and high-speed digital networks. For an overview of PPPoE, see "Support for DSL Users Through PPPoE" on page 434.

A PPPoE tunnel involves three participants: a consumer, a telephone company, and an ISP. As system administrator, you either configure PPPoE for consumers—PPPoE clients at your company or consumers in their homes—or on a server at an ISP.

This section contains planning information for running PPPoE on both clients and access servers. The following topics are covered:

- Planning information for the PPPoE host and access server
- Explanation of the PPPoE scenario introduced in "Example—Configuration for a PPPoE Tunnel" on page 454

For tasks for setting up a PPPoE tunnel, see Chapter 33.

Before You Set Up a PPPoE Tunnel

Your preconfiguration activities depend on whether you configure the client side or server side of the tunnel. In either instance, you or your organization must contract with a telephone company. It provides the DSL lines for clients, and some form of bridging and possibly an ATM pipe for access servers. In most contracts, the telephone company assembles its equipment at your site.

Before Configuring a PPPoE Client

PPPoE client implementations usually consist of the following equipment:

- Personal computer or other system used by an individual.

- DSL modem, which is usually installed by the telephone company or Internet access provider.
- [Optional]A hub, if more than one client is involved, as is the case for corporate DSL consumers
- [Optional] A splitter, usually installed by the provider.

Many different DSL configurations are possible, which depends on the user or corporation's needs and the services that are offered by the provider.

TABLE 29–6 Planning for PPPoE Clients

Information	Action
If setting up a home PPPoE client for an individual or yourself, get any setup information that is outside the scope of PPPoE.	Ask the telephone company or ISP if it requires any setup procedures.
If setting up PPPoE clients at a corporate site, get the names of users to get PPPoE clients. If you configure remote PPPoE clients, it might be your responsibility to give users information for getting DSL equipment into their homes.	Ask management at your for a list of authorized users.
Find out what interfaces are available on the PPPoE client .	Run the `ifconfig -a` command on each machine for interface names.
[Optional] Get the password for the PPPoE client.	Ask users for passwords that they prefer or assign them. Note that this password is used for link authentication, not for UNIX login.

Before Configuring a PPPoE Server

Planning for a PPPoE access server involves working with the telephone company that provides your connection to its data service network. The telephone company installs its lines, often ATM pipes, at your site, and provides some sort of bridging into your access server. You need to configure the Ethernet interfaces that access the services your company provides, for example, Internet access, as well as the Ethernet interfaces from the telephone company's bridge.

TABLE 29–7 Planning for a PPPoE Access Server

Information	Action
Interfaces that are used for lines from data service network	Run the `ifconfig -a`command to identify interfaces.
Types of services to provide from the PPPoE server	Ask management, network planners for their requirements and suggestions.

TABLE 29–7 Planning for a PPPoE Access Server *(Continued)*

Information	Action
[Optional] Types of services to provide to the consumers	Ask management, network planners for their requirements and suggestions.
[Optional] Host names and passwords for remote clients	Ask network planners and other individuals at your site responsible for contract negotiations. The host names and passwords are used for PAP or CHAP authentication, not UNIX login.

Example—Configuration for a PPPoE Tunnel

This section contains a sample PPPoE tunnel, which is used as an illustration for the tasks in Chapter 33. Though the illustration shows all participants in the tunnel, you only administer one end, either the client side or server side

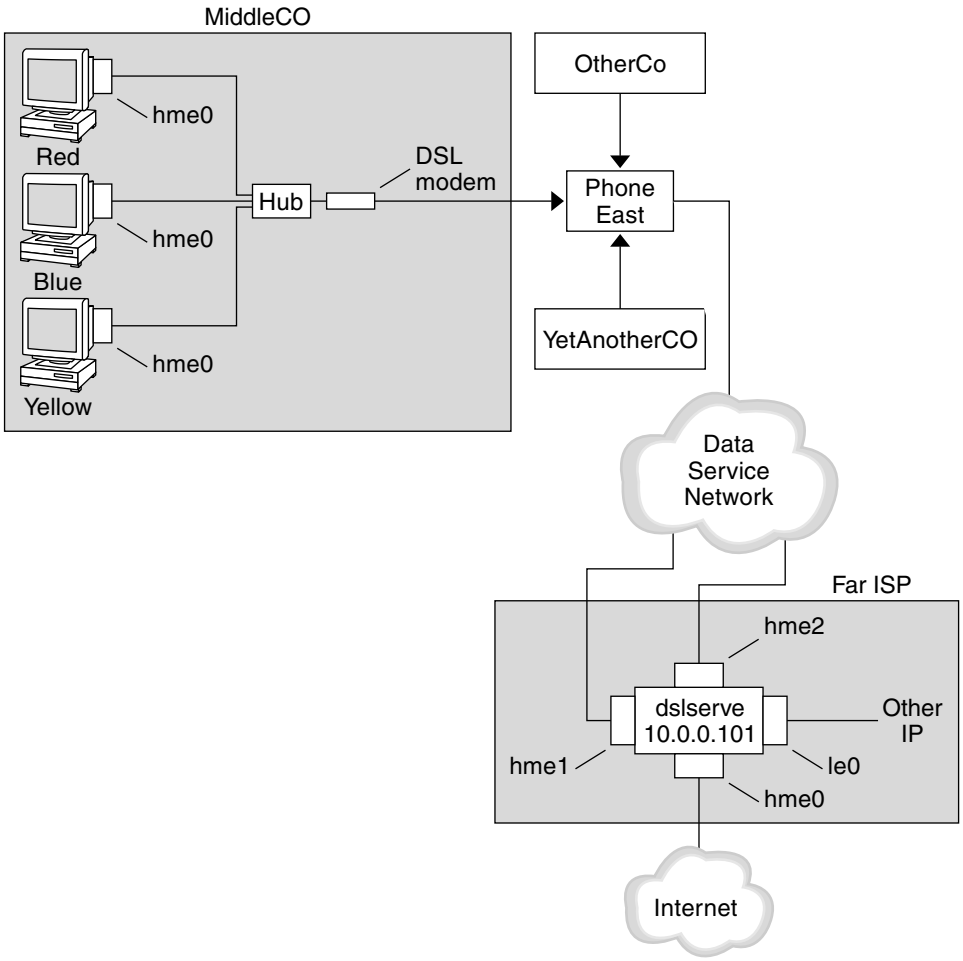


FIGURE 29–5 Example—PPPoE Tunnel

In the sample, MiddleCo wants to provide its employees with high-speed Internet access. MiddleCo buys a DSL package from Phone East, which, in turn, contracts with service provider FarISP. FarISP offers Internet and other IP services to customers who buy DSL from Phone East.

Example—PPPoE Client Configuration

MiddleCo buys a package from Phone East that provides one DSL line for the site. The package includes a dedicated, authenticated connection to the ISP for MiddleCo's PPPoE clients. The system administrator cables the prospective PPPoE clients to a hub. Technicians from Phone East cable the hub to their DSL equipment.

Example—PPPoE Server Configuration

To implement the business arrangement FarISP has with Phone East, the system administrator at FarISP configures the access server `dslserve`. This server has the following four interfaces:

- `le0` – Primary network interface, connecting to the local network
- `hme0` – Interface through which FarISP provides Internet service for its customers
- `hme1` – Interface contracted by MiddleCo for authenticated PPPoE tunnels
- `hme2` – Interface contracted by other customers for their PPPoE tunnels

Where to Get More Information About PPPoE

Task	For Information
Set up a PPPoE client	"Setting Up the PPPoE Client" on page 496
Set up a PPPoE access server	"Setting Up a PPPoE Access Server" on page 499
Get detailed information about PPPoE	"Creating PPPoE Tunnels for DSL Support" on page 552 and the `pppoed`(1M)), `pppoec`(1M), and `sppptun`(1M) man pages

CHAPTER 30

Setting Up a Dial-up PPP Link (Tasks)

This chapter explains the tasks for configuring the most common PPP link, the dial-up link. Major topics include:

Major Tasks for Setting Up the Dial-up PPP Link (Task Map)

You set up the dial-up PPP link by configuring modems, modifying network database files, and modifying the PPP configuration files that are described in Table 35–1.

The next table lists the major tasks to configure both sides of a dial-up PPP link. Typically, you configure only one end of the link, either the dial-out machine or dial-in server.

TABLE 30–1 Task Map for Setting Up the Dial-up PPP Link

Task	Description	For Instructions
1 Gather preconfiguration information	Gather data that is needed prior to setting up the link, such as peer host names, target phone numbers, and modem speed.	"Planning a Dial-up PPP Link" on page 440.
2 Configure the dial-out machine	Set up PPP on the machine that makes the call over the link.	Table 30–2

TABLE 30–1 Task Map for Setting Up the Dial-up PPP Link *(Continued)*

Task	Description	For Instructions
2 Configure the dial-in server	Set up PPP on the machine that receives incoming calls.	Table 30–4
3 Call the dial-in server	Type the `pppd` command to initiate communications.	"How to Call the Dial-In Server" on page 471.

Configuring the Dial-out Machine

The tasks in this section explain how to configure a dial-out machine. The tasks use as an example the dialing in from home scenario that was introduced in Figure 29–1. You can perform the tasks at your company before passing on the machine to a prospective user. Alternatively, you can instruct experienced users so that they can set up their own home machines. Anyone setting up a dial-out machine must have root permission for that machine.

Tasks for Configuring the Dial-out Machine (Task Map)

TABLE 30–2 Task Map for Setting Up the Dial-out Machine

Task	Description	For Instructions
1 Gather preconfiguration information.	Gather data that is needed prior to setting up the link, such as peer host names, target phone numbers, and modem speed.	"Planning a Dial-up PPP Link" on page 440.
2 Configure the modem and serial port.	Set up the modem and serial port.	"How to Configure the Modem and Serial Port (Dial-out Machine)" on page 460
3 Configure the serial line communication.	Configure the characteristics of the transmission across the serial line.	"How to Define Communications Over the Serial Line " on page 461
4 Define the conversation between the dial-out machine and the peer.	Gather communications data and use the information to create the chat script.	"How to Create the Instructions for Calling a Peer" on page 462
5 Configure information about a particular peer.	Configure PPP options to call an individual dial-in server.	"How to Define the Connection With an Individual Peer" on page 463

TABLE 30–2 Task Map for Setting Up the Dial-out Machine *(Continued)*

Task	Description	For Instructions
6 Call the peer	Type the `pppd` command to initiate communications.	"How to Call the Dial-In Server" on page 471.

Dialup PPP Template Files

Solaris PPP 4.0 provides template files, each of which contains common options for a particular PPP configuration file. The next table lists the sample templates that can be used for setting up a dialup link, and their equivalent Solaris PPP 4.0 files.

Template File	PPP Configuration File	For More Information About the Template File
`/etc/ppp/options.tmpl`	`/etc/ppp/options`	"`/etc/ppp/options.tmpl` Template" on page 526
`/etc/ppp/options.ttya.tmpl`	`/etc/ppp/options.`*ttyname*	"`options.ttya.tmpl` Template File" on page 528
`/etc/ppp/myisp-chat.tmpl`	File with the name of your choice to contain the chat script.	"`/etc/ppp/myisp-chat.tmpl` Chat Script Template" on page 536
`/etc/ppp/peers/myisp.tmpl`	`/etc/ppp/peers/`*peer-name*	"`/etc/ppp/peers/myisp.tmpl` Template File" on page 532

If you decide to use one of the template files, be sure to rename it to its equivalent PPP configuration file. The one exception is the chat file template `/etc/ppp/myisp-chat.tmpl`. You can give chat scripts any names that you want.

Configuring Devices on the Dial-out Machine

The first task for setting up a dial-out PPP machine is to configure the devices on the serial line: the modem and serial port.

Note – Tasks that apply to a modem usually apply to an ISDN TA.

Before performing the next procedure, you must have done the following.

- Installed the Solaris 9 operating environment on the dial-out machine
- Determined the optimum modem speed
- Decided which serial port to use on the dial-out machine

- Obtained the root password for the dial-out machine

For planning information, see Table 29–2.

▼ How to Configure the Modem and Serial Port (Dial-out Machine)

1. **Program the modem.**

   Even though a variety of modem types is available, most modems are shipped with the correct settings for Solaris PPP 4.0. The following table lists basic settings for modems that use Solaris PPP 4.0.

TABLE 30–3 Modem Settings for Dial-Up PPP

Parameter	Setting
DCD	Follow carrier
DTR	Low so that the modem hangs up (puts the modem on-hook)
Flow Control	RTS/CTS for full-duplex hardware flow control
Attention Sequences	Disable

   If you have problems in setting up the link and suspect that the modem is at fault, first consult the modem manufacturer's documentation. Also, a number of sites on the World Wide Web offer help with modem programming. Finally, you can find some suggestions for clearing modem problems in "How to Diagnose Modem Problems" on page 511.

2. **Attach the modem cables to the serial port on the dial-out machine and to the telephone jack.**

3. **Become superuser on the dial-out machine.**

4. **Run `admintool`, as explained in "Setting Up Terminals and Modems" in *System Administration Guide: Advanced Administration*.**

   a. **Click the port where you have attached the modem, either port a or port b.**

      The Modify Serial Port window is displayed.

   b. **Specify modem direction as dial-out only.**

      Although you can set up the modem as bidirectional (the default template for `admintool`) the dial-out-only choice is more secure against possible intruders.

Note – You can set the baud rate and timeout from `admintool`, however, the `pppd` daemon ignores these settings.

5. **Click Okay to convey the changes.**

Configuring Communications on the Dial-out Machine

The procedures in this section show how to configure communications over the serial line of the dial-out machine. Before you can use these procedures, you must have configured the modem and serial port, as described in "How to Configure the Modem and Serial Port (Dial-out Machine)" on page 460.

The next tasks show how to enable the dial-out machine to successfully initiate communications with the dial-in server, based on options that are defined in the PPP configuration files. You need to create the following files:

- `/etc/ppp/options`
- `/etc/ppp/options.`*ttyname*
- Chat script
- `/etc/ppp/peers/`*peer-name*

Solaris PPP 4.0 provides templates for the PPP configuration files, which you can tailor to suit your needs. Refer to"Dialup PPP Template Files" on page 459 for detailed information about these files.

▼ How to Define Communications Over the Serial Line

1. **Become superuser on the dial-out machine.**
2. **Create a file called `/etc/ppp/options` with the following entry:**

```
lock
```

The `/etc/ppp/options` file is used for defining global parameters that apply to all communications by the local machine. The `lock` option enables UUCP-style locking of the form `/var/spool/locks/LK.xxx.yyy.zzz`.

Note – If the dial-out machine does not have an `/etc/ppp/options` file, only the superuser can run the `pppd` command. However, the `/etc/ppp/options` can be empty.

For a complete description of `/etc/ppp/options`, refer to "`/etc/ppp/options` Configuration File" on page 526.

3. **[Optional]Create a file called** `/etc/ppp/options.`*ttyname* **for defining how communications should be initiated from a specific serial port**

   The next example shows an `/etc/ppp/options.`*ttyname* file for the port with the device name `/dev/cua/a`.

   ```
   # vi /etc/ppp/options.cua.a
   crtscts
   ```

   The PPP option `crtscts` tells the `pppd` daemon to turn on hardware flow control for serial port a.

   For more information about the `/etc/ppp/options.ttyname` file, go to "`/etc/ppp/options.`*ttyname* Configuration File" on page 527.

4. **Set the modem speed, as described in "How to Set the Modem Speed" on page 466.**

▼ How to Create the Instructions for Calling a Peer

Before the dial-out machine can initiate a PPP link, you must collect information about the dial-in server to become the peer. Then you use this information to create the chat script, which describes the actual conversation between the dial-out machine and the peer.

1. **Determine the speed at which the dial-out machine's modem needs to run.**

   For more information, see "Configuring the Modem Speed" on page 533.

2. **Obtain the following information from the dial-in server's site:**

   - Server's telephone number
   - Authentication protocol that is used, if appropriate
   - Login sequence that is required by the peer for the chat script

3. **Obtain the names and IP addresses of name servers at the dial-in server's site.**

4. **Put instructions for initiating calls to the particular peer in a chat script.**

   For example, you might create the following chat script, `/etc/ppp/mychat`, to call the dial-in server `myserver`.

   ```
   SAY "Calling the peer\n"
           TIMEOUT 10
   ```

```
ABORT BUSY
ABORT 'NO CARRIER'
ABORT ERROR
REPORT CONNECT
"" AT&F1&M5S2=255
TIMEOUT 60
OK ATDT1-123-555-1234
CONNECT \c
SAY "Connected; logging in.\n"
TIMEOUT 5
ogin:--ogin: pppuser
TIMEOUT 20
ABORT 'ogin incorrect'
ssword: \qmypassword
"% " \c
SAY "Logged in.  Starting PPP on peer system.\n"
ABORT 'not found'
"" "exec pppd"
~ \c
```

The script contains instructions for calling a Solaris dial-in server that requires a login sequence. For a description of each instruction, refer to "Basic Chat Script Enhanced for a UNIX-Style Login" on page 538. For complete details on creating a chat script, read the section "Defining the Conversation on the Dial-up Link" on page 534.

Note – You do not invoke the chat script directly, Rather, you use the file name of the chat script as an argument to the `connect` option, which invokes the script.

If a peer runs Solaris or a similar UNIX-based operating system, consider using the previous chat script as a template for your dial-out machines.

▼ How to Define the Connection With an Individual Peer

1. **Become superuser on the dial-out machine.**
2. **Update DNS databases by creating the following `/etc/resolv.conf` file:**

```
domain bigcompany.com
nameserver 10.10.111.15
nameserver 10.10.130.8
```

:

- **`domain bigcompany.com`** – Specifies that the peer's DNS domain is `bigcompany.com`.
- **`nameserver 10.10.111.15`** and **`nameserver 10.10.130.8`** – Lists the IP addresses of name servers at bigcompany.com.

For complete details on DNS implementation, refer to "Setting Up DNS Service" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

3. **Edit the** `/etc/nsswitch.conf` **file to have the DNS database searched first for host information.**

```
hosts:      dns [NOTFOUND=return] files
```

4. **Create the** `/etc/ppp/peers` **directory, and then add a file for the peer.**

   For example, you would create the following file to define the dial-in server `myserver`:

```
# cd /etc/ppp
# mkdir peers
# cd peers
# vi myserver
/dev/cua/a
57600
noipdefault
defaultroute
idle 120
noauth
connect "chat -U 'mypassword' -T 1-123-555-1213 -f /etc/ppp/mychat"
```

   - **`/dev/cua/a`** – Specifies that the device `/dev/cua/a` should be used as the serial interface for calls to `myserver`.
   - **`57600`** – Defines the speed of the link.
   - **`noipdefault`** – Specifies that for transactions with peer `myserver`, the dial-out machine initially has an IP address of 0.0.0.0. `myserver` assigns an IP address to the dial-out machine for every dial-up session.
   - **`idle 120`** – Indicates that the link is to time out after it is idle for 120 seconds.
   - **`noauth`** – Specifies that the peer `myserver` does not need to provide authentication credentials when negotiating the connection with the dial-out machine..
   - **`connect "chat -U 'mypassword' -T 1-123-555-1213 -f /etc/ppp/mychat"`** – Specifies the `connect` option and its arguments, including the phone number of the peer, and the chat script `/etc/ppp/mychat` with calling instructions.

Where to Go From Here

Task	For Instructions
Configure another dial-out machine	"How to Configure the Modem and Serial Port (Dial-out Machine)" on page 460

Task	For Instructions
Test modem connectivity by dialing out to a another computer	`cu`(1C) and `tip`(1) man pages. These utilities can help you test if your modem is properly configured and can establish a connection with another machine.
Get detailed information about the PPP configuration files	"Using PPP Options in Files and on the Command Line" on page 521.
Begin configuring the dial-in server	"Configuring Devices on the Dial-in Server" on page 466

Configuring the Dial-in Server

The tasks in this section are for configuring the dial-in server, the peer machine that receives the call over the PPP link from the dial-out machine. The tasks show how to configure the dial-in server `myserver` introduced in Figure 29–1.

Tasks for Configuring the Dial-in Server (Task Map)

TABLE 30–4 Task Map for Setting Up the Dial-in Server

Task	Description	For Instructions
1 Gather preconfiguration information.	Gather data that is needed prior to setting up the link, such as peer host names, target phone numbers, and modem speed.	"Planning a Dial-up PPP Link" on page 440.
2 Configure the modem and serial port.	Set up the modem and serial port.	"How to Configure the Modem and Serial Port (Dial-in Server)" on page 466
3 Configure calling peer information.	Set up the user environments and PPP options for every dial-out machine permitted to call the dial-in server.	"How to Configure Users of the Dial-in Server" on page 467.
4 Configure the serial line communication.	Configure the characteristics of the transmission across the serial line.	"How to Define Communications Over the Serial Line (Dial-in Server)" on page 469.

Configuring Devices on the Dial-in Server

The following procedure explains how to configure the modem and serial port on the dial-in server.

Before you do the next procedure, you must have completed the following activities on the peer dial-in server:

- Installed the Solaris 9 operating environment
- Determined the optimum modem speed
- Decided which serial port to use

How to Configure the Modem and Serial Port (Dial-in Server)

1. **Program the modem, as instructed in the modem manufacturer's documentation.**

   For other suggestions, refer to "How to Configure the Modem and Serial Port (Dial-out Machine)" on page 460

2. **Attach the modem to the serial port on the dial-in server.**

3. **Become superuser on the dial-in server.**

4. **Configure the serial port by using `admintool`, as described in "Setting Up Terminals and Modems" in *System Administration Guide: Advanced Administration*.**

   Use `admintool` to do the following:

   a. **Select the serial port where you have attached the modem, either port a or port b.**

      The Modify Serial Port window is displayed.

   b. **Specify modem direction as dial-in only.**

      Note – Solaris PPP 4.0 does support bidirectional communications for a modem.

      ---

   c. **Click Okay to convey the changes.**

▼ How to Set the Modem Speed

The next procedure explains how to set the modem speed for a dial-in server. For suggestions on speeds to use with Sun Microsystems' computers, see "Configuring the Modem Speed" on page 533.

1. **Log in to the dial-in server**
2. **Use the tip command to reach the modem.**
   Instructions for using `tip` to set the modem speed are in the `tip`(1) man page.
3. **Configure the modem for a fixed DTE rate.**
4. **Lock the serial port to that rate, using `ttymon` or `admintool`, as discussed in "Setting Up Terminals and Modems" in** *System Administration Guide: Advanced Administration***.**

Where to Go From Here

Task	For Instructions
Configure another serial port and modem on the dial-in server	"How to Configure the Modem and Serial Port (Dial-in Server)" on page 466
Configure information about users that call the dial-in server	"How to Configure Users of the Dial-in Server" on page 467

Setting Up Users of the Dial-in Server

Part of the process of setting up a dial-in server involves configuring information about each known remote caller.

Before starting the procedures in this section, you must have done the following:

- Obtained the UNIX user names for all users who are permitted to log in from remote dial-out machines.
- Set up the modem and serial line, as described in "How to Configure the Modem and Serial Port (Dial-in Server)" on page 466..
- Dedicated an IP address to be assigned to incoming calls from remote users. Though optional, a dedicated IP address for all calls is useful when the number of potential callers exceeds the number of modems and serial ports on the dial-in server. For complete information about creating dedicated IP addresses, go to "Creating an IP Addressing Scheme for Callers" on page 550.

▼ How to Configure Users of the Dial-in Server

1. **Become superuser on the dial-in server.**

2. **Create a new account on the dial-in server for each remote PPP user.**

   You can use `admintool` or the Solaris Management Console to create a new user. For instructions for creating a new user through Solaris Management Console, see "Setting Up User Accounts (Task Map)" in *System Administration Guide: Basic Administration*. For instructions for creating a new user through `admintool`, see `admintool`(1M).

Note – The remaining steps show how to create an account using `admintool`. You can use the same parameters for creating an account with Solaris Management Console.

---

3. **Use the Add User template to create the new user.**

   For example, the next table shows how you might fill out PPP-related parameters for an account called `pppuser` for `user1` on the dial-out machine `myhome`.

Template Parameter	Value	Definition
User Name	`pppuser`	The user account name for the remote user. This account name should correspond to the account name that is given in the login sequence of the chat script. For example, `pppuser` is the account name found the chat script in "How to Create the Instructions for Calling a Peer" on page 462 .
Login Shell	`/usr/bin/pppd`	The default login shell for the remote user. The login shell `/usr/bin/pppd` initially restricts the caller to a dedicated PPP environment.
Create Home Dir Path	`/export/home/pppuser`	The home directory `/export/home/pppuser` is set when the caller successfully logs in to the dial-in server.

4. **Create for each caller a `$HOME/.ppprc` file that contains various options that are specific to the user's PPP session.**

   For example, you might create the following `.ppprc` file for `pppuser`.

```
#cd /export/home/pppuser
#vi .ppprc
noccp
```

   where `noccp` turns off compression control on the link

Where to Go From Here

Task	For Instructions
Set up more users of the dial-in server	"How to Configure Users of the Dial-in Server" on page 467
Configure communications over the dial-in server	"How to Define Communications Over the Serial Line (Dial-in Server)" on page 469

Configuring Communications Over the Dial-in Server

The next task shows how to enable the dial-in server to open communications with any dial-out machine, based on options defined in the following PPP configuration files:

- `/etc/ppp/options`
- `/etc/ppp/options.`*ttyname*

For detailed information about these files, refer to "Using PPP Options in Files and on the Command Line" on page 521.

Before you proceed, you should have done the following:

- Configured the serial port and modem on the dial-in server, as described in "How to Configure the Modem and Serial Port (Dial-in Server)" on page 466.
- Configured information about the prospective users of the dial-in server, as described in "How to Configure Users of the Dial-in Server" on page 467.

How to Define Communications Over the Serial Line (Dial-in Server)

1. **Become superuser on the dial-in server.**
2. **Create the `/etc/ppp/options` file with the following entry.**

   ```
   nodefaultroute
   ```

   where `nodefaultroute` indicates that no route is defined for the server.

Note – If the dial-in server does not have an `/etc/ppp/options` file, only the superuser can run the `pppd` command. However, the `/etc/ppp/options` file can be empty.

3. **Create the file `/etc/options.`*ttyname* to define how calls that are received over serial port *ttyname* should be handled.**

   The following `/etc/options.ttya` file defines how the dial-in server's serial port `/dev/ttya` should handle incoming calls.

```
:10.0.0.80
xonxoff
```

   - **`:10.0.0.80`** – Assigns the IP address 10.0.0.80 to all peers calling in over serial port `ttya`.
   - **`xonxoff`** – Allows the serial line to handle communications from modems with software flow control enabled.

Where to Go From Here

If you have followed all the procedures in this chapter, you have completed the configuration of the dial-up link.

Task	For Instructions
Test modem connectivity by dialing out to another computer	`cu`(1C) and`tip`(1) man pages. These utilities can help you test if your modem is properly configured and can establish a connection with another machine.
Configure more options for the dial-in server	"Configuring the Dial-in Server" on page 465
Configure more dial-out machines	"Configuring the Dial-out Machine" on page 458
Have the remote machine call the dial-in server	"Calling the Dial-in Server" on page 470

Calling the Dial-in Server

You establish a dial-up PPP link by having the dial-out machine call the dial-in server. You can instruct the dial-out machine to call the server by specifying the `demand`

option in the PPP configuration files on the dial-out machine. But the most common method for establishing the link is for the user to run the `pppd` command on the dial-out machine.

Before you proceed to the next task, you should have done either or both of the following:

- Set up the dial-out machine, as described in "Configuring the Dial-out Machine" on page 458
- Set up the dial-in server, as described in "Configuring the Dial-in Server" on page 465

▼ How to Call the Dial-In Server

1. **Log in to the dial-out machine by using your regular user account, not `root`.**
2. **Call the dial-in server by running the `pppd` command.**

   For example, the following command initiates a link between the dial-out machine and dial-in server `myserver`:

   ```
   % pppd call myserver
   ```

   - **`pppd`** – Starts the call by invoking the `pppd` daemon
   - **`57600`** – Sets the speed of the line between host and modem
   - **`call myserver`** – Invokes the `call` option of `pppd`. `pppd` then reads options in the file `/etc/ppp/peers/myserver`, which was created in "How to Define the Connection With an Individual Peer" on page 463.
3. **Contact a host on the server's network, for example the host `lindyhop` shown in Figure 29–1:**

   ```
   ping lindyhop
   ```

   If the link is working correctly, the standard Telnet login sequence should be displayed in the terminal window. If the link is not working correctly, refer to Chapter 34.
4. **Terminate the PPP session:**

   ```
   % pkill -TERM -x pppd
   ```

Where to Go From Here

If you have followed all the procedures in this chapter, you have completed the configuration of the dial-up link.

Task	For Instructions
Have users start working on their dial-out machines.	"How to Call the Dial-In Server" on page 471
Fix problems on the link	Chapter 34
Learn more about the files and options that are used in this chapter.	"Using PPP Options in Files and on the Command Line" on page 521

CHAPTER 31

Setting Up a Leased-Line PPP Link (Tasks)

This chapter explains how to configure a PPP link that uses a leased line between peers. Major sections include:

Setting Up a Leased Line (Task Map)

Leased-line links are relatively easy to set up, in comparison to dial-up links. In most instances, you do not have to configure the CSU/DSU, dialing services, or authentication. If you do need to configure the CSU/DSU, refer to the manufacturer's documentation for aid with this complex task.

The task map in the next table describes all the tasks involved in setting up the basic leased-line link.

Note – Some types of leased lines, such as Frame Relay that uses Switched Virtual Circuits (SVCs) or Switched 56 service, do require the CSU/DSU to "dial" the address of the opposite peer.

TABLE 31–1 Task Map for Setting Up the Leased Line Link

Task	Description	For Instructions
Gather preconfiguration information	Gather data needed prior to setting up the link.	Table 29–4

TABLE 31–1 Task Map for Setting Up the Leased Line Link *(Continued)*

Task	Description	For Instructions
Set up the leased line hardware	Assemble the CSU/DSU and synchronous interface card.	"How to Configure Synchronous Devices" on page 474
Configure the interface card, if required	Configure the interface script to be used when the leased line is brought up.	"How to Configure Synchronous Devices" on page 474
Configure information about the remote peer	Define how communications between your local machine and the remote peer should work.	"How to Configure a Machine on a Leased-Line" on page 476
Start up the leased line	Configure your machine so that it starts up PPP over the leased line as part of the booting process.	"How to Configure a Machine on a Leased-Line" on page 476

Configuring Synchronous Devices on the Leased Line

The task in this section involves configuring equipment that is required by the leased-line topology that is introduced in"Example—Configuration for a Leased-Line Link" on page 445 . The synchronous devices that are required to connect to the leased line include the interface and modem.

Prerequisites for Synchronous Devices Setup

Before you perform the next procedure, you must have the following items:

- Working leased line installed at your site by the provider
- Synchronous unit (CSU/DSU)
- Solaris 9 operating environment release installed on your system
- Synchronous interface card of the type that is required by your system

▼ How to Configure Synchronous Devices

1. **Physically install the interface card into the local machine, if it is necessary.**
   Follow the instructions in the manufacturer's documentation.

2. **Connect the cables from the CSU/DSU to the interface and, if it is necessary, from the CSU/DSU to the leased-line jack or similar connector.**

3. **Configure the CSU/DSU, as instructed in the documentation from the manufacturer or network provider.**

Note – The provider from whom you rented the leased line might supply and configure the CSU/DSU for your link.

---

4. **Configure the interface card, if necessary, as instructed in the interface documentation.**

   The configuration of the interface card involves the creation of a startup script for the interface. The router at LocalCorp in the leased-line configuration shown in Figure 29–2 uses an HSI/S interface card.

   The following script `hsi-conf`, starts up the HSI/S interface:

```
#!/bin/ksh
/opt/SUNWconn/bin/hsi_init hih1 speed=1536000 mode=fdx loopback=no \
nrzi=no txc=txc rxc=rxc txd=txd rxd=rxd signal=no 2>&1 > /dev/null
```

   - `hih1` – Indicates that HSI/S is the synchronous port used,
   - `speed=1536000` – Sets the speed of the CSU/DSU to 1536000.

Where to Go From Here

Task	For Instructions
Configure the local machine on the leased line	"How to Configure a Machine on a Leased-Line" on page 476

Configuring a Machine on the Leased Line

The task in this section explains how to set up a router to function as the local peer on your end of a leased line. The task uses the leased line that was introduced in "Example—Configuration for a Leased-Line Link" on page 445 as an example.

Prerequisites for Configuring the Local Machine on a Leased Line

Before you perform the next procedure, you must have completed the following:

- Set up and configured the synchronous devices for the link, as described in "Configuring Synchronous Devices on the Leased Line " on page 474.
- Obtained the root password for the local machine on the leased line.
- Set up the local machine to run as a router on the network(s) to use the services of the leased-line provider.

▼ How to Configure a Machine on a Leased-Line

1. **Become superuser on the local machine (router).**
2. **Add an entry for the remote peer in the router's** `/etc/hosts` **file.**

```
# vi /etc/hosts
#
# Internet host table
#
127.0.0.1       localhost
192.168.130.10  local2-peer        loghost
192.168.130.11  local1-net
10.0.0.25     farISP
```

   The sample `/etc/hosts` file is for the local router at the fictitious LocalCorp. Note the IP address and host name for the remote peer `farISP` at the service provider.

3. **Create the file** `/etc/ppp/peers/`*peer-name* **to hold information about the provider's peer.**

   For the sample leased-line link you create the file `/etc/ppp/peers/farISP`.

```
#vi /etc/ppp/peers/farISP
init '/etc/ppp/conf_hsi'
local
/dev/hih1
sync
noauth
192.168.130.10:10.0.0.25
nodefaultroute
passive
persist
noccp
nopcomp
novj
noaccomp
```

The following table explains the options and parameters that are used in /etc/ppp/peers/farISP.

Option	Definition
init '/etc/ppp/conf_hsi'	Starts up the link and configures the HSI interface, using the parameters in the script /etc/ppp/conf_hsi.
local	Tells the pppd daemon not to change the state of the Data Terminal Ready (DTR) signal and to ignore the Data Carrier Detect (DCD) input signal.
/dev/hih1	Gives the device name of synchronous interface.
sync	Establishes synchronous encoding for the link.
noauth	Disables authentication on the link.
192.168.130.10:10.0.0.25	Defines the IP addresses of the local peer and the remote peer, separated by a colon.
passive	Tells the pppd daemon on the local machine go quiet after issuing maximum number of LCP Configure-Requests and wait for the peer to start.
persist	Tells the pppd daemon to try to restart the link after a connection ends.
noccp, nopcomp, novj, noaccomp	Disables the Compression Control Protocol (CCP), Protocol Field compression, Van Jacobson compression, and address and control field compression, respectively. Though these forms of compression speed up transmissions on a dial-up link, they might slow down a leased line.

4. **Create an initialization script that is called** demand**, which creates the PPP link as part of the booting process.**

```
# cd /etc/ppp/
# vi demand
if [ -f /var/run/ppp-demand.pid ] &&
           /usr/bin/kill -s 0 `/bin/cat /var/run/ppp-demand.pid`
        then
                :
        else
                /usr/bin/pppd call farISP
        fi
```

The demand script contains the pppd command for establishing a leased-line link. The following table explains the contents of $PPPDIR/demand.

Code Sample	Explanation
echo "Starting Solaris PPP 4.0\c"	Displays "Starting Solaris PPP 4.0" during the booting process.

Code Sample	Explanation

`if ps -e | grep '\<pppd\ > /dev/null 2>&1 ; then echo "\npppd daemon is still running"`
`echo "or in the process of exiting"`
`exit 0`
	Search for an already existing `pppd` daemon.
	If `pppd` is found, then send out a message and exit the `demand` script.

`echo "\nEstablishing PPP session...\n"`	Display "Establishing PPP session" during booting.
`/usr/bin/pppd call farISP`	Run the `pppd` command, using the options in `/etc/ppp/peers/farISP`.

The Solaris PPP 4.0 startup script `/etc/rc2.d/S47pppd` invokes the `demand` script as part of the Solaris booting process. The following lines in `/etc/rc2.dS47pppd` search for the presence of a file that is called `$PPPDIR/demand`.

```
if [ -f $PPPDIR/demand ]; then
            . $PPPDIR/demand
    fi
```

If `$PPPDIR/demand` is found, it is executed. During the course of executing `$PPPDIR/demand`, the link is established.

Where to Go From Here

If you have followed all the procedures in this chapter, you have completed the configuration of the leased-line link.

Task	For Instructions
Instruct users to start communicating with machines on the Internet or other network served by the remote peer.	Have users run `telnet`, `ftp`, `rsh`, or similar commands to reach machines outside the local network.
Fix problems on the link.	"Fixing Leased-Line Problems" on page 515 for troubleshooting information.
Learn more about the files and options used in this chapter.	"Using PPP Options in Files and on the Command Line" on page 521

CHAPTER 32

Setting Up Authentication (Tasks)

This chapter contains tasks for setting up PPP authentication. Subjects that are covered include:

- "Configuring PAP Authentication " on page 480
- "Configuring CHAP Authentication" on page 487

The procedures show how to implement authentication over a dial-up link because dial-up links are more likely to be configured for authentication than leased-line links. However, if your corporate security policy requires it, you can configure authentication over leased lines. For leased-line authentication, use the tasks in this chapter as guidelines.

If you want to use PPP authentication but are not sure which protocol to use, review the section "Why Use PPP Authentication?" on page 434. More detailed information about PPP authentication is in the`pppd`(1M) man page and in "Authenticating Callers on a Link" on page 543.

Configuring PPP Authentication (Task Map)

This section contains task maps to help you quickly access procedures for PPP authentication.

TABLE 32–1 Task Map for General PPP Authentication

Task	For Information
Configure PAP authentication	"Setting Up PAP Authentication (Task Maps)" on page 480
Configure CHAP authentication	"Setting Up CHAP Authentication (Task Maps)" on page 488

Configuring PAP Authentication

The tasks in this section explain how to implement authentication on a PPP link by using the Password Authentication Protocol (PAP). The tasks use the example that is shown in "Example—PPP Authentication Configurations " on page 448 to illustrate a working PAP scenario for a dial-up link. Use the instructions as the basis for implementing PAP authentication at your site.

Before you perform the next procedures, you must have done the following:

- Set up and tested the dial-up link between the dial-in server and dial-out machines belonging to trusted callers
- Ideally, for dial-in server authentication, obtained superuser permission for the machine where the network password database (in LDAP, NIS, NIS+, or local files) is administered
- Obtained superuser authority for the local machine, either dial-in server or dial-out machine

Setting Up PAP Authentication (Task Maps)

Use the next task maps to quickly access PAP-related tasks for the dial-in server and trusted callers on dial-out machines.

TABLE 32–2 Task Map for PAP Authentication (Dial-in Server)

Task	Description	For Information
1. Gather preconfiguration information	Collect data, such as user names, that is needed for authentication	"Planning for Authentication on a Link" on page 447
2. Update the password database, if necessary	Ensure that all potential callers are in the server's password database	"How to Create a PAP Credentials Database (Dial-in Server)" on page 481

TABLE 32–2 Task Map for PAP Authentication (Dial-in Server) *(Continued)*

Task	Description	For Information
3. Create the PAP database	Create security credentials for all prospective callers in `/etc/ppp/pap-secrets`	"How to Create a PAP Credentials Database (Dial-in Server)" on page 481
4. Modify the PPP configuration files	Add options specific to PAP to the `/etc/ppp/options` and `/etc/ppp/peers/`*peer-name* files	"How to Add PAP Support to the PPP Configuration Files (Dial-in Server)" on page 483

TABLE 32–3 Task Map for PAP Authentication (Dial-out Machine)

Task	Description	For Information
1 Gather preconfiguration information	Collect data, such as user names that is needed for authentication	"Planning for Authentication on a Link" on page 447
2 Create the PAP database for the trusted caller's machine	Create the security credentials for the trusted caller and, if necessary, security credentials for other users who call the dial-out machine, in `/etc/ppp/pap-secrets`	"How to Configure PAP Authentication Credentials for the Trusted Callers" on page 484
3 Modify the PPP configuration files	Add options specific to PAP to the `/etc/ppp/options` and `/etc/ppp/peers/`*peer-name* files	"How to Add PAP Support to the PPP Configuration Files (Dial-out Machine)" on page 486

Configuring PAP Authentication on the Dial-in Server

To set up PAP authentication you must do the following:

- Create a PAP credentials database
- Modify PPP configuration files for PAP support

▼ How to Create a PAP Credentials Database (Dial-in Server)

This procedure modifies the `/etc/ppp/pap-secrets` file, which contains the PAP security credentials that are used to authenticate callers on the link. `/etc/ppp/pap-secrets` must exist on both machines on a PPP link.

The sample PAP configuration that was introduced in Figure 29–3 uses the `login` option of PAP. If you plan to use this option, you might also need to update your

network's password database. For more information on the `login` option, refer to "Using the `login` Option With `/etc/ppp/pap-secrets`" on page 547.

1. **Assemble a list of all potential trusted callers—people to be granted permission to call the dial-in server from their remote machines.**
2. **Verify that each trusted caller already has a UNIX user name and password in the dial-in server's password database.**

Note – This is particularly important for the sample PAP configuration, which uses `login` option of PAP to authenticate callers. If you choose not to implement `login` for PAP, the callers' PAP user names do not have to correspond to their UNIX user names. For information on standard `/etc/ppp/pap-secrets`, refer to "`/etc/ppp/pap-secrets` File" on page 544.

Do the following if a potential trusted caller does not have a UNIX user name and password:

a. **For callers that you do not know, confirm with their managers or other system administrators that these remote users are permitted to access the dial-in server.**
b. **Create UNIX user names and passwords for these callers in the manner that is directed by your corporate security policy.**

3. **Become superuser on the dial-in server, and edit the `/etc/ppp/pap-secrets` file.**

   Solaris PPP 4.0 provides a `pap-secrets` file in `/etc/ppp` that contains comments regarding how to use PAP authentication but no options. You can add the following options at the end of the comments.

```
#
user1      myserver        ""          *
user2      myserver        ""          *
myserver   user2           serverpass  *
```

   To use the `login` option of `/etc/ppp/pap-secrets`, you must type the UNIX user name of each trusted caller. Wherever a set of double quotes ("") appears in the third field, the password for the caller is looked up in the server's password database.

   The entry `myserver * serverpass *` contains the PAP user name and password for the dial-in server. In Figure 29–3, the trusted caller `user2` requires authentication from remote peers. Therefore, `myserver`'s `/etc/ppp/pap-secrets` file contains PAP credentials for use when establishing a link with `user2`.

Where to Go From Here

Task	For Instructions
Modify the PPP configuration files to support PAP authentication	"Modifying the PPP Configuration Files for PAP (Dial-in Server)" on page 483
Set up PAP authentication on the dial-out machines of trusted callers	"Configuring PAP Authentication for Trusted Callers (Dial-out Machines)" on page 484

Modifying the PPP Configuration Files for PAP (Dial-in Server)

The tasks in this section explain how to update existing PPP configuration files to support PAP authentication on the dial-in server.

▼ How to Add PAP Support to the PPP Configuration Files (Dial-in Server)

The procedure uses the PPP configuration files that were introduced in "How to Define Communications Over the Serial Line (Dial-in Server)" on page 469 as examples.

1. **Log in to the dial-in server as superuser.**
2. **Add authentication options to the** `/etc/ppp/options` **file.**

   For example you would add the options in bold to an existing `/etc/ppp/options`file to implement PAP authentication:

   ```
   lock
   idle 120
   nodefaultroute
   name myserver
   auth
   require-pap
   user myserver
   remotename user2
   login
   ```

   - **`name myserver`** – Sets `myserver` as the PAP name of the user on the local machine. If the `login` option is used, the PAP name must be the same as the user's UNIX user name in the password database.
   - **`auth`** – States that the server must authenticate callers before establishing the link

- **`require-pap`** – Requires callers to provide PAP credentials.
- **`user myserver`** – Defines `myserver` as the user name of the local machine.
- **`remotename user2`** – Defines `user2` as a peer that requires authentication credentials from the local machine.
- **`login`** – Specifies that the local machine must use the `login` option of PAP for authentication, wherever it is called for in the `/etc/ppp/pap-secrets` file.

3. **Create an `/etc/ppp/options.`*ttyname* file, as described in "How to Define Communications Over the Serial Line " on page 461.**
4. **Set up the `$HOME/.pppprc` file for each remote caller, as explained in "How to Configure Users of the Dial-in Server" on page 467.**

Where to Go From Here

Task	For Instructions
Configure PAP authentication credentials for trusted callers of the dial-in server	"Configuring PAP Authentication for Trusted Callers (Dial-out Machines)" on page 484

Configuring PAP Authentication for Trusted Callers (Dial-out Machines)

This section contains tasks for setting up PAP authentication on the dial-out machines of trusted callers. As system administrator, you can set up PAP authentication on the machines before distributing them to the prospective callers. Or, if the remote callers already have their machines, you can give them the tasks in this section.

Configuring PAP for trusted callers involves two tasks:

- Configuring the callers' PAP security credentials
- Configuring the callers' dial-out machines to support PAP authentication

▼ How to Configure PAP Authentication Credentials for the Trusted Callers

This procedure shows how to set up PAP credentials for two trusted callers, one of which requires authentication credentials from remote peers. The steps in the procedure assume that you, the system administrator, are creating the PAP credentials on the trusted callers' dial-out machines.

1. **Become superuser on a dial-out machine**

   Using the sample PAP configuration introduced in Figure 29–3, assume that the dial-out machine belongs to `user1`.

2. **Modify the pap-secrets database for the caller:**

   Solaris PPP 4.0 provides an `/etc/ppp/pap-secrets` file that contains helpful comments but no options. You can add the following options to this `/etc/ppp/pap-secrets` file.

   ```
   # user1    myserver  pass1    *
   ```

   Note that `user1`'s password `pass1` is passed in readable ASCII form over the link. `myserver` is caller `user1`'s name for the peer.

3. **Become superuser on another dial-out machine.**

   Using the PAP authentication example, assume that this dial-out machine belongs to the caller `user2`.

4. **Modify the pap-secrets database for the caller:**

   You can add the next options to the end of the existing `/etc/ppp/pap-secrets` file.

   ```
   # user2     myserver   pass2       *
   myserver  user2      serverpass  *
   ```

   In this example, `/etc/ppp/pap-secrets` has two entries. The first entry contains the PAP security credentials that `user2` passes to dial-in server `myserver` for authentication.

   `user2` requires PAP credentials from the dial-in server as part of link negotiation. Therefore, the `/etc/ppp/pap-secrets` also contains PAP credentials that are expected from `myserver` on the second line.

Note – Most ISPs do not supply authentication credentials, so the scenario just discussed is not realistic for them.

Where to Go From Here

Task	Instructions
Create PAP credentials for additional callers	"How to Create a PAP Credentials Database (Dial-in Server)" on page 481
Configure a dial-out machine to support PAP authentication	"How to Configure PAP Authentication Credentials for the Trusted Callers" on page 484

Modifying PPP Configuration Files for PAP (Dial-out Machine)

The tasks in this section explain how to update existing PPP configuration files to support PAP authentication on the dial-out machines of trusted callers.

The procedure uses the following parameters to configure PAP authentication on the dial-out machine that belongs to `user2`, who was introduced in Figure 29–3. `user2` requires incoming callers to authenticate, including calls from dial-in `myserver`.

▼ How to Add PAP Support to the PPP Configuration Files (Dial-out Machine)

This procedure uses the PPP configuration files introduced in "How to Define Communications Over the Serial Line " on page 461 as examples. The procedure configures the dial-out machine belonging to `user2`, as shown in Figure 29–3.

1. **Log in to the dial-out machine as superuser.**
2. **Modify the** `/etc/ppp/options` **file .**

   The next `/etc/ppp/options` file contains options for PAP support, which are shown in bold.

   ```
   #vi /etc/ppp/options
   lock
   nodefaultroute
   name user2
   auth
   require-pap
   ```

   - **`name user2`** – Sets `user2` as the PAP name of the user on the local machine. If the `login` option is used, the PAP name must be the same as the user's UNIX user name in the password database.
   - **`auth`** – States that the dial-out machine must authenticate callers before establishing the link
   - **`require-pap`** – Requires peers to provide PAP credentials when returning the call from the dial-out machine.
3. **Create an** `/etc/ppp/peers/`*peer-name* **file for the remote machine** `myserver`**.**

   The next sample shows how to add PAP support to the existing `/etc/ppp/peers/myserver` file created in "How to Define the Connection With an Individual Peer" on page 463.

   ```
   #cd /etc/ppp
   # mkdir peers
   # cd peers
   ```

```
# vi myserver
/dev/cua/a
57600
noipdefault
defaultroute
idle 120
user user2
remotename myserver
connect "chat -U 'mypassword' -f /etc/ppp/mychat"
```

The new options in bold add PAP requirements for peer `myserver`.

- **`user user2`** – Defines `user2` as the user name of the local machine.
- **`remotename myserver`** – Defines `myserver` as a peer that requires authentication credentials from the local machine.

Where to Go From Here

Task	For Instructions
Test the PAP authentication setup by calling the dial-in server.	Procedures for calling the dial-in server, "How to Call the Dial-In Server" on page 471.
Learn more about PAP authentication	"Password Authentication Protocol (PAP)" on page 544

Configuring CHAP Authentication

The tasks in this section explain how to implement authentication on a PPP link using the Challenge-Handshake Authentication Protocol (CHAP). The tasks use the example shown in Figure 29–4 to illustrate a working CHAP scenario for dialing up a private network. Use the instructions as the basis for implementing CHAP authentication at your site.

Before you perform the next procedures, you must have done the following:

- Set up and tested the dial-up link between the dial-in server and dial-out machines that belong to trusted callers
- Obtained superuser permission for the local machine, either dial-in server or dial-out machine

Setting Up CHAP Authentication (Task Maps)

TABLE 32–4 Task Map for CHAP Authentication (Dial-in Server)

Task	Description	For Information
1 Assign CHAP secrets to all trusted callers	Create (or have the callers create) their CHAP secrets	"How to Create a CHAP Credentials Database (Dial-in Server)" on page 489
2 Create the chap-secrets database	Add the security credentials for all trusted callers to the `/etc/ppp/chap-secrets` file	"How to Create a CHAP Credentials Database (Dial-in Server)" on page 489
3 Modify the PPP configuration files	Add options specific to CHAP to the `/etc/ppp/options` and `/etc/ppp/peers/`*peer-name* files	"How to Add CHAP Support to the PPP Configuration Files (Dial-in Server)" on page 490

TABLE 32–5 Task Map for CHAP Authentication (Dial-out Machine)

Task	Description	For Information
1 Create the CHAP database for the trusted caller's machine	Create the security credentials for the trusted caller and, if necessary, security credentials for other users who call the dial-out machine, in `/etc/ppp/chap-secrets`	"How to Create a CHAP Credentials Database (Dial-in Server)" on page 489
2 Modify the PPP configuration files	Add options specific to CHAP to the `/etc/ppp/options` file	"How to Add CHAP Support to the PPP Configuration Files (Dial-out Machine)" on page 492

Configuring CHAP Authentication on the Dial-in Server

The first task in setting up CHAP authentication is modifying the `/etc/ppp/chap-secrets` file. This file contains the CHAP security credentials, including the CHAP secret, that are used to authenticate callers on the link.

Note – UNIX or PAM authentication mechanisms do not work with CHAP. For example, you cannot use the PPP `login` option as described in "How to Create a PAP Credentials Database (Dial-in Server)" on page 481. If your authentication scenario requires PAM or UNIX-style authentication, choose PAP instead.

The next procedure implements CHAP authentication for a dial-in server in a private network. The PPP link is the only connection to the outside world. The only callers

who are allowed to access the network are individuals that have been granted permission by managers of the network, possibly including the system administrator.

▼ How to Create a CHAP Credentials Database (Dial-in Server)

1. **Assemble a list that contains the user names of all trusted callers—people who have been granted permission to call the private network.**
2. **Assign each user a CHAP secret.**

Note – Be sure to choose a good CHAP secret that is not easily guessed. No other restrictions are placed on the CHAP secret's contents.

The method for assigning CHAP secrets depends on your site's security policy. Either you have the responsibility for creating the secrets, or the callers must create their own secrets. If you are not responsible for CHAP secret assignment, be sure to get the CHAP secrets that were created by, or for, each trusted caller.

3. **Become superuser on the dial-in server, and modify the `/etc/ppp/chap-secrets` file.**

   Solaris PPP 4.0 includes an `/etc/ppp/chap-secrets` file that contains helpful comments but no options. You can add the following options for the server `CallServe` at the end of the existing `/etc/ppp/chap-secrets` file.

```
account1  CallServe   key123   *
account2  CallServe   key456   *
```

   `key123` is the CHAP secret for trusted caller `account1`. `key456` is the CHAP secret for trusted caller `account2`.

Where to Go From Here

Task	For Instructions
Create CHAP credentials for additional trusted callers	"How to Create a CHAP Credentials Database (Dial-in Server)" on page 489
Update the PPP configuration files to support CHAP	"How to Add CHAP Support to the PPP Configuration Files (Dial-in Server)" on page 490
Set up CHAP authentication on the dial-out machines of trusted callers	"Configuring CHAP Authentication for Trusted Callers (Dial-out Machines)" on page 491

Modifying the PPP Configuration Files for CHAP (Dial-in Server)

The task in this section explains how to update existing PPP configuration files to support CHAP authentication on the dial-in server.

▼ How to Add CHAP Support to the PPP Configuration Files (Dial-in Server)

1. **Log in to the dial-in server as superuser.**
2. **Modify the** `/etc/ppp/options` **file.**

   Add the options that are shown in bold for CHAP support.

   ```
   #vi /etc/ppp/options
   lock
   nodefaultroute
   name CallServe
   auth
   require-chap
   ```

   - **`name CallServe`** – Defines `CallServe` as the CHAP name of the user on the local machine (dial-in server).
   - **`auth`** – Makes the local machine authenticate callers before establishing the link
   - **`require-chap`** – Requires peers to provide CHAP credentials before the link can be established.
3. **Create the remaining PPP configuration files to support the trusted callers.**

   See "How to Configure Users of the Dial-in Server" on page 467 and "How to Define Communications Over the Serial Line (Dial-in Server)" on page 469.

Where to Go From Here

Task	Instructions
Configure CHAP authentication credentials for trusted callers	"How to Create a CHAP Credentials Database (Dial-in Server)" on page 489

Configuring CHAP Authentication for Trusted Callers (Dial-out Machines)

This section contains tasks for setting up CHAP authentication on the dial-out machines of trusted callers. Depending on your site's security policy, either you, or the trusted callers, might be responsible for setting up CHAP authentication.

If remote callers are to configure CHAP, be sure that the callers' CHAP secrets correspond with the CHAP secrets listed for them in the dial-in server's `/etc/ppp/chap-secrets` file. Then give them the tasks in this section for configuring CHAP.

Configuring CHAP for trusted callers involves two tasks:

- Creating the callers' CHAP security credentials
- Configuring the callers' dial-out machines to support CHAP authentication

▼ How to Configure CHAP Authentication Credentials for the Trusted Callers

This procedure shows how to set up CHAP credentials for two trusted callers. The steps in the procedure assume that you, the system administrator, are creating the CHAP credentials on the trusted callers' dial-out machines.

1. **Become superuser on a dial-out machine.**

   Using the sample CHAP configuration introduced in ,"Example—Configuration Using CHAP Authentication" on page 450, assume that the dial-out machine belongs to trusted caller `account1`.

2. **Modify the chap-secrets database for caller** `account1`**:**

   Solaris PPP 4.0 includes an `/etc/ppp/chap-secrets` file that has helpful comments but no options. You can add the following options to this existing `/etc/ppp/chap-secrets` file.

   ```
   # account1  CallServe   key123   *
   ```

   `CallServe` is the name for the peer that `account1` is trying to reach. `key123` is the CHAP secret to be used for links between `account1` and `CallServer`.

3. **Become superuser on another dial-out machine.**

   Assume that this machine belongs to caller `account2`.

4. **Modify the** `/etc/ppp/chap-secrets` **database for caller account2:**

   ```
   # account2  CallServe   key456   *
   ```

Now account2 has secret key456 as its CHAP credentials for use over links to peer CallServe.

Where to Go From Here

Task	For Instructions
Create CHAP credentials on the dial-out machines of trusted callers	"How to Create a CHAP Credentials Database (Dial-in Server)" on page 489
Configure a dial-out machine to support CHAP authentication	"How to Configure CHAP Authentication Credentials for the Trusted Callers" on page 491

▼ Adding CHAP to the Configuration Files (Dial-out Machine)

The next task configures the dial-out machine that belongs to caller account1, introduced in "Example—Configuration Using CHAP Authentication" on page 450.

How to Add CHAP Support to the PPP Configuration Files (Dial-out Machine)

1. **Log in to the dial-out machine as superuser.**
2. **Ensure that the** /etc/ppp/options **file has the following options**

```
#vi /etc/ppp/options
lock
nodefaultroute
```

3. **Create an /etc/ppp/peers/***peer-name* **file for the remote machine** CallServe**.**

```
# mkdir /etc/ppp/peers
# vi CallServe
/dev/cua/a
57600
noipdefault
defaultroute
idle 120
user account1
connect "chat -U 'mypassword' -f /etc/ppp/mychat"
```

The option **user account1** sets account1 as the CHAP user name to be given to CallServe. For a description of the other options in the previous file, see the similar

`/etc/ppp/peers/myserver` file in "How to Define the Connection With an Individual Peer" on page 463.

Where To Go From Here

Task	For Instructions
Test CHAP authentication by calling the dial-in server.	"How to Call the Dial-In Server" on page 471
Learn more about CHAP authentication.	"Challenge-Handshake Authentication Protocol (CHAP)" on page 547

CHAPTER 33

Setting Up a PPPoE Tunnel (Tasks)

This chapter contains tasks for setting up the participants on either end of the PPPoE tunnel: the PPPoE client and PPPoE access server. Specific topics include:

- "Major Tasks for Setting Up a PPPoE Tunnel (Task Maps)" on page 495
- "Setting Up the PPPoE Client" on page 496
- "Setting Up a PPPoE Access Server" on page 499

The tasks use the scenario introduced in "Planning for DSL Support Over a PPPoE Tunnel" on page 452 as an example. For an overview of PPPoE, refer to "Support for DSL Users Through PPPoE" on page 434.

Major Tasks for Setting Up a PPPoE Tunnel (Task Maps)

The following tables list the major tasks for configuring PPPoE clients and the PPPoE access server. To implement PPPoE at your site, you need to set up only your end of the PPPoE tunnel, either the client side or access-server side.

TABLE 33–1 Task Map for Setting Up a PPPoE Client

Task	Description	For Instructions
1. Configure an interface for PPPoE	Define the Ethernet interface to be used for the PPPoE tunnel.	"How to Configure an Interface for a PPPoE Client" on page 497
2. Configure information about the PPPoE access server	Define parameters for the access server at the service provider end of the PPPoE tunnel.	"How to Define a PPPoE Access Server Peer" on page 497

TABLE 33–1 Task Map for Setting Up a PPPoE Client *(Continued)*

Task	Description	For Instructions
3. Set up the PPP configuration files	Define the PPP configuration files for the client, if you have not done so already.	"How to Define Communications Over the Serial Line " on page 461
4. Create the tunnel	Call the access server.	step 5

TABLE 33–2 Task Map for Setting Up a PPPoE Access Server

Task	Description	For Instructions
1. Configure an interface for PPPoE	Define the Ethernet interface to be used for the PPPoE tunnel.	"How to Configure the Access Server's Interfaces for PPPoE" on page 499
2. Configure the services that the access server offers	Describe the services provided so that they can be "discovered" by prospective PPPoE clients.	"How to Provide Services to Clients of the Access Server" on page 500
3. Set up the PPP configuration files	Define the PPP configuration files for the client, if you have not done so already.	"Configuring Communications Over the Dial-in Server" on page 469
[Optional]Restrict use of an interface	Use PPPoE options and PAP authentication to restrict use of a particular Ethernet interface to certain clients.	"How to Restrict the Use of an Interface to Particular Clients" on page 501

Setting Up the PPPoE Client

To provide PPP service to client machines over a DSL modem, you must first configure PPPoE on the interface that is connected to the modem or hub. Then you need to tailor the PPP configuration files to define the access server on the opposite end of the PPPoE.

Prerequisites for Setting Up the PPPoE Client

Before you set up the PPPoE client, you must have done the following:

- Installed Solaris 8, Update 6 release or later releases on the client machines to use the PPPoE tunnel
- Contacted the service provider for information about its PPPoE access server

- Had the telephone company or service provider assemble the devices that are used by the client machines (DSL modem, splitter, and so forth), or assembled them yourself

▼ How to Configure an Interface for a PPPoE Client

1. **Become superuser on the PPPoE client.**
2. **Add the name of the Ethernet interface with the DSL connection to the** `/etc/ppp/pppoe.if` **file.**

   For example, you add the following entry to `/etc/ppp/pppoe.if` for a PPPoE client that uses `hme0` as the network interface that is connected to the DSL modem.

   ```
   hme0
   ```

   For more information about `/etc/ppp/pppoe.if`, go to "`/etc/ppp/pppoe.if` File" on page 553.
3. **Configure the interface for PPPoE use by typing:**

   ```
   # /etc/init.d/pppd start
   ```
4. **[Optional] Verify that the interface is now plumbed for PPPoE by typing:**

   ```
   # /usr/sbin/sppptun query
   hme0:pppoe
   hme0:pppoed
   ```

   You can also use the `/usr/sbin/sppptun` command to manually plumb interfaces PPPoE. For instructions, refer to "`/usr/sbin/sppptun` Command" on page 554.

▼ How to Define a PPPoE Access Server Peer

You define the access server in the `/etc/ppp/peers/`*peer-name* file. Many of the options that are used for the access server are also used to define the dial-in server in a dial-up scenario. For a detailed explanation of `/etc/ppp/peers.`*peer-name*, refer to "`/etc/ppp/peers/`*peer-name* File" on page 531.

1. **Become superuser on the PPPoE client.**
2. **Define the service provider's PPPoE access server in the** `/etc/ppp/peers/`*peer-name* **file.**

   For example, the following file, `/etc/ppp/peers/dslserve`, defines the access server `dslserve` at FarISP that are introduced in "Example—Configuration for a PPPoE Tunnel" on page 454.

   ```
   # cat /etc/ppp/peers/dslserve
   sppptun
   ```

```
plugin pppoe.so
connect "/usr/lib/inet/pppoec hme0"
noccp
noauth
user Red
password redsecret
noipdefault
defaultroute
```

For a definition of the options in this file, go to " `/etc/ppp/peers/`*peer-name* File for Defining an Access Server Peer" on page 561.

3. **Modify the other PPP configuration files on the PPPoE client.**

   a. **Configure `/etc/ppp/options` as described in the instructions for configuring a dial-out machine in "Configuring the Dial-out Machine" on page 458.**

   b. **Create an `/etc/ppp/options.sppptun` file to describe PPP options for the serial port to which the interface plumbed for PPPoE is attached.**

      You can use any options available for the `/etc/ppp/options.`*ttyname* file that is described in "`/etc/ppp/options.`*ttyname* Configuration File" on page 527. You must name the file `/etc/ppp/options.sppptun` because `sppptun` is the specified device name in the `pppd` configuration.

4. **Ensure that all users can start PPP on the client by typing:**

   ```
   # touch /etc/ppp/options
   ```

5. **Test if PPP can run over the DSL line by typing:**

   ```
   & pppd debug updetach call dslserve
   ```

   `dslserve` is the name that is given to the access server at the ISP that is shown in "Example—Configuration for a PPPoE Tunnel" on page 454. The `debug updetach` option causes debugging information to display in a terminal window.

   If PPP is running correctly, the terminal shows the link coming up. If PPP still does not run, try the following command to see if the servers are running correctly:

   ```
   # /usr/lib/inet/pppoec -i hme0
   ```

Where to Go From Here

Task	For Instructions
Configure another PPPoE client	"Setting Up the PPPoE Client" on page 496.
Learn more about PPPoE	"Creating PPPoE Tunnels for DSL Support" on page 552.
Have users of configured PPPoE clients begin running PPP over the DSL line	Instruct them to type **`pppd call ISP-server-name`** and then run an application or service.

Task	For Instructions
Troubleshoot PPPoE and PPP problems	Chapter 34
Configure a PPPoE access server	"Setting Up a PPPoE Access Server" on page 499.

Setting Up a PPPoE Access Server

If your company is a service provider, you can offer Internet and other services to clients that reach your site through DSL connections. First, you must determine which interfaces on the server to involve in the PPPoE tunnel. Then you define which services are made available to the users.

▼ How to Configure the Access Server's Interfaces for PPPoE

1. **Become superuser on the access server.**
2. **Add the name of the Ethernet interfaces dedicated to the PPPoE tunnels to the** `/etc/ppp/pppoe.if` **file.**

   For example, you would use the following `/etc/ppp/pppoe.if` file for the access server `dslserve` shown in "Example—Configuration for a PPPoE Tunnel" on page 454. .

   ```
   # cat /etc/ppp/pppoe.if
   hme1
   hme2
   ```
3. **Configure the interfaces for PPPoE use by typing:**

   ```
   # /etc/init.d/pppd start
   ```
4. **[Optional]Verify that interfaces on the server are now plumbed for PPPoE by typing:**

   ```
   # /usr/sbin/sppptun query
   hme1:pppoe
   hme1:pppoed
   hme2:pppoe
   hme2:pppoed
   ```

   The previous sample shows that interfaces `hme1` and `hme2` are currently plumbed for PPPoE.

You can also use the `/usr/sbin/sppptun` command to manually plumb interfaces for PPPoE. For instructions, refer to "`/usr/sbin/sppptun` Command" on page 554.

▼ How to Provide Services to Clients of the Access Server

1. **Become superuser on the access server.**
2. **Define global services provided by the access server in the `/etc/ppp/pppoe` file.**

   The following `/etc/ppp/pppoe` file lists the services provided by access server `dslserve`, that was shown in Figure 29–5.

   ```
   device hme1,hme2
   service internet
       pppd "proxyarp 192.168.1.1:"
   service debugging
       pppd "debug proxyarp 192.168.1.1:"
   ```

   In the file example, Internet service is announced for `dslserve`'s Ethernet interfaces `hme1` and `hme2`. Debugging is turned on for PPP links on the Ethernet interfaces.
3. **Set up the PPP configuration files in the same way that you would for a dial-in server.**

   For steps to use, see "Configuring Communications Over the Dial-in Server" on page 469.
4. **Start the `pppoed` daemon by typing:**

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

   `pppd` also plumbs the interfaces that are listed in `/etc/ppp/pppoe.if`.

▼ How to Modify an Existing `/etc/ppp/pppoe` File

1. **Become superuser on the access server.**
2. **Modify `/etc/ppp/pppoe`, as needed.**
3. **Cause the `pppoed` daemon to recognize the new services by typing:**

   ```
   # pkill -HUP pppoed
   ```

▼ How to Restrict the Use of an Interface to Particular Clients

The next procedure shows how to restrict an interface to a group of PPPoE clients. Before doing this task, you need to obtain the real Ethernet MAC addresses of the clients you are assigning to the interface.

Note – Some systems allow you to change the MAC address on the Ethernet interface. You should view this as a convenience factor, not a security measure..

Using the example that is shown in "Example—Configuration for a PPPoE Tunnel" on page 454, these steps show how to reserve one of `dslserve`'s interfaces, `hme1`, to clients at MiddleCo.

1. **Configure the access server's interfaces, as shown in "How to Configure the Access Server's Interfaces for PPPoE" on page 499.**

2. **Define services, as shown in "How to Provide Services to Clients of the Access Server" on page 500.**

3. **Create entries for clients in the server's `/etc/ethers` database.**

   Here is a sample entry for clients Red, Blue, and Yellow.

   ```
   8:0:20:1:40:30 redether
   8:0:20:1:40:10 yellowether
   8:0:20:1:40:25 blueether
   ```

   The sample assigns the symbolic names `redether`, `yellowether`, and `blueether` to the Ethernet addresses of clients Red, Yellow, and Blue. The assignment of symbolic names to the MAC addresses is optional.

4. **Restrict services that are provided on a specific interface by defining the following information in the `/etc/ppp/pppoe.`*device* file.**

   In this file, *device* is the name of the device to be defined.

   ```
   # vi /etc/ppp/pppoe.hme1
   service internet
        pppd "name dslserve-hme1"
        clients redether,yellowether,blueether
   ```

   `dslserve-hme1` is the access server's name, which is used in matching entries in the `pap-secrets` file. The `clients` option restricts the use of interface `hme1` to clients with the symbolic Ethernet names `redether`, `yellowether`, and `blueether`.

   If you did not define symbolic names for client's MAC addresses in `/etc/ethers`, you can use the numeric addresses as arguments to the `clients` option. The advantage here is that you can use wildcards.

   For example, you can specify the numeric address `clients 8:0:20:*:*:*` This address allows access only to clients that are listed in `/etc/ethers` with MAC

addresses that begin with the number 8:0:20.

5. **Create the** `/etc/ppp/pap-secrets` **file for the access server:**

```
# Red          dslserve-hme1   redpasswd     *
Blue         dslserve-hme1   bluepasswd    *
Yellow       dslserve-hme1   yellowpassd   *
```

The entries are the PAP names and passwords of clients allowed to run PPP over `dslserve`'s `hme1` interface.

For more information on PAP authentication, go to "Configuring PAP Authentication " on page 480.

Where to Go From Here

Task	For Instructions
Learn more about PPPoE	"Creating PPPoE Tunnels for DSL Support" on page 552.
Troubleshoot PPPoE and PPP problems	"Diagnosing and Fixing PPPoE Problems" on page 517
Configure a PPPoE client	"Setting Up the PPPoE Client" on page 496
Configure PAP authentication for a client	"Configuring PAP Authentication for Trusted Callers (Dial-out Machines)" on page 484
Configure PAP authentication on a server	"Configuring PAP Authentication on the Dial-in Server" on page 481

CHAPTER 34

Fixing Common Problems (Tasks)

This chapter contains information for diagnosing and troubleshooting common problems that occur with Solaris PPP 4.0. The following topics are covered:

The sources *PPP Design, Implementation, and Debugging* by James Carlson and the Australian National University's web site also have detailed advice for PPP troubleshooting. For more information, see"Trade Books" on page 425 and "Web Sites" on page 425.

Solving PPP Problems (Task Map)

Use the following task map to quickly access advice and solutions for common PPP problems.

TABLE 34–1 Task Map for Troubleshooting PPP

Task	Definition	For Instructions
Obtain diagnostic information about the PPP link	Use PPP diagnostic tools to get output for troubleshooting.	"How to Obtain Diagnostic Information From pppd" on page 505
Obtain debugging information for the PPP link	Use the `pppd debug` command to generate output for troubleshooting	"How to Turn on PPP Debugging" on page 506
Troubleshoot general problems with the network layer	Identify and fix PPP problems that are network-related by using a series of checks	"How to Diagnose Network Problems" on page 507
Troubleshoot general communications problems	Identify and fix communications problems that affect the PPP link	"How to Diagnose and Fix Communications Problems" on page 509
Troubleshoot configuration problems	Identify and fix problems in the PPP configuration files	"How to Diagnose Problems With the PPP Configuration" on page 510
Troubleshoot modem-related problems	Identify and fix modem problems	"How to Diagnose Modem Problems" on page 511
Troubleshoot chat script-related problems	Identify and fix chat script problems on a dial-out machine	"How to Obtain Debugging Information for Chat Scripts" on page 512
Troubleshoot serial line speed problems	Identify and fix line speed problems on a dial-in server	"How to Diagnose and Fix Serial Line Speed Problems" on page 515
Troubleshoot common problems for leased lines	Identify and fix performance problems on a leased line.	"Fixing Leased-Line Problems" on page 515
Troubleshoot problems related to authentication	Identify and fix problems related to the authentication databases	"Diagnosing and Fixing Authentication Problems" on page 516
Troubleshoot problem areas for PPPoE	Use PPP diagnostic tools to get output for identifying and fixing PPPoE problems	"How to Obtain Diagnostic Information for PPPoE" on page 517

Tools for Troubleshooting PPP

PPP links generally have three major areas of failure:

- Failure of the link to be established

- Poor performance of the link as it is used
- Problems that can be traced to the networks on either side of the link

The easiest way to find out if PPP works is to run a command, such as `ping` or `traceroute`, to a host on the peer's network, and observe the results. However, to monitor performance of a link that has been established or to troubleshoot a problematic link, you need to use PPP and UNIX debugging tools.

This section explains how to get diagnostic information from `pppd` and its associated log files. The remaining sections in this chapter describe common problems with PPP that you can discover and fix with the aid of the PPP troubleshooting tools.

▼ How to Obtain Diagnostic Information From pppd

The next procedure shows how to view the current operation of a link on the local machine.

1. **Become superuser on the local machine.**
2. **Run `pppd` with the serial device configured for PPP as the argument:**

   # **pppd /dev/***ttyname*

   The next examples show the resulting displays for a dial-up link and a leased line link when `pppd` runs in the foreground. If you run `pppd debug` in the background, the output produced is sent to the `/etc/ppp/connect-errors` file.

EXAMPLE 34–1 Output From a Properly-Operating Dial-up Link

```
# pppd /dev/cua/b
have route to 0.0.0.0/0.0.0.0 via 172.21.0.4
serial speed set to 230400 bps
Using interface sppp0
Connect: sppp0 <--> /dev/cua/b
sent [LCP ConfReq id=0x7b <asyncmap 0x0> <magic 0x73e981c8> <pcomp> <accomp>]
rcvd [LCP Ident id=0x79 magic=0x0 "ppp-2.4.0b1 (Sun Microsystems, Inc., Dec  6 20
00 09:36:22)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Dec  6 2000 09:36:22)
rcvd [LCP ConfRej id=0x7b <asyncmap 0x0>]
sent [LCP Ident id=0x7c magic=0x0 "ppp-2.4.0b1 (Sun Microsystems, Inc., Nov 15 20
00 09:38:33)"
sent [LCP ConfReq id=0x7d <magic 0x73e981c8> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x7d <magic 0x73e981c8> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x78 <magic 0xdd4ad820> <pcomp> <accomp>]
sent [LCP ConfAck id=0x78 <magic 0xdd4ad820> <pcomp> <accomp>]
sent [LCP Ident id=0x7e magic=0x73e981c8 "ppp-2.4.0b1 (Sun Microsystems, Inc., No
v 15 2000 09:38:33)"]
sent [IPCP ConfReq id=0x3d <addr 0.0.0.0> <compress VJ 0f 01>]
rcvd [LCP Ident id=0x7a magic=0xdd4ad820 "ppp-2.4.0b1 (Sun Microsystems, Inc., De
c  6 2000 09:36:22)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Dec  6 2000 09:36:22)
```

EXAMPLE 34–1 Output From a Properly-Operating Dial-up Link *(Continued)*

```
rcvd [IPCP ConfReq id=0x92 <addr 10.0.0.1> <compress VJ 0f 01>
sent [IPCP ConfAck id=0x92 <addr 10.0.0.1> <compress VJ 0f 01>
rcvd [IPCP ConfNak id=0x3d <addr 10.0.0.2>]]
sent [IPCP ConfReq id=0x3e <addr 10.0.0.2> <compress VJ 0f 01>]
rcvd [IPCP ConfAck id=0x3e <addr 10.0.0.2> <compress VJ 0f 01>]
local  IP address 10.0.0.2
remote IP address 10.0.0.1
```

EXAMPLE 34–2 Output From a Properly-Operating Leased-Line Link

(output to come)

▼ How to Turn on PPP Debugging

The next task shows how to use the `pppd` command to obtain debugging information.

1. **Turn on debugging for calls to a particular peer by using the following syntax of `pppd`.**

   ```
   % pppd debug call peer-name
   ```

   peer-name must be the name of a file in the `/etc/ppp/peers` directory.

2. **Create a log file to hold output from `pppd`.**

   ```
   % touch /var/log/pppdebug
   ```

3. **Add the following `syslog` facilities for `pppd` in `/etc/syslog.conf`.**

   ```
   daemon.debug;local2.debug       /var/log/pppdebug
   ```

4. **Restart `syslogd`.**

   ```
   % pkill -HUP -x syslogd
   ```

5. **View the contents of the log file.**

   ```
   % tail -f /var/log/pppdebug
   ```

   For an example of a log file, see Example 34–3.

Fixing Network Problems That Affect PPP Performance

If the PPP link comes up but few hosts on the remote network are reachable, this can indicate a network problem. This section explains how to isolate and fix network problems that affect a PPP link.

▼ How to Diagnose Network Problems

1. **Become superuser on the local machine and take down the problematic link.**
2. **Check the PPP configuration files to see if any optional protocols, such as CCP compression, are listed.**

   Remove these protocols from the configuration files.
3. **Call the remote peer with debugging turned on.**

   `% pppd debug call` *peer-name*
4. **Try to recreate the problem by using Telnet or other applications to reach the remote hosts.**

   Observe the debugging logs. If you still cannot reach remote hosts, this might indicate that the PPP problem is really network—related.
5. **Verify that the IP addresses of the remote hosts are registered Internet addresses.**

   Some organizations assign internal IP addresses that are known within the local network but cannot be routed to the Internet. If the remote hosts are within your company, you or another administrator must set up a name-to-address translation (NAT) or proxy server to reach the Internet. If the remote hosts are not within your company, you should report the problem to the remote organization
6. **Examine the routing tables.**

   a. **Check the routing tables on both the local machine and the peer.**

   b. **Check the routing tables for any routers that are in the path from the peer to the remote system and the path back to the peer.**

      Make sure that the intermediate routers have not been misconfigured. Often the problem can be found in the path back to the peer.

7. **[Optional]If the machine is a router, you can check the optional features by typing the following:**

   ```
   # ndd -set /dev/ip ip_forwarding 1
   ```

   For more information about `ndd`, refer to the `ndd`(1M) man page.

8. **Check the statistics obtained from `netstat -s` and similar tools.**

   For complete details on `netstat`, refer to the`netstat`(1M) man page.

   a. **Run statistics on the local machine.**

   b. **Call the peer.**

   c. **Observe the new statistics generated by `netstat -s`.**

   You can use the messages generated by `netstat -s` to isolate and fix the network problems shown in the next table.

TABLE 34–2 Common Network Problems That Affect PPP

Message	Problem	Solution
`IP packets not forwardable`	The local host is missing a route.	Add the missing route to the local host's routing tables .
`ICMP input destination unreachable`	The local host is missing a route.	Add the missing route to the local host's routing tables .
`ICMP time exceeded`	Two routers are forwarding the same destination address to each other, causing the packet to bounce back and forth until the time-to-live (TTL) value is exceeded.	Use `traceroute` to find the source of the routing loop, and then contact the administrator of the router in error. For information about `traceroute`, refer to the `traceroute`(1M) man page.
`IP packets not forwardable`	The local host is missing a route.	Add the missing route to the local host's routing tables .
`ICMP input destination unreachable`	The local host is missing a route.	Add the missing route to the local host's routing tables .

9. **Check the DNS configuration.**

   A faulty name service configuration causes applications to fail because IP addresses cannot be resolved.

   You can find information for fixing name service problems in "DNS Problems and Solutions" in *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

Fixing General Communications Problems

Communications problems occur when the two peers cannot successfully establish a link. Sometimes these problems are actually negotiation problems, caused by incorrectly configured chat scripts. This section contains information for clearing communications problems. For clearing negotiation problems caused by a faulty chat script, see Table 34–5.

▼ How to Diagnose and Fix Communications Problems

1. **Become superuser on the local machine and call the peer.**
2. **Call the remote peer with debugging turned on.**

   `% pppd debug call` *peer-name*

   You might need to obtain debugging information from the peer in order to fix certain communications problems.
3. **Check the resulting logs for the communications problems in the next table**

TABLE 34–3 General Communications Problems That Affect PPP

Symptom	Problem	Solution
`too many Configure-Requests`	One peer cannot hear the other peer.	Check for the following problems: 1. The machine or modem might have faulty cabling. 2. The modem configuration might have incorrect bit settings or have broken flow control. 3. The chat script might have failed. In this situation, see Table 34–5.
The `pppd debug` output shows that LCP comes up, but higher level protocols fail or show CRC errors	The asynchronous control character map (ACCM) is incorrectly set.	Set the ACCM to the standard default of FFFFFFFF.
The `pppd debug` output shows that IPCP comes up but terminates immediately.	IP addresses might be incorrectly configured .	1. Check the chat scripts to verify whether it has incorrect IP addresses. 2. If the chat script is okay, request debug logs for the peer, and check IP addresses in the peer logs.

TABLE 34–3 General Communications Problems That Affect PPP *(Continued)*

Symptom	Problem	Solution
The link exhibits very poor performance	The modem might be incorrectly configured, with flow control configuration errors, modem setup errors, and incorrectly configured DTE rates.	Check the modem configuration and adjust accordingly.

Fixing PPP Configuration Problems

Some PPP problems can be traced to problems in the PPP configuration files. This section contains information for isolating and fixing general configuration problems.

▼ How to Diagnose Problems With the PPP Configuration

1. **Become superuser on the local machine.**
2. **Call the remote peer with debugging turned on.**

   `% pppd debug call` *peer-name*
3. **Check the resulting log for the configuration problems listed in the next table.**

TABLE 34–4 Common PPP Configuration Problems

Symptom	Problem	Solution
`pppd debug` output shows Configure-Rejected during IP address negotiation.	The `/etc/ppp/peers/`*peer-name* file does not have an IP address for the peer. The peer does not provide an IP address for itself during link negotiation.	Supply an IP address for the peer on the pppd command line or in `/etc/ppp/peers/`*peer-name* using the following format: 0:10.0.0.10 0 represents the source address
`pppd debug` output shows that CCP data compression has failed and the link is dropped.	The peers' PPP compression configurations may be in conflict.	Ensure that both peers are configured to use the same compression algorithms.

Fixing Modem-Related Problems

Modems can be major problem areas for a dial-up link. The most common indicator of problems with the modem configuration is no response from the peer. But it is often difficult to determine if a link problem is, indeed, the result of modem configuration problems.

For basic modem troubleshooting suggestions, refer to "Troubleshooting Terminal and Modem Problems" in *System Administration Guide: Advanced Administration*. Modem manufacturers' documentation and web sites also contain solutions for problems with their particular equipment. This section contains more suggestions for isolating and fixing modem problems.

▼ How to Diagnose Modem Problems

The following steps help determine whether a faulty modem configuration causes link problems.

1. **Call the peer with debugging turned on, as explained in "How to Turn on PPP Debugging" on page 506.**

2. **Display the resulting** `/var/log/pppdebug` **log.**

   Either of the following symptoms in the output can indicate a faulty modem configuration:

   - No "`recvd`" messages have come from the peer.
   - The output contains LCP messages from the peer, but the link fails with too many LCP Configure-Requests sent by the local machine.

     These messages indicate that the local machine can hear the peer, but the peer cannot hear the local machine.
   - The link terminates with a SIGHUP signal.

3. **Use** `ping` **to send packets of various sizes over the link.**

   For complete details about `ping`, refer to the `ping`(1M) man page.

   If small packets are received, but larger packets are dropped, this can indicate modem problems.

4. **Check for errors on interface** `sppp0`**:**

```
% netstat -ni
Name  Mtu  Net/Dest   Address      Ipkts    Ierrs Opkts    Oerrs Collis Queue
lo0   8232 127.0.0.0  127.0.0.1    826808   0     826808   0     0      0
hme0  1500 172.21.0.0 172.21.3.228 13800032 0     1648464  0     0      0
```

```
sppp0 1500 10.0.0.2   10.0.0.1     210      0     128      0     0      0
```

If interface errors increase over time, this can indicate problems with the modem configuration.

Fixing Chat Script-Related Problems

Chat scripts are trouble-prone areas for dial-up links. This section contains a procedure for obtaining debugging information from `chat` and suggestions for clearing common problems.

▼ How to Obtain Debugging Information for Chat Scripts

1. **Become superuser on the dial-out machine.**
2. **Edit the** `/etc/ppp/peers/`*peer-name* **file for the peer to be called.**
3. **Add** `-v` **as an argument to the** `chat` **command specified in** `connect` **option.**

   `connect "/usr/bin/chat` **`-v`** `-f /etc/ppp/`*chat-script-name*`"`
4. **View chat script errors in the file** `/etc/ppp/connect-errors`**.**

   `Placeholder for error list`

   The next table lists common chat script errors and suggestions for fixing them.

TABLE 34–5 Common Chat Script Problems

Symptom	Problem	Solution
`pppd debug` output contains `Connect script failed`	Your chat script supplies a user name and ssword. `ogin:` *user-name* ssword: *password* However, the peer you want to connect to does not prompt for this information.	1. Delete the login and password from the chat script 2. Try to call the peer again. 3. If you still get the message, call the ISP and ask them for the correct login sequence.

TABLE 34–5 Common Chat Script Problems *(Continued)*

Symptom	Problem	Solution

The `/usr/bin/chat -v` log contains the message: `"expect (login:)" alarm read timed out`	Your chat script supplies a user name and ssword.
	`ogin: pppuser`
	`ssword: \q\U`
	However, the peer you want to connect to does not prompt for this information.
		1. Delete the login and password from the chat script
		2. Try to call the peer again.
		3. If you still get the message, call the ISP and ask them for the correct login sequence.

`pppd debug` output contains: `possibly looped-back`	The local machine or its peer is hanging at the command line and not running PPP, due to an incorrectly configured login name and password in the chat script.	1. Delete the login and password from the chat script 2. Try to call the peer again. 3. If you still get the message, call the ISP and ask them for the correct login sequence.
`pppd debug` output shows that LCP comes up, but the link terminates soon afterward.	The password in the chat script might be incorrect.	1. Ensure that you have the correct password for the local machine. 2. Check the password in the chat script and fix it if it is incorrect. 3. Try to call the peer again. 4. If you still get the message, call the ISP and ask them for the correct login sequence.

Text from the peer begins with a tilde (~).	Your chat script supplies a user name and ssword.
	`ogin: pppuser`
	`ssword: \q\U`
	However, the peer you want to connect to does not prompt for this information.
		1. Delete the login and password from the chat script
		2. Try to call the peer again.
		3. If you still get the message, call the ISP and ask them for the correct login sequence.

The modem hangs.	Your chat script contains the following line to make the local machine wait for the `CONNECT` message from the peer: `CONNECT ”`	Use the following line when you want the chat script to wait for CONNECT from the peer: `CONNECT /c` End the chat script with `~ /c`.

TABLE 34–5 Common Chat Script Problems *(Continued)*

Symptom	Problem	Solution
`pppd debug` output contains: `LCP: timeout sending Config-Requests`	Your chat script contains the following line to make the local machine wait for the `CONNECT` message from the peer: `CONNECT ″`	Use the following line when you want the chat script to wait for CONNECT from the peer: `CONNECT /c` End the chat script with `~ /c` .
pppd debug output contains: `Serial link is not 8-bit clean`	Your chat script contains the following line to make the local machine wait for the`CONNECT` message from the peer: `CONNECT ″`	Use the following line when you want the chat script to wait for CONNECT from the peer: `CONNECT /c` End the chat script with `~ /c`.
pppd debug output contains: `Loopback detected`	Your chat script contains the following line to make the local machine wait for the`CONNECT` message from the peer: `CONNECT ″`	Use the following line when you want the chat script to wait for CONNECT from the peer: `CONNECT /c` End the chat script with `~ /c`.
`pppd debug` output contains: `SIGHUP`.	Your chat script contains the following line to make the local machine wait for the`CONNECT` message from the peer: `CONNECT ″`	Use the following line when you want the chat script to wait for CONNECT from the peer: `CONNECT /c` End the chat script with `~ /c`.

Fixing Serial Line Speed Problems

Dial-in servers can experience problems due to conflicting speeds set. The next procedure helps you isolate the cause of the link problem to conflicting serial-line speeds.

The following are typical causes for speed problems:

- You invoked PPP through a program like `/bin/login` and specified the speed of the line.
- You started PPP from `mgetty` and accidentally supplied the bit rate.

pppd changes the speed originally set for the line to the speed set by `/bin/login` or `mgetty`, which causes the line to fail.

▼ How to Diagnose and Fix Serial Line Speed Problems

1. **Log in to the dial-in server and call the peer with debugging turned on.**
   If you need instructions, see "How to Turn on PPP Debugging" on page 506.
2. **Display the resulting `/var/log/pppdebug` log.**
   Check the output for the following message:
   ```
   LCP too many configure requests
   ```
   This message indicates that the speeds of serial lines configured for PPP might potentially be in conflict.
3. **Check if PPP is invoked through a program like `/bin/login` and the line speed was set.**
   In such a situation, `pppd` changes the originally configured line speed to the speed that is specified in `/bin/login`.
4. **Check if a user started PPP from the `mgetty` command and accidentally specified a bit rate.**
   This action also causes conflicting serial-line speeds.
5. **Fix the conflicting serial-line speed problem as follows:**
   a. **Lock the DTE rate on the modem.**
   b. **Do not use autobaud**
   c. **Do not change the line speed after it has been configured.**

Fixing Leased-Line Problems

The most common problem with leased lines is poor performance. In most situations, you need to work with the telephone company to fix the problem.

TABLE 34–6 Common Leased Line Problems

Symptom	Problem	Solution
The link does not start.	CSU bio-polar violations (CSU BPVs) can be the cause. One end of the link is set up for AMI lines and the other is set up or ESF bit 8 zero substitute (B8Zs).	If you are in the United States or Canada, you can directly fix this problem from the menu of the CSU/DSU. Check the CSU/DSU manufacturer's documentation for details. In other locales, the provider might be responsible for fixing CSU BPVs.
The link has poor performance.	The `pppd debug` output shows CRC errors when there is sustained traffic on the link. Your line might have a clocking problem, caused by misconfigurations between the telephone company and your network.	Contact the telephone company to ensure that it has used "loop clocking." On some unstructured leased lines, you might have to supply clocking. North American users should use loop clocking.

Diagnosing and Fixing Authentication Problems

TABLE 34–7 General Authentication Problems

Symptom	Problem	Solution
`pppd debug` output shows the message `Peer is not authorized to use remote address` *address*	You are using PAP authentication, and the IP address for the remote peer is not in the `/etc/ppp/pap-secrets` file.	Add an asterisk (*) after the entry for the peer in the `/etc/ppp/pap-secrets` file.
`pppd debug` output shows that LCP comes up but terminates shortly afterward	The password might be incorrect in the database for the particular security protocol	Check the password for the peer in the `/etc/ppp/pap-secrets` or `/etc/ppp/chap-secrets` file.

Diagnosing and Fixing PPPoE Problems

You can use PPP and standard UNIX utilities to identify problems with PPPoE. This section explains how to obtain debugging information for PPPoE and fix PPPoE-related problems.

How to Obtain Diagnostic Information for PPPoE

When you experience problems on the link and suspect PPPoE is the culprit, use the following diagnostic tools to obtain troubleshooting information.

1. **Become superuser on the machine that runs the PPPoE tunnel, either PPPoE client or PPPoE access server.**
2. **Turn on debugging, as explained in the procedure "How to Turn on PPP Debugging" on page 506.**
3. **View the contents of the log file** `/var/log/pppdebug`**.**

   The following example shows part of a log file generated for a link with a PPPoE tunnel

EXAMPLE 34–3 Log File for a Link With a PPPoE Tunnel

```
Sep  6 16:28:45 enyo pppd[100563]: [ID 702911 daemon.info] Plugin
  pppoe.so loaded.
Sep  6 16:28:45 enyo pppd[100563]: [ID 860527 daemon.notice] pppd
  2.4.0b1 (Sun Microsystems, Inc.,
Sep  5 2001 10:42:05) started by troot, uid 0
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] connect option:
   '/usr/lib/inet/pppoec
-v hme0' started (pid 100564)
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.info] Serial connection established.
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.info] Using interface sppp0
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.notice] Connect: sppp0
   <--> /dev/sppptun
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] /etc/ppp/pap-secrets
  is apparently empty
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] /etc/ppp/chap-secrets
  is apparently empty
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] sent
  [LCP ConfReq id=0xef <mru 1492>
asyncmap 0x0 <magic 0x77d3e953><pcomp><acomp>
Sep  6 16:28:46 enyo pppd[100563]: [ID 702911 daemon.debug] rcvd
  [LCP ConfReq id=0x2a <mru 1402>
asyncmap 0x0 <magic 0x9985f048><pcomp><acomp
```

If the debugging output does not help you isolate the problem, continue on with this procedure.

4. **Get diagnostic messages from PPPoE.**

   # /usr/lib/inet/pppoec -v *interface-name*

   pppoec sends diagnostic information to the stderr. If you run pppd in the foreground, the output appears on the screen. If pppd runs in the background, the output is sent to /etc/ppp/connect-errors.

   The next example shows the messages generated as the PPPoE tunnel is negotiated.

   EXAMPLE 34–4 PPPoE Diagnostic Messages

```
Connect option: '/usr/lib/inet/pppoec -v hme0' started (pid 100564)
/usr/lib/inet/pppoec: PPPoE Event Open (1) in state Dead (0): action SendPADI (2)
/usr/lib/inet/pppoec: Sending PADI to ff:ff:ff:ff:ff:ff: 18 bytes
/usr/lib/inet/pppoec: PPPoE State change Dead (0) -> InitSent (1)
/usr/lib/inet/pppoec: Received Active Discovery Offer from 8:0:20:cd:c1:2/hme0:pppoed
/usr/lib/inet/pppoec: PPPoE Event rPADO+ (5) in state InitSent (1): action SendPADR+ (5)
/usr/lib/inet/pppoec: Sending PADR to 8:0:20:cd:c1:2: 22 bytes
/usr/lib/inet/pppoec: PPPoE State change InitSent (1) -> ReqSent (3)
/usr/lib/inet/pppoec: Received Active Discovery Session-confirmation from
   8:0:20:cd:c1:2/hme0:pppoed
/usr/lib/inet/pppoec: PPPoE Event rPADS (7) in state ReqSent (3): action Open (7)
/usr/lib/inet/pppoec: Connection open; session 0002 on hme0:pppoe
/usr/lib/inet/pppoec: PPPoE State change ReqSent (3) -> Convers (4)
/usr/lib/inet/pppoec: connected
```

If the diagnostic messages do not help you isolate the problem, continue on with this procedure.

5. **Run** snoop **and save the trace to a file.**

   For information about snoop, refer to the snoop(1M) man page.

   # **snoop -o** *pppoe-trace-file*

6. **View the** snoop **trace file.**

   # **snoop -i** *pppoe-trace-file* **-v pppoe**

   The PPPoE trace that results displays on the screen.

   EXAMPLE 34–5 PPPoE snoop trace

```
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 1 arrived at 6:35:2.77
ETHER: Packet size = 32 bytes
ETHER: Destination = ff:ff:ff:ff:ff:ff, (broadcast)
ETHER: Source      = 8:0:20:78:f3:7c, Sun
ETHER: Ethertype = 8863 (PPPoE Discovery)
ETHER:
PPPoE: ----- PPP Over Ethernet -----
```

EXAMPLE 34–5 PPPoE snoop trace *(Continued)*

```
PPPoE:
PPPoE: Version = 1
PPPoE: Type = 1
PPPoE: Code = 9 (Active Discovery Initiation)
PPPoE: Session Id = 0
PPPoE: Length = 12 bytes
PPPoE:
PPPoE: ----- Service-Name -----
PPPoE: Tag Type = 257
PPPoE: Tag Length = 0 bytes
PPPoE:
PPPoE: ----- Host-Uniq -----
PPPoE: Tag Type = 259
PPPoE: Tag Length = 4 bytes
PPPoE: Data = Ox00000002
PPPoE:
.
.
.
ETHER: ----- Ether Header -----
ETHER:
ETHER: Packet 5 arrived at 6:35:2.87
ETHER: Packet size = 60 bytes
ETHER: Destination = 8:0:20:78:f3:7c, Sun)
ETHER: Source      = 0:2:fd:39:7f:7,
ETHER: Ethertype = 8864 (PPPoE Session)
ETHER:
PPPoE: ----- PPP Over Ethernet -----
PPPoE:
PPPoE: Version = 1
PPPoE: Type = 1
PPPoE: Code = 0 (PPPoE Session)
PPPoE: Session Id = 24383
PPPoE: Length = 20 bytes
PPPoE:
PPP: ----- Point-to-Point Protocol -----
PPP:
PPP-LCP: ----- Link Control Protocol -----
PPP-LCP:
PPP-LCP: Code = 1 (Configure Request)
PPP-LCP: Identifier = 80
PPP-LCP: Length = 18
```

CHAPTER 35

Solaris PPP 4.0 Reference

This chapter provides detailed conceptual information about Solaris PPP 4.0. Topics include:

Using PPP Options in Files and on the Command Line

Solaris PPP 4.0 contains a large set of options, which you use to define your PPP configuration. You use these options in the PPP configuration files, or on the command line, or by using a combination of files and command-line options. This section contains detailed information about the use of PPP options in configuration files and as arguments to PPP commands.

Where to Define PPP Options

Solaris PPP 4.0 is very flexible in the manner in which you can configure it. You can define PPP *options* in the following places:

- PPP configuration files
- PPP commands that are issued on the command line
- A combination of both places

The next table lists the PPP configuration files and commands.

TABLE 35–1 Summary of PPP Configuration Files and Commands

File or Command	Definition	Where it is Described
`/etc/ppp/options`	File that contains characteristics that apply by default to all PPP links on the system, for example, whether the machine requires peers to authenticate themselves. If this file is absent, non-root users are prohibited from using PPP.	"`/etc/ppp/options` Configuration File" on page 526
`/etc/ppp/options.`*ttyname*	File that describes the characteristics of all communications over the serial port *ttyname*.	"`/etc/ppp/options.`*ttyname* Configuration File" on page 527
`/etc/ppp/peers`	Directory that usually contains information about peers with which a dial-out machine connects. Files in this directory are used with the `call` option of the `pppd` command.	"Specifying Information About the Dial-in Server" on page 530
`/etc/ppp/peers/`*peer-name*	File that contains characteristics of the remote peer *peer-name*, such as its phone number and chat script for negotiating the link with the peer.	"`/etc/ppp/peers/`*peer-name* File" on page 531
`/etc/ppp/pap-secrets`	File that contains the necessary security credentials for Password Authentication Protocol (PAP) authentication.	"`/etc/ppp/pap-secrets` File" on page 544
`/etc/ppp/chap-secrets`	File that contains the necessary security credentials for Challenge-Handshake Authentication Protocol (CHAP) authentication.	"`/etc/ppp/chap-secrets` File" on page 548
`~/.ppprc`	File in the home directory of a PPP user, most often used with dial-in servers. This file contains specific information about each user's configuration.	"Configuring User-Specific Options" on page 529
`pppd` *options*	Command and options for initiating a PPP link and describing its characteristics.	"How PPP Options Are Processed" on page 523

Refer to the pppd(1M) man page for details on the PPP files and comprehensive descriptions of all options available to the pppd command. Sample templates for all the PPP configuration files are available in /etc/ppp.

How PPP Options Are Processed

All Solaris PPP 4.0 operations are handled by the pppd daemon, which starts when a user runs the pppd command. When a user calls a remote peer, the following occurs:

1. The pppd daemon parses:
   - /etc/ppp/options
   - $HOME/.ppprc
   - Any files that are opened by the file or call option in /etc/ppp/options and $HOME/.ppprc
2. pppd scans the command line to determine the device in use. The daemon does not yet interpret any options that are encountered.
3. pppd tries to discover the serial device to use by using the following criteria:
   a. If a serial device is specified on the command line, or a previously processed configuration file, pppd uses the name of that device.
   b. If no serial device is named, then pppd searches for the notty, pty, or socket option on the command line. If one of these options is specified, pppd assumes that no device name exists.
   c. Otherwise, if pppd discovers that standard input is attached to a tty, then the name of the tty is used.
   d. If pppd still cannot find a serial device, it terminates the connection and issues an error.
4. pppd then checks for the existence of the /etc/ppp/options.*ttyname* file. If the file is found, pppd parses the file.
5. pppd processes any options on the command line.
6. pppd negotiates the link control protocol (LCP) to set up the link.
7. [Optional] If authentication is required, pppd reads /etc/ppp/pap-secrets or /etc/ppp/chap-secrets to authenticate the opposite peer.

The file /etc/ppp/peers/*peer-name* is read when the pppd daemon encounters the option call *peer-name* on the command line or in the other configuration files.

How PPP Configuration File Privileges Work

Solaris PPP 4.0 configuration includes the concept of *privileges*. Privileges determine the precedence of configuration options, particularly when the same option is invoked in more than one place. An option that is invoked from a privileged source takes precedence over the same option that is invoked from a non-privileged source.

User Privileges

The only privileged user is superuser (`root`), with the UID of zero. All other users are not privileged.

File Privileges

The following are privileged configuration files, regardless of their ownership:

- `/etc/ppp/options`
- `/etc/ppp/options.`*ttyname*
- `/etc/ppp/peers/`*peer-name*

The file `$HOME/.ppprc` is owned by the user. Options read from `$HOME/.ppprc` and from the command line are privileged only if the user who is invoking `pppd` is root.

Arguments following the `file` option are privileged.

Effects of Option Privileges

Some options require the invoking user or source to be privileged in order to work. Options that are invoked on the command line are assigned the privileges of the user who is running the `pppd` command. These options are not privileged unless the user invoking `pppd` is `root`.

Option	Status	Explanation
`domain`	Privileged	Requires privileges for use.
`linkname`	Privileged	Requires privileges for use.
`noauth`	Privileged	Requires privileges for use.
`nopam`	Privileged	Requires privileges for use.
`pam`	Privileged	Requires privileges for use.
`plugin`	Privileged	Requires privileges for use.

Option	Status	Explanation
`privgroup`	Privileged	Requires privileges for use.
`allow-ip` *addresses*	Privileged	Requires privileges for use.
`name` *hostname*	Privileged	Requires privileges for use.
`plink`	Privileged	Requires privileges for use.
`noplink`	Privileged	Requires privileges for use.
`plumbed`	Privileged	Requires privileges for use.
`proxyarp`	Becomes privileged if `noproxyarp` has been specified	Cannot be overridden by an unprivileged use.
`defaultroute`	Privileged if `nodefaultroute` is set in a privileged file or by a privileged user	Cannot be overridden by an unprivileged user.
`disconnect`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an unprivileged user.
`bsdcomp`	Privileged if set in a privileged file or by a privileged user	The non-privileged user cannot specify a code size larger than the privileged user has specified.
`deflate`	Privileged if set in a privileged file or by a privileged user	The non-privileged user cannot specify a code size larger than the privileged user has specified.
`connect`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an unprivileged user.
`init`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an unprivileged user.
`pty`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an unprivileged user.
`welcome`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an unprivileged user.
ttyname	Privileged if set in a privileged file	Opened with root permissions regardless of who invokes `pppd`.
	Not privileged if set in a non-privileged file	Opened with the privileges of the user who invokes `pppd`.

/etc/ppp/options Configuration File

You use the /etc/ppp/options file to define global options for all PPP communications on the local machine. /etc/ppp/options is a privileged file. /etc/ppp/options should be owned by root, although pppd does not enforce this rule. Options that you define in /etc/ppp/options have precedence over definitions of the same options in all other files and the command line.

Typical options that you might use in /etc/ppp/options include:

- **lock**– Enables UUCP-style file locking
- **noauth** – Indicates that the machine does not authenticate callers

---

Note – The Solaris PPP 4.0 software does not include a default /etc/ppp/options file. pppd does not require the /etc/ppp/options file to work. But be aware that if a machine does not have an /etc/ppp/options file, only root can run pppd on that machine.

You must create /etc/ppp/options by using a text editor, as shown in "How to Define Communications Over the Serial Line " on page 461. If a machine does not require global options, you might want to create an empty /etc/ppp/options file. Then both root and regular users can run pppd on the local machine.

/etc/ppp/options.tmpl Template

The /etc/ppp/options.tmpl contains helpful comments about the /etc/ppp/options file, plus three common options for the global /etc/ppp/options file.

```
lock
nodefaultroute
noproxyarp
```

Option	Definition
lock	Enables UUCP-style file locking
nodefaultroute	Specifies that no default route is defined
noproxyarp	Disallows proxyarp.

To use /etc/ppp/options.tmpl as the global options file, rename /etc/ppp/options.tmpl to /etc/ppp/options. Then modify the file contents as needed by your site.

Where to Find Sample /etc/ppp/options Files

TABLE 35–2 Examples of the /etc/ppp/options File

Example /etc/ppp/options	Go to
For a dial-out machine	"How to Define Communications Over the Serial Line " on page 461
For a dial-in server	"How to Define Communications Over the Serial Line (Dial-in Server)" on page 469
For PAP support on a dial-in server	"How to Add PAP Support to the PPP Configuration Files (Dial-in Server)" on page 483
For PAP support on a dial-out machine	"How to Add PAP Support to the PPP Configuration Files (Dial-out Machine)" on page 486
For CHAP support on a dial-in server	"How to Add CHAP Support to the PPP Configuration Files (Dial-in Server)" on page 490

/etc/ppp/options.*ttyname* Configuration File

You can configure the characteristics of communications on the serial line in the /etc/ppp/options.*ttyname* file. /etc/ppp/options.*ttyname* is a privileged file. It is read by pppd after parsing the /etc/ppp/options and $HOME/.ppprc files, if they exist. Otherwise, pppd reads /etc/ppp/options.*ttyname* after parsing /etc/ppp/options.

ttyname is used for both dial-up and leased-line links. *ttyname* represents a particular serial port on a machine, such as cua/a or cua/b, where a modem or ISDN TA might be attached.

When naming the /etc/ppp/options.*ttyname* file, replace the slash (/) in the device name with a dot (.) . For example, the options file for device cua/b should be named /etc/ppp/options.cua.b.

Note – Solaris PPP 4.0 does not require an /etc/ppp/options.*ttyname* file to work correctly. If the server only has one serial line for PPP and requires few options, you can specify these options in another configuration file or on the command line.

Using /etc/ppp/options.*ttyname* on a Dial-in Server

For a dial-up link, you might choose to create individual /etc/ppp/options.*ttyname* files for every serial port on a dial-in server with a modem attached. Typical options include:

- IP address required by the dial-in server

  Set this option if you require incoming callers on serial port *ttyname* to use a particular IP address. Your address space might have a limited number of IP addresses available for PPP compared to the number of potential callers. If this is the situation, consider assigning an IP address to each serial interface that is used for PPP on the dial-in server. This assignment implements dynamic addressing for PPP.

- `asyncmap` *map_value*

  The `asyncmap` option maps control characters that cannot be received over the serial line by the particular modem or ISDN TA. When the `xonxoff` option is used, `pppd` automatically sets an asyncmap of 0xa0000.

  map_value states, in hexadecimal format, the control characters that are problematic.

- `init "chat -U -f /etc/ppp/mychat"`

  The `init` option tells the modem to initialize communications over the serial line by using the information in the `chat` –`U` command. The modem uses the chat string in the file `/etc/ppp/mychat`.

- Security parameters that are listed in the `pppd`(1m) man page.

Using `/etc/ppp/options.`*ttyname* on a Dial-out Machine

For a dial-out machine, you can create an `/etc/ppp/options.`*ttyname* file for the serial port with the modem, or elect not to use `/etc/ppp/options.`*ttyname*.

Note – Solaris PPP 4.0 does not require an `/etc/ppp/options.`*ttyname* file to work correctly. If the dial-out machine only has one serial line for PPP and requires few options, you can specify these options in another configuration file or on the command line.

`options.ttya.tmpl` Template File

The `/etc/ppp/options.ttya.tmpl` file contains helpful comments about the `/etc/ppp/options.`*tty-name* file. It contains three common options for the `/etc/ppp/options.`*tty-name* file.

```
38400
asyncmap 0xa0000
:192.168.1.1
```

Option	Definition
`38400`	Use this baud rate for port ttya.
`asyncmap 0xa0000`	Assign the asyncmap value of 0xa0000 so that the local machine can communicate with broken peers.
:192.168.1.1	Assign the IP address 192.168.1.1 to all peers calling in over the link

To use `/etc/ppp/options.ttya.tmpl` at your site, rename `/etc/ppp/options.tmpl` to `/etc/ppp/options.`*ttya-name*. Replace *ttya-name* with the name of the serial port with the modem. Then modify the file contents as needed by your site.

Where to Find Sample `/etc/ppp/options.`*ttyname* Files

TABLE 35–3 Examples of the `/etc/ppp/options.`*ttyname* File

Example `/etc/ppp/options.`*ttyname*	**For Instructions**
For a dial-out machine	"How to Define Communications Over the Serial Line " on page 461
For a dial-in server	"How to Define Communications Over the Serial Line (Dial-in Server)" on page 469

Configuring User-Specific Options

This section contains detailed information on setting up users on the dial-in server.

Configuring `$HOME/.pprc` on a Dial-in Server

The `$HOME/.ppprc` file is intended for users who are configuring preferred PPP options. As administrator, you can also configure `$HOME/.ppprc` for users.

The options in `$HOME/.ppprc` are privileged only when the user who is invoking the file is privileged.

When a caller uses the `pppd` command to initiate a call, the `.ppprc` file is the second file that is checked by the `pppd` daemon.

See "Setting Up Users of the Dial-in Server" on page 467 for instructions on setting up `$HOME/.pppprc` on the dial-in server.

Configuring `$HOME/.pppprc` on a Dial-out Machine

Note – The `$HOME/.pppprc` is not needed on the dial-out machine for Solaris PPP 4.0 to work correctly.

You do not need to have a `$HOME/.pppprc` on a dial-out machine, except for special circumstances. Create one or more `.pppprc` files if you do the following:

- Allow multiple users with differing communications needs to call remote peers from the same machine. In such an instance, create individual `.pppprc` files in the home directories of each user who must dial out.
- Need to specify options that control problems specific to your link, such as disabling Van Jacobson compression. See James Carlson's *PPP Design, Implementation, and Debugging* and the `pppd`(1M) man page for assistance in troubleshooting link problems.

Because the `.pppprc` file is most often used when configuring a dial-in server, refer to "How to Configure Users of the Dial-in Server" on page 467 for configuration instructions for `.pppprc`.

Specifying Information About the Dial-in Server

To communicate with a dial-in server, you need to gather information about the server and edit a few files. Most significantly, you must configure the communications requirements of all dial-in servers that the dial-out machine needs to call. You can specify options about a dial-in server, such as an ISP phone number, in the `/etc/ppp/options.`*ttyname* file. However, the optimum place to configure peer information is in `/etc/ppp/peers/`*peer-name* files.

/etc/ppp/peers/*peer-name* File

Note – The /etc/ppp/peers/*peer-name* file is not needed on the dial-out machine for Solaris PPP 4.0 to work correctly.

Use the /etc/ppp/peers/*peer-name* file to provide information for communicating with a particular peer. /etc/ppp/peers/*peer-name* allows ordinary users to invoke preselected privileged options that they are not allowed to set.

For example, a non-privileged user cannot override the noauth option if it is specified in the /etc/ppp/peers/*peer-name* file. Suppose the user wants to set up a link to peerB, which does not provide authentication credentials. As superuser, you can create a /etc/ppp/peers/peerB file that includes the noauth option. noauth indicates that the local machine does not authenticate calls from peerB.

The pppd daemon reads /etc/ppp/peers/*peer-name* when it encounters the following option:

call *peer-name*

You can create a /etc/ppp/peers/*peer-name* file for each target peer with which the dial-out machine needs to communicate. This is particularly convenient for permitting ordinary users to invoke special dial-out links without needing root privileges.

Typical options that you specify in /etc/ppp/peers/*peer-name* include the following:

- user *user_name*

  Supply *user_name* to the dial-in server, as the login name of the dial-out machine, when authenticating with PAP or CHAP.

- remotename *peer-name*

  Use *peer-name* as the name of the dial-in machine. remotename is used in conjunction with PAP or CHAP authentication, when scanning the /etc/ppp/pap-secrets or /etc/ppp/chap-secrets files.

- connect "chat *chat_script*..."

  Open communication to the dial-in server, using the instructions in the chat script.

- noauth

  Do not authenticate the peer *peer-name* when initiating communications.

- noipdefault

  Set the initial IP address that is used in negotiating with the peer to 0.0.0.0. Use noipdefault when setting up a link to most ISPs to help facilitate IPCP negotiation between the peers.

- defaultroute

Install a default IPv4 route when IP is established on the link.

See the pppd(1M)) man page for more options that might apply to a specific target peer.

/etc/ppp/peers/myisp.tmpl Template File

The /etc/ppp/peers/myisp.tmpl file contains helpful comments about the /etc/ppp/peers/*peer-name* file. The template concludes with common options such as you would use for an /etc/ppp/peers/*peer-name* file:

```
connect "/usr/bin/chat -f /etc/ppp/myisp-chat"
user myname
remotename myisp
noauth
noipdefault
defaultroute
updetach
noccp
```

Option	Definition
connect "/usr/bin/chat -f /etc/ppp/myisp-chat"	Call the peer using the chat script /etc/ppp/myisp-chat.
user myname	Use this account name for the local machine. It is the name for this machine in the peer's /etc/ppp/pap-secrets file.
remotename myisp	Recognize myisp as the name of the peer in the local machine's /etc/ppp/pap-secrets file.
noauth	Do not require calling peers to provide authentication credentials.
noipdefault	Do not use a default IP address for the local machine.
defaultroute	Use the default route assigned to the local machine
updetach	Log errors in the PPP log files, rather than on the standard output
noccp	Do not use CCP compression.

To use /etc/ppp/peers/myisp.tmpl at your site, rename /etc/ppp/peers/myisp.tmpl to /etc/ppp/peers/.*peer-name*. Replace *peer-name* with the name of the peer to be called. Then modify the file contents as needed by your site.

Where to Find Sample /etc/ppp/peers/*peer-name* Files

TABLE 35–4 Examples of /etc/ppp/peers/*peer-name* Files

Example /etc/ppp/peers/*peer-name*	**Go to**
For a dial-out machine	"How to Define the Connection With an Individual Peer" on page 463
For a local machine on a leased line	"How to Configure a Machine on a Leased-Line" on page 476
To support PAP authentication on a dial-out machine	"How to Add PAP Support to the PPP Configuration Files (Dial-out Machine)" on page 486
To support CHAP authentication on a dial-out machine	"How to Add CHAP Support to the PPP Configuration Files (Dial-out Machine)" on page 492
To support PPPoE on a client system	"Setting Up the PPPoE Client" on page 496

Configuring Modems for a Dial-up Link

This section contains information about configuring modems.

Configuring the Modem Speed

A major issue in modem configuration is designating the speed at which the modem should operate. The following guidelines apply to modems that are used with Sun Microsystems computers:

- Older SPARC systems – Check the hardware documentation that accompanies the system. Many SPARCstation™machines require modem speed not to exceed 38400 bps.
- UltraSPARC™ machines – Set the modem speed to 115200 bps, which is useful with modern modems and fast enough for a dial-up link. If you plan to use a dual-channel ISDN TA with compression, you need to increase the modem speed. The limit on an UltraSPARC is 460800 bps for an asynchronous link.

For a *dial-out machine*, set the modem speed in the PPP configuration files, such as /etc/ppp/peers/*peer-name*, or by specifying the speed as an option for pppd.

For a *dial-in server*, you need to set the speed by using the `ttymon` facility or `admintool`, as described in "Configuring Devices on the Dial-in Server" on page 466.

Defining the Conversation on the Dial-up Link

The dial-out machine and its remote peer communicate across the PPP link by negotiating and exchanging various instructions. When configuring a dial-out machine, you need to determine what instructions are required by the local and remote modems. Then you create a file called a chat script that contains these instructions . This section discusses information about configuring modems and creating chat scripts.

Contents of the Chat Script

Each remote peer that the dial-out machine needs to connect to probably requires its own chat script.

Note – Chat scripts are typically used only on dial-up links. Leased-line links do not use chat scripts unless an asynchronous interface is used that requires startup configuration.

The contents of the chat script are determined by the requirements of your modem model or ISDN TA, and the remote peer. These contents appear as a set of expect-send strings that the dial-out machine and its remote peers exchange as part of the communications initiation process.

An *expect* string contains characters that the dial-out host machine expects to receive from the remote peer to initiate conversation. A *send* string contains characters that the dial-out machine sends to the remote peer after receiving the expect string.

Information in the chat script usually includes the following:

- Modem commands (often referred to as *AT commands*) , which enable the modem to transmit data over the telephone
- Phone number of the target peer

  This phone number might be the number that is required by your ISP, or a dial-in server at a corporate site, or an individual machine.

- Time-out value, if required
- Login sequence expected from the remote peer
- Login sequence that is sent by the dial-out machine

Chat Script Examples

This section contains chat scripts that you can use as a reference for creating your own chat scripts. The modem manufacturer's guide and information from your ISP and other target hosts contain chat requirements for the modem and your target peers. In addition, numerous PPP web sites have sample chat scripts.

Basic Modem Chat Script

The following is a basic chat script that you can use as a template for creating your own chat scripts.

```
ABORT   BUSY
ABORT   'NO CARRIER'
REPORT  CONNECT
TIMEOUT 10
"" AT&F1M0&M5S2=255
SAY     "Calling myserver\n"
TIMEOUT 60
OK      "ATDT1-123-555-1212"
ogin: pppuser
ssword: \q\U
% pppd
```

The next table describes the contents of the chat script.

Script Contents	Explanation
`ABORT 'NO CARRIER'`	Abort transmission if the modem reports `ABORT 'NO CARRIER'` when dialing. The cause for this message is usually a dialing or modem negotiation failure.
`REPORT CONNECT`	Gather the `CONNECT` string from the modem and print it out.
`TIMEOUT 10`	Set initial timeout to 10 seconds. The modem's response should be immediate.
`"" AT&F1M0&M5S2=255`	`M0` – Turn off the speaker during connect. `&M5` – Make the modem require error control. S2=255 – Disable the TIES "+++" break sequence.
`SAY "Calling myserver\n"`	Display the message "Calling myserver" on the local machine.

Script Contents	Explanation
`TIMEOUT 60`	Reset the timeout to 60 seconds to allow more time for link negotiation.
`OK "ATDT1-123-555-1212"`	Call the remote peer, using the phone number 123-555-1212
`ogin: pppuser`	Log in to the peer by using UNIX-style login. Supply the user name `pppuser`.
`ssword: \q\U`	`\q` – Do not log if debugging with the `-v` option. `\U` – Insert the contents of the string that follows `-U`, which is specified on the command line (usually the password) here.
`% pppd`	Wait for the % shell prompt, and run the `pppd` command.

`/etc/ppp/myisp-chat.tmpl` Chat Script Template

Solaris PPP 4.0 includes the `/etc/ppp/myisp-chat.tmpl`, which you can modify for use at your site. `/etc/ppp/myisp-chat.tmpl` is similar to the basic modem chat script, except that it does not include a login sequence.

```
ABORT   BUSY
ABORT   'NO CARRIER'
REPORT  CONNECT
TIMEOUT 10
""      "AT&F1"
OK      "AT&C1&D2"
SAY     "Calling myisp\n"
TIMEOUT 60
OK      "ATDT1-123-555-1212"
CONNECT \c
```

Script Contents	Explanation
`ABORT BUSY`	Abort transmission if the modem receives this message from the opposite peer.
`ABORT 'NO CARRIER`	Abort transmission if the modem reports `ABORT 'NO CARRIER'` when dialing. The cause for this message is usually a dialing or modem negotiation failure.
`REPORT CONNECT`	Gather the `CONNECT` string from the modem and print it out.
`TIMEOUT 10`	Set initial timeout to 10 seconds. The modem's response should be immediate.
`"" "AT&F1"`	Reset the modem to factory defaults

Script Contents	Explanation
`OK "AT&C1&D2"`	Reset the modem so that, for &C1, DCD from the modem follows carrier. If the remote side hangs up the phone for some reason, then the DCD will drop. For &D2, DTR high-to-low transition causes the modem to go on-hook (hang up).
`SAY "Calling myisp\n"`	Display the message “Calling myisp” on the local machine.
`TIMEOUT 60`	Reset the timeout to 60 seconds to allow more time for link negotiation.
`OK "ATDT1-123-555-1212"`	Call the remote peer, using the phone number 123-555-1212
`CONNECT \c`	Wait for the `CONNECT` message from the opposite peer’s modem.

Modem Chat Script for Calling an ISP

Use the next chat script as a template for calling an ISP from a dial-out machine with a US Robotics Courier modem.

```
ABORT   BUSY
ABORT   'NO CARRIER'
REPORT  CONNECT
TIMEOUT 10
"" AT&F1M0&M5S2=255
SAY     "Calling myisp\n"
TIMEOUT 60
OK      "ATDT1-123-555-1212"
CONNECT \c
\r \d\c
SAY "Connected; running PPP\n"
```

The following table describes the contents of the chat script.

Script Contents	Explanation
`ABORT BUSY`	Abort transmission if the modem receives this message from the opposite peer.
`ABORT 'NO CARRIER'`	Abort transmission if the modem receives this message from the opposite peer.
`REPORT CONNECT`	Gather the `CONNECT` string from the modem and print it out.
`TIMEOUT 10`	Set initial timeout to 10 seconds. The modem’s response should be immediate.

Script Contents	Explanation
`"" AT&F1M0M0M0M0&M5S2=255`	`M0` – Turn off the speaker during connect. `&M5` – Make the modem require error control. S2=255 – Disable the TIES "+++" break sequence.
`SAY "Calling myisp\n"`	Display the message "Calling myisp" on the local machine.
`TIMEOUT 60`	Reset the timeout to 60 seconds to allow more time for link negotiation.
`OK "ATDT1-123-555-1212"`	Call the remote peer by using the phone number 123-555-1212.
`CONNECT \c`	Wait for the `CONNECT` message from the opposite peer's modem.
`\r \d\c`	Wait until the end of the CONNECT message.
SAY "Connected; running PPP\n"	Display the informative message "Connected; running PPP" on the local machine.

Basic Chat Script Enhanced for a UNIX-Style Login

The next chat script is a basic script that is enhanced for calling a remote Solaris peer or other UNIX type peer. This chat script is used in "How to Create the Instructions for Calling a Peer" on page 462.

```
SAY "Calling the peer\n"
TIMEOUT 10
ABORT BUSY
ABORT 'NO CARRIER'
ABORT ERROR
REPORT CONNECT
"" AT&F1&M5S2=255
TIMEOUT 60
OK ATDT1-123-555-1234
CONNECT \c
SAY "Connected; logging in.\n"
TIMEOUT 5
ogin:--ogin: pppuser
TIMEOUT 20
ABORT 'ogin incorrect'
ssword: \qmypassword
"% " \c
SAY "Logged in.  Starting PPP on peer system.\n"
ABORT 'not found'
"" "exec pppd"
~ \c
```

The following table explains the parameters of the chat script

Script Contents	Explanation
`TIMEOUT 10`	Set initial timeout to 10 seconds. The modem's response should be immediate.
`ABORT BUSY`	Abort transmission if the modem receives this message from the opposite peer.
`ABORT 'NO CARRIER'`	Abort transmission if the modem receives this message from the opposite peer.
`ABORT ERROR`	Abort transmission if the modem receives this message from the opposite peer.
`REPORT CONNECT`	Gather the `CONNECT` string from the modem and print it out.
`"" AT&F1&M5S2=255`	`&M5` – Make the modem require error control. S2=255 – Disable the TIES "+++" break sequence.
`TIMEOUT 60`	Reset the timeout to 60 seconds to allow more time for link negotiation.
`OK ATDT1-123-555-1234`	Call the remote peer, using the phone number 123-555-1212.
`CONNECT \c`	Wait for the `CONNECT` message from the opposite peer's modem.
`SAY "Connected; logging in.\n"`	Display the informative message "Connected; logging in," to give the user status.
`TIMEOUT 5`	Change the timeout to enable quick display of the login prompt.
`ogin:--ogin: pppuser`	Wait for the login prompt. If it is not received, send a RETURN and wait. Then send the user name `pppuser` to the peer. The sequence that follows is referred to by most ISPs as the PAP login, though it is not related in any way to PAP authentication
`TIMEOUT 20`	Change the timeout to 20 seconds to allow for slow password verification.
`ssword: \qmysecrethere`	Wait for the password prompt from the peer. When the prompt is received, send the password `\qmysecrethere`. The `\q` prevents the password from being written to the system log files.
`"% " \c`	Wait for a shell prompt from the peer. The chat script uses the C shell. Change this value if the user prefers to log in with a different shell.
`SAY "Logged in. Starting PPP on peer system.\n"`	Display the informative message "Logged in. Starting PPP on peer system" to give the user status.
`ABORT 'not found'`	Abort the transmission if the shell encounters errors.
`"" "exec pppd"`	Start `pppd` on the peer.
`~ \c`	Wait for PPP to start on the peer.

Starting PPP right after the CONNECT \c is often called a *PAP login* by ISPs, though the PAP login has nothing to do with PAP authentication.

The phrase ogin:--ogin: pppuser instructs the modem to send the user name, in this example pppuser, in response to the login prompt received from the dial-in server. pppuser is a special PPP user account name that was created for remote user1 on the dial-in server. (For instructions on creating PPP user accounts on a dial-in server, refer to "How to Configure Users of the Dial-in Server" on page 467.)

Chat Script for External ISDN TA

The following chat script is for calling from a dial-out machine with a ZyXEL omni.net. ISDN TA.

```
SAY "Calling the peer\n"
TIMEOUT 10
ABORT BUSY
ABORT 'NO CARRIER'
ABORT ERROR
REPORT CONNECT
"" AT&FB40S83.7=1&K44&J3X7S61.3=1S0=0S2=255
OK ATDI18882638234
CONNECT \c
\r \d\c
SAY "Connected; running PPP\n"
```

The following table explains the parameters of the chat script.

Script Contents	Explanation
SAY "Calling the peer"	Display this message on the screen of the dial-out machine.
TIMEOUT 10	Set the initial timeout to 10 seconds.
ABORT BUSY	Abort transmission if the modem receives this message from the opposite peer.
ABORT 'NO CARRIER'	Abort transmission if the modem receives this message from the opposite peer.
ABORT ERROR	Abort transmission if the modem receives this message from the opposite peer.
REPORT CONNECT	Gather the CONNECT string from the modem and print it out.

Script Contents	Explanation
"" AT&FB40S83.7= 1&K44&J3X7S61.3=1 S0=0S2=255	The letters in this line have the following meaning ■ &F –Use factory default ■ B40 – Do asynchronous PPP conversion ■ S83.7=1 – Use data over speech bearer ■ &K44 – Enable CCP compression ■ &J3 – Enable MP ■ X7 – Report DCE side rates ■ S61.3=1 – Use packet fragmentation ■ S0=0 – No auto answer ■ S2=255 – Disable TIES escape
OK ATDI18882638234	Make an ISDN call. For multi-link, the second call is placed to the same telephone number, which is normally what is required by most ISPs. If the remote peer requires a different second phone number, append "+*nnnn*" (*nnnn* represents the second phone number).
CONNECT \c	Wait for the CONNECT message from the opposite peer's modem.
\r \d\c	Wait until the end of the CONNECT message.
SAY "Connected; running PPP\n"	Display this message on the screen of the dial-out machine.

Refer to the`chat`(1M) man page for descriptions of options and other detailed information about the chat script. For an explanation of expect-send strings, refer to "UUCP Chat-Script Field" on page 591.

For More Chat Script Examples

A number of web sites offer sample chat scripts and assistance in creating them.

The PPP Frequently Asked Questions (FAQ) available from Australian National University posts *URL*.

Invoking the Chat Script

You call chat scripts by using the `connect` option. You can use `connect "chat ..."` in any PPP configuration file or on the command line.

Chat scripts are not executable, but the program that is invoked by `connect` must be executable. If you use the chat utility as that program and store your chat script in an external file by using the `-f` option, then your chat script file is not executable.

The `chat` program that is described in `chat`(1m) executes the actual chat script. The `pppd` daemon invokes the `chat` program whenever `pppd` encounters the `connect "chat ..."` option.

Note – You can use any external program, such as `Perl` or `Tcl`, to create advanced chat scripts. Solaris PPP 4.0 provides the `chat` utility as a convenience.

▼ How to Invoke a Chat Script (Task)

1. **Create the chat script as an ASCII file.**
2. **Invoke the chat script in any PPP configuration file by using the following syntax:**

   ```
   connect 'chat  -f /etc/ppp/chatfile'
   ```

   The `-f` flag indicates that a file name is to follow. */etc/ppp/chatfile* represents the name of the chat file.
3. **Give read permission for the external chat file to the user who will run the `pppd` command.**

Caution – The chat program always runs with the user's privileges, even if the `connect 'chat ...'` option is invoked from a privileged source. Thus, a separate chat file that is read with the `-f` option must be readable by the invoking user. This privilege can be a security problem if the chat script contains passwords or other sensitive information.

Chat Script in an External File

If the chat script that is needed for a particular peer is long or complicated, consider creating the script as a separate file. External chat files are easy to maintain and document. You can add comments to the chat file by preceding them with the hash (#) sign.

The procedure "How to Create the Instructions for Calling a Peer" on page 462 shows the use of a chat script that is contained in an external file.

Inline Chat Script

You can place the entire chat script conversation on a single line, such as the following::

```
connect 'chat "" "AT&F1" OK ATDT5551212 CONNECT "\c"'
```

The phrase that follows the `chat` keyword and terminates with `"\c"'` is the complete chat script. You use this form in any PPP configuration file or on the command line, as an argument to `pppd`.

Creating a Chat File That Is Executable

You can create a chat file that is an executable script to be run automatically when the dial-up link is initiated. Thus, you can run additional commands, such as `stty` for parity settings, besides those that are contained in a traditional chat script, during link initiation.

This executable chat script logs in to an old-style UNIX system that requires 7 bits/even parity and then changes to 8 bits/no parity when running PPP.

```
#!/bin/sh
chat "" "AT&F1" OK "ATDT555-1212" CONNECT "\c"
stty evenp
chat ogin: pppuser ssword: "\q\U" % "exec pppd"
stty -evenp
```

▼ How to Create an Executable Chat Program

1. **Use your text editor to create an executable chat program, such as the previous example.**
2. **Make the chat program executable by typing:**

   # **chmod +x /etc/ppp/***chatprogram*

3. **Invoke the chat program by typing the following on the command line:**

   `connect /etc/ppp/`*chatprogram*

   Chat programs do not have to be located within the `/etc/ppp` file system. You can store them in any location.

Authenticating Callers on a Link

This section explains how the PPP authentication protocols work and explains the databases that are associated with them.

Password Authentication Protocol (PAP)

PAP authentication is somewhat similar in operation to the UNIX `login` program, though it does not grant shell access to the user. PAP uses the PPP configuration files and PAP database in the form of the `/etc/ppp/pap-secrets` file for setting up authentication and defining PAP security credentials. These credentials include a peer name (a "user name" in PAP parlance), password, and related information for each caller who is permitted to link to the local machine. The PAP user names and passwords can be identical to or different from the UNIX user names and passwords in the password database. .

`/etc/ppp/pap-secrets` File

The PAP database is implemented in the `/etc/ppp/pap-secrets` file. Machines on both sides of the PPP link must have properly configured PAP credentials in their `/etc/ppp/pap-secrets` files for successful authentication. The caller (authenticatee) supplies credentials in the `user` and `password` columns of the `/etc/ppp/pap-secrets` file or in the obsolete `+ua` file. The server (authenticator) validates these credentials against information in `/etc/ppp/pap-secrets`, through the UNIX `passwd` database, or the PAM facility.

The `/etc/ppp/pap-secrets` file has the following syntax.

TABLE 35–5 Syntax of `/etc/ppp/pap-secrets`

Caller	Server	Password	IP Addresses
`myclient`	`ISP-server`	`mypassword`	*

The parameters have the following meaning:

- `myclient` is the PAP user name of the caller. Often this name is identical to the caller's UNIX user name, particularly where the dial-in server uses the `login` option of PAP.
- `ISP-server` is the name of the remote machine, often a dial-in server.
- `mypassword` is the caller's PAP password.
- `IP address` is the IP address that is associated with the caller. Use an asterisk (*) to indicate any IP address.

Creating PAP Passwords

PAP passwords are sent over the link *in the clear* (in readable ASCII format). For the caller (authenticatee), the PAP password must be stored in the clear in any of the following locations::

- In `/etc/ppp/pap-secrets`
- In another external file
- In a named pipe through the `pap-secrets` @ feature
- As an option to `pppd`, either on the command line or in a PPP configuration file
- Through the `+ua` file

On the server (authenticator), the PAP password can be hidden by doing one of the following:

- Specifying `papcrypt` and using passwords that are hashed by `crypt`(3C) in the `pap-secrets` file.
- Specifying the `login` option to `pppd` and omitting the password from the `pap-secrets` file by placing double quotes ("") in the password column. In this instance, authentication is done through the UNIX passwd database or the `pam`(3pam) mechanism.

What Happens During PAP Authentication

PAP authentication occurs in the following sequence.

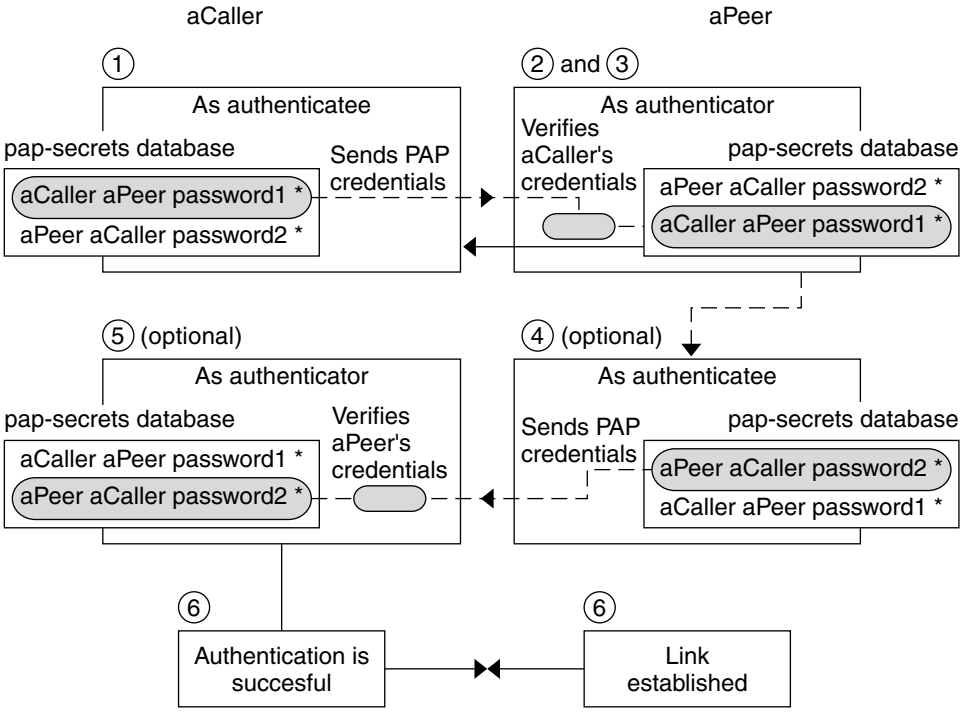


FIGURE 35–1 PAP Authentication Process

1. The caller (authenticatee) calls the remote peer (authenticator) and provides its PAP user name and password as part of link negotiation.
2. The peer verifies the identity of the caller in its`/etc/ppp/pap-secrets` file. If the peer uses the `login` option of PAP, it verifies the caller's user name and password in its password database.
3. If authentication is successful, the peer continues link negotiation with the caller. If authentication fails, the link is dropped.
4. [Optional]If the caller authenticates responses from remote peers, the remote peer must send its own PAP credentials to the caller. Thus, the remote peer becomes the authenticatee and the caller the authenticator.
5. The original caller reads its own `/etc/ppp/pap-secrets` to verify the identity of the remote peer .

Note – In situations where the original caller does require authentication credentials from the remote peer, Steps 1 and 4 happen in parallel.

If the peer is authenticated, negotiation continues. Otherwise, the link is dropped.

6. Negotiation between caller and peer continues until the link is successfully established.

Using the `login` Option With `/etc/ppp/pap-secrets`

You can add the `login` option for authenticating PAP credentials to any PPP configuration file. When `login` is specified, for example, in `/etc/ppp/options`, `pppd` verifies that the caller's PAP credentials exist in the Solaris password database. The following table shows the format of a `/etc/ppp/pap-secrets` file with the `login` option:

TABLE 35–6 `/etc/ppp/pap-secrets` With `login` Option

Caller	Server	Password	IP Addresses
joe	*	" "	*
sally	*	" "	*
sue	*	" "	*

The parameters have the following meaning:

- **Caller** column contains names of all authorized callers.
- **Server** column contains an asterisk, which indicates that any server name is valid. (The `name` option is not required in the PPP configuration files.)
- **Password** column contains double quotes, which indicate that any password is valid.

  If you type a password in this column, then the password that is supplied by the peer must match both the PAP password and the UNIX `passwd` database.
- **IP Addresses** contains an asterisk, which indicates that any IP address is allowed.

Challenge-Handshake Authentication Protocol (CHAP)

CHAP authentication uses the notion of the *challenge* and *response*, wherein the peer (authenticator) challenges the caller (authenticatee) to prove its identity. The challenge

includes a random number and a unique ID that is generated by the authenticator. The caller must use the ID, random number, and its CHAP security credentials to generate the proper response (handshake) to send to the peer.

CHAP security credentials include a CHAP user name and a CHAP *secret*, an arbitrary string that is known to both caller and peer before they negotiate a PPP link. You configure CHAP security credentials in the CHAP database, `/etc/ppp/chap-secrets`.

`/etc/ppp/chap-secrets` File

The CHAP database is implemented in the `/etc/ppp/chap-secrets` file. Machines on both sides of the PPP link must have each others' CHAP credentials in their `/etc/ppp/chap-secrets` files for successful authentication.

Note – Unlike PAP, the shared secret must be in the clear on both peers. You cannot use crypt, PAM, or the PPP login option with CHAP.

The `/etc/ppp/chap-secrets` file has the following syntax.

TABLE 35–7 Syntax of `/etc/ppp/chap-secrets`

Caller	Server	CHAP secret	IP Addresses
`myclient`	`myserver`	`secret5748`	*

The parameters have the following meanings:

- `myclient` is the CHAP user name of the caller. This name can be the same or different from the caller's UNIX user name.
- `myserver` is the name of the remote machine, often a dial-in server.
- `secret5748` is the caller's CHAP secret.

  Note – Unlike PAP passwords, CHAP secrets are never sent over the link. Rather, they are used when the local machines compute the response.

- `IP address` is the IP address that is associated with the caller. Use an asterisk (*) to indicate any IP address.

What Happens During CHAP Authentication

CHAP authentication occurs in the following sequence.

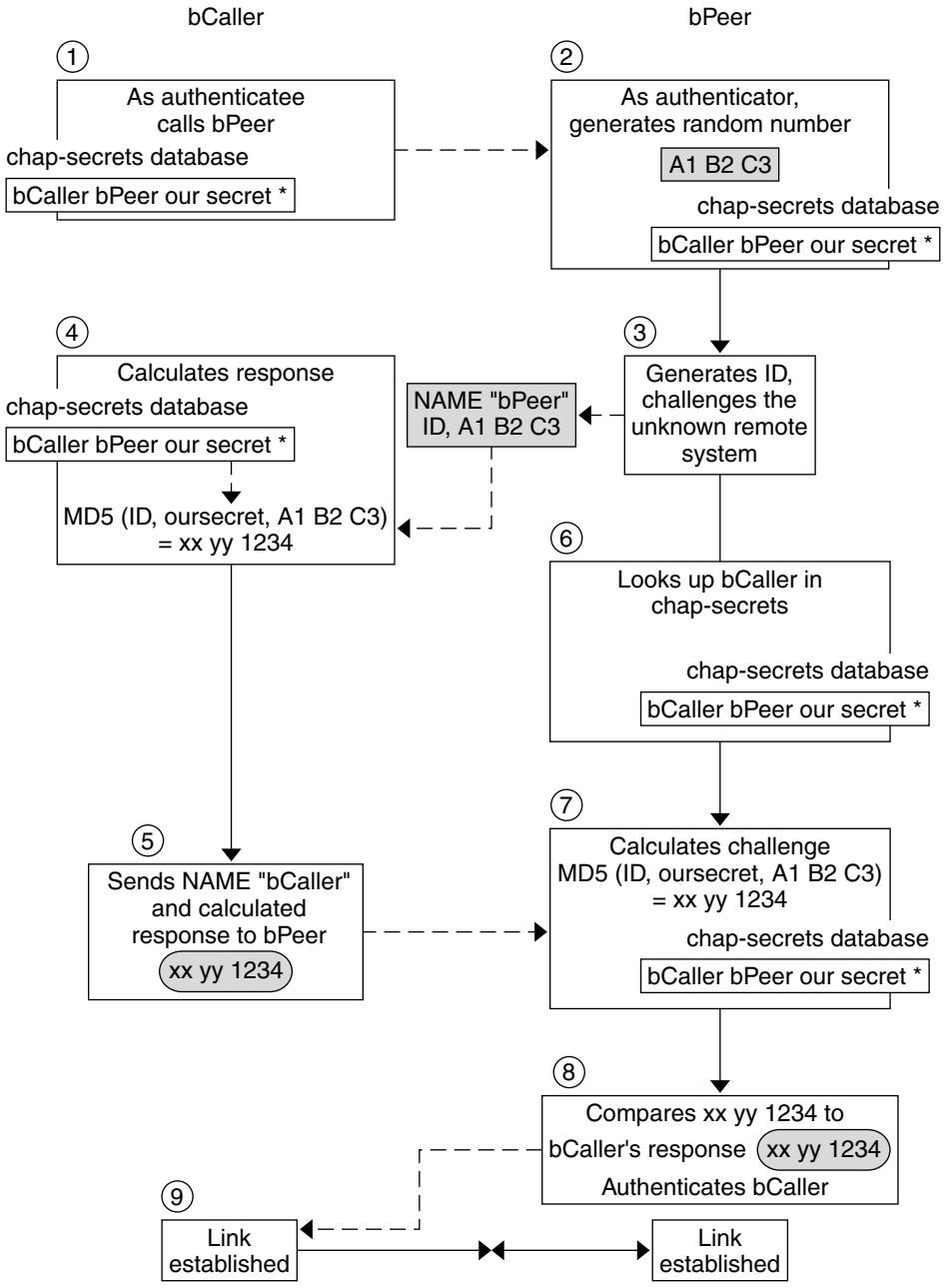


FIGURE 35–2 CHAP Authentication Sequence

1. Two peers that are about to initiate communications agree on a secret to be used for authentication during negotiation of a PPP link.
2. The administrators of both machines add the secret, CHAP user names, and other CHAP credentials to the `/etc/ppp/chap-secrets` database of their respective machines.
3. The caller (authenticatee) calls the remote peer (authenticator).
4. The authenticator generates a random number and an ID, and sends them to the authenticatee as a challenge.
5. The authenticatee looks up the peer's name and secret in its `/etc/ppp/chap-secrets` database.
6. The authenticatee calculates a response by applying the MD5 computational algorithm to the secret and the peer's random number challenge. Then the authenticatee sends the results as its response to the authenticator.
7. The authenticator looks up the authenticatee's name and secret in its `/etc/ppp/chap-secrets` database.
8. The authenticator calculates its own figure by applying MD5 to the number that was generated as the challenge and the secret for the authenticatee in `/etc/ppp/chap-secrets`.
9. The authenticator compares its results with the response from the caller. If the two numbers are the same, the peer has successfully authenticated the caller, and link negotiation continues. Otherwise the link is dropped.

Creating an IP Addressing Scheme for Callers

Consider creating one or more IP addresses for all incoming calls instead of assigning a unique IP address to each remote user. Dedicated IP addresses are particularly important if the number of potential callers exceeds the number of serial ports and modems on the dial-in server. You can implement a number of different scenarios, depending on your site's needs. Moreover, the scenarios are not mutually exclusive.

Assigning Dynamic IP Addresses to Callers

Dynamic addressing involves the assignment to each caller the IP address that is defined in `/etc/ppp/options.`*ttyname*. Dynamic addressing occurs on a per-serial port basis. Each time a call comes in over a particular serial line, the caller is given the

IP address that is defined in the `/etc/ppp/options.`*ttyname* file for the serial interface that is handling the call.

For example, suppose a dial-in server has four serial interfaces that provide dial-up service to incoming calls:

- For serial port `term/a`, create the file `/etc/ppp/options.term.a` with the entry:

  ```
  :10.1.1.1
  ```

- For serial port `term/b`, create the file `/etc/ppp/options.term.b` with the entry:

  ```
  :10.1.1.2
  ```

- For serial port `term/c`, create the file `/etc/ppp/options.term.c` with the entry:

  ```
  :10.1.1.3
  ```

- For serial port `term/d`, create the file `/etc/ppp/options.term.d` with the entry:

  ```
  :10.1.1.4
  ```

With this addressing scheme, an incoming call on serial interface `/dev/term/c` is given the IP address 10.1.1.3 for the duration of the call. After the first caller hangs up, a later call that comes in over serial interface `/dev/term/c` is also given the IP address 10.1.1.3.

The advantages of dynamic addressing include:

- You can track PPP network usage down to the serial port.
- You can assign a minimum number of IP addresses for PPP use.
- You can administer IP filtering in a more simplified fashion.

Assigning Static IP Addresses to Callers

If your site implements PPP authentication, you can assign specific, *static* IP addresses to individual callers. In this scenario, every time a dial-out machine calls the dial-in server, the caller receives the same IP address.

You implement static addresses in either the pap-secrets or chap-secrets database. Here is a sample `/etc/ppp/pap-secrets` file with static IP addresses defined.

Caller	Server	Password	IP Addresses
`joe`	`myserver`	`joepasswd`	10.10.111.240
`sally`	`myserver`	`sallypasswd`	10.10.111.241

Caller	Server	Password	IP Addresses
sue	myserver	suepasswd	10.10.111.242

Here is a sample /etc/ppp/chap-secrets file that defines static IP addresses.

Caller	Server	CHAP secret	IP Addresses
account1	myserver	secret5748	10.10.111.244
account2	myserver	secret91011	10.10.111.245

Assigning IP Addresses by sppp Unit Number

If you are using either PAP or CHAP authentication, you can assign IP addresses to callers by the sppp unit number. The next table shows an example of this usage.

Caller	Server	Password	IP Addresses
myclient	ISP-server	mypassword	10.10.111.240/28+

The plus (+) indicates that the unit number is added to the IP address. Addresses 10.10.111.240 through 10.10.111.255 are assigned to remote users. sppp0 gets IP address 10.10.111.240. sppp1 gets IP address 10.10.111.241, and so on.

Creating PPPoE Tunnels for DSL Support

By using PPPoE, you can provide PPP over high-speed digital services to multiple clients that are using one or more DSL modems. PPPoE implements these services by creating an Ethernet tunnel through three participants: the enterprise, the telephone company, and the service provider.

- For an overview and description of how PPPoE works, see "PPPoE Overview" on page 435.
- For tasks for setting up PPPoE tunnels, see Chapter 33.

This section contains detailed information about PPPoE commands and files, which are summarized in the next table.

TABLE 35–8 PPPoE Commands and Configuration Files

File or Command	Description	Where it Is Described
`/etc/ppp/pppoe`	File that contains characteristics that are applied by default to all tunnels that were set up by PPPoE on the system	"`/etc/ppp/pppoe` File" on page 556
`/etc/ppp/pppoe.`*device*	File that contains characteristics of a particular interface that is used by PPPoE for a tunnel	"`/etc/ppp/pppoe.`*device* File" on page 557
`/etc/ppp/pppoe.if`	File that lists the Ethernet interface over which the tunnel set up by PPPoE runs	"`/etc/ppp/pppoe.if` File" on page 553
`/usr/sbin/sppptun`	Command for configuring the Ethernet interfaces that are involved in a PPPoE tunnel	"`/usr/sbin/sppptun` Command" on page 554
`/usr/lib/inet/pppoed`	Command and options for using PPPoE to set up a tunnel	"`/usr/lib/inet/pppoed` Daemon" on page 555

Files for Configuring Interfaces for PPPoE

The interfaces that are used at either end of the PPPoE tunnel must be configured before the tunnel can support PPP communications. Use `/usr/sbin/sppptun` and `/etc/ppp/pppoe.if` files for this purpose. You must use these tools to configure Ethernet interfaces on all Solaris PPPoE clients and access servers.

`/etc/ppp/pppoe.if` File

The `/etc/ppp/pppoe.if` file lists the names of all Ethernet interfaces on a host to be used for the PPPoE tunnels. This file is processed during system boot up, when the interfaces that are listed are plumbed for use in PPPoE tunnels.

You need to explicitly create `/etc/ppp/pppoe.if`. Type the name of one interface to be configured for PPPoE on each line.

Sample `/etc/ppp/pppoe.if` *File*

The following example shows an `/etc/ppp/pppoe.if` file for a server that offers three interfaces for PPPoE tunnels

```
# cat /etc/ppp/pppoe.if
hme1
hme2
hme3
```

PPPoE clients usually have only one interface that is listed in `/etc/ppp/pppoe.if`.

`/usr/sbin/sppptun` Command

You can use the `/usr/sbin/sppptun` command to manually plumb and unplumb the Ethernet interfaces to be used for PPPoE tunnels. (By contrast, `/etc/ppp/pppoe.if` is only read when the system boots up.) These interfaces should correspond to the interfaces that are listed in `/etc/ppp/pppoe.if`.

`sppptun` plumbs the Ethernet interfaces that are used in PPPoE tunnels in a manner similar to the `ifconfig` command. Unlike `ifconfig`, you must plumb interfaces twice to support PPPoE because two Ethernet protocol numbers are involved.

The basic syntax for `sppptun` is as follows:

 # /usr/sbin/sppptun plumb pppoed *device-name*
device-name`:pppoed`
 # /usr/sbin/sppptun plumb pppoe `device-name`
device-name`:pppoe`

In this syntax, *device-name* is the name of the device to be plumbed for PPPoE

The first time you issue the `sppptun` command, the discovery protocol `pppoed` is plumbed on the interface. The second time you run `sppptun`, the session protocol `pppoe` is plumbed. `sppptun` prints the name of the interface just plumbed. You use this name to unplumb the interface, when necessary.

For more information, refer to the`sppptun`(1M) man page.

Sample `sppptun` *Commands for Administering Interfaces*

- The following sample shows how to manually plumb an interface for PPPoE by using `/usr/sbin/sppptun`.

  EXAMPLE 35–1 To plumb an Interface to Support PPPoE

  ```
  # /usr/sbin/sppptun plumb pppoed hme0
  hme0:pppoed
   # /dev/sppptun plumb pppoe hme0
   hme0:pppoe
  ```

- This sample shows how to list the interfaces on an access server plumbed for PPPoE.

  EXAMPLE 35–2 To List All Interfaces on a PPPoE Access Server

  ```
  /usr/sbin/sppptun query
  hme0:pppoe
  hme0:pppoed
  ```

EXAMPLE 35–2 To List All Interfaces on a PPPoE Access Server *(Continued)*

```
hme1:pppoe
hme1:pppoed
hme2:pppoe
hme2:pppoed
```

- This sample shows how to unplumb an interface.

EXAMPLE 35–3 To Unplumb an Interface With a PPPoE Tunnel

```
# sppptun unplumb hme0:pppoed
# sppptun unplumb hme0:pppoe
```

PPPoE Access Server Commands and Files

A service provider that offers DSL services or support to customers can use an access server that is running Solaris PPPoE. The PPPoE access server and client do function in the traditional client-server relationship.This relationship is similar to that of the dial-out machine and dial-in server on a dial-up link, in that one system initiates communications and one system answers. By contrast, the PPP protocol has no notion of the client-server relationship and considers both machines equal peers.

The commands and files that set up a PPPoE access server include:

- "`/usr/sbin/sppptun` Command" on page 554
- "`/usr/lib/inet/pppoed` Daemon" on page 555
- "`/etc/ppp/pppoe` File" on page 556
- "`/etc/ppp/pppoe.`*device* File" on page 557
- "`pppoe.so` Plugin" on page 560

`/usr/lib/inet/pppoed` Daemon

The `pppoed` daemon accepts broadcasts for services from prospective PPPoE clients. Additionally, `pppoed` negotiates the server side of the PPPoE tunnel and runs `pppd`, the PPP daemon, over that tunnel.

You configure `pppoed` services in the `/etc/ppp/pppoe` and `/etc/ppp/pppoe.`*device* files. If `/etc/ppp/pppoe` exists when the system boots up, `pppoed` runs automatically. You can also explicitly run the `pppoed` daemon on the command line by typing `/usr/lib/inet/pppoed`.

/etc/ppp/pppoe File

The /etc/ppp/pppoe describes the services that are offered by an access server, plus options that define how PPP runs over the PPPoE tunnel. You can define services for individual interfaces, or globally, that is, for all interfaces on the access server. The access server sends the information in the /etc/ppp/pppoe file in response to a broadcast from a potential PPPoE client.

The following is the basic syntax of /etc/ppp/pppoe:

global-options
service *service-name*
service-specific-options
service *another-service-name*
service-specific-options
device *interface-name*

The parameters have the following meanings:

- *global options* – Sets the default options for the /etc/ppp/pppoe file. These options can be any options available through pppoed or pppd. For complete lists of options, see the man pages pppoed(1M) and pppd(1M).

  For example, you must list the Ethernet interfaces available for the PPPoE tunnel as part of *global options*. If you do not define devices in /etc/ppp/pppoe, the services are not offered on any interface.

  To define devices as a global option, use the following form:

  device *interface <,interface>*
  interface specifies the interface where the service listens for potential PPPoE clients. If more than one interface is associated with the service, separate each name with a comma.
- **service** *service-name* – Starts the definition of the service *service-name*. *service-name* is a string that can be any phrase appropriate to the services that are provided.
- *service-specific-options* – Lists the PPPoE and PPP options specific to this service.
- device *interface-name* – Specifies the interface where the previously listed service is available.

For additional options to /etc/ppp/pppoe, refer to the pppoed(1M) and pppd(1M) man pages.

A typical /etc/ppp/pppoe file might look like the following.

EXAMPLE 35–4 Basic /etc/ppp/pppoe File

```
device hme1,hme2,hme3
service internet
   pppd "name internet-server"
service intranet
   pppd "192.168.1.1:"
service debug
   device hme1
```

EXAMPLE 35–4 `Basic /etc/ppp/pppoe File` *(Continued)*

```
pppd "debug name internet-server"
```

In this file, the following apply:

- `hme1,hme2,hme3` are three interfaces on the access server to be used for PPPoE tunnels.
- `service internet` advertises a service that is called `internet` to prospective clients. The provider that offers the service also determines how `internet` is defined. For example, a provider might `internet` to mean various IP services, as well as access to the Internet.
- `pppd` sets the command-line options that are used when the caller invokes `pppd`. The option `"name internet-server"` gives the name of the local machine (the access server) as `internet-server`.
- `service intranet` advertises another service, called `intranet`, to prospective clients.
- `pppd "192.168.1.1:"` sets the command-line options that are used when the caller invokes `pppd`. When the caller invokes `pppd`, 192.168.1.1 is set as the IP address for the local machine (the access server).
- `service debug` advertises a third service, debugging , on the interfaces that are defined for PPPoE.
- `device hme1` restricts debugging to PPPoE tunnels to `hme1`.
- `pppd "debug name internet-server"` sets the command line options that are used when the caller invokes `pppd`, in this instance PPP debugging on `internet-server`, the local machine.

`/etc/ppp/pppoe.`*device* File

The `/etc/ppp/pppoe.`*device* file describes the services that are offered on one interface of a PPPoE access server, plus options that define how PPP runs over the PPPoE tunnel. `/etc/ppp/pppoe.`*device* is an optional file, which operates exactly like the global `/etc/ppp/pppoe`. However, if `/etc/ppp/pppoe.`*device* is defined for an interface, its parameters take precedence for that interface over the global parameters that are defined in `/etc/ppp/pppoe`.

The basic syntax of `/etc/ppp/pppoe.`*device* is:

`service` *service-name*
    service-specific-options
`service` *another-service-name*
    service-specific-options

The only difference between this syntax and that of /etc/ppp/pppoe is that you cannot use the device option that is shown in "/etc/ppp/pppoe File" on page 556.

pppoe.so Plugin

pppoe.so is the PPPoE shared object file that must be invoked by PPPoE access servers and clients. This file limits MTU and MRU to 1492, filters packets from the driver, and negotiates the PPPoE tunnel, along with pppoed. On the access server side, pppoe.so is automatically invoked by the pppd daemon.

Using PPPoE and PPP Files to Configure an Access Server

This section contains samples of all files that are used to configure an access server. The access server is multihomed and attached to three subnets: green, orange, and purple. pppoed runs as root on the server, which is the default.

PPPoE clients can access the orange and purple networks through interfaces hme0 and hme1. Clients log in to the server by using the standard UNIX login. The server authenticates them by using PAP.

The green network is not advertised to clients. The only way clients can access green is by directly specifying "green-net" and supplying CHAP authentication credentials. Moreover, only clients joe and mary are allowed to access the green network. They must use static IP addresses to do so.

EXAMPLE 35–5 /etc/ppp/pppoe File for an Access Server

```
service orange-net
     device hme0,hme1
     pppd "require-pap login name orange-server orange-server:"
service purple-net
     device hme0,hme1
     pppd "require-pap login name purple-server purple-server:"
service green-net
     device hme1
     pppd "require-chap name green-server green-server:"
     nowildcard
```

This sample describes the services available from the access server:The first service section describes the services of the orange network.

```
service orange-net
     device hme0,hme1
     pppd "require-pap login name orange-server orange-server:"
```

Clients access the orange network over interfaces hme0 and hme1. The options that are given to the pppd command make the server require PAP credentials from

potential clients. The pppd options also set the server's name to orange-server, as used in the pap-secrets file.

The service section for the purple network is identical to that of the orange network, except for the network and server names.

The next section describes the services of the green network:

```
service green-net
     device hme1
     pppd "require-chap name green-server green-server:"
     nowildcard
```

This section restricts client access to interface hme1. Options that are given to the pppd command make the server require CHAP credentials from prospective clients. The pppd options also set the server name to green-server, to be used in the chap-secrets file. The nowildcard option specifies that the existence of the green network is not advertised to clients.

For the access server scenario just discussed, you might set up the following /etc/ppp/options file.

EXAMPLE 35–6 /etc/ppp/options File for an Access Server

```
auth
proxyarp
nodefaultroute
name no-service    # don't authenticate otherwise
```

The option name no-service overrides the server name that is normally searched for during PAP or CHAP authentication. The server's default name is the one that found in the /usr/bin/hostname file. The name option in the previous example changes the server's name to no-service, a name not likely to be found in a pap or chap-secrets file. This action prevents a random user from running pppd and overriding the auth and name options that are set in /etc/ppp/options. pppd then fails because it cannot find any secrets for the client with a server name of no-service.

The access server scenario uses the following /etc/hosts file:

EXAMPLE 35–7 /etc/hosts File for an Access Server

```
172.16.0.1    orange-server
172.17.0.1    purple-server
172.18.0.1    green-server
172.18.0.2    joes-pc
172.18.0.3    marys-pc
```

Here is the /etc/ppp/pap-secrets file that is used for PAP authentication for clients that attempt to access the orange and purple networks.

EXAMPLE 35–8 `/etc/ppp/pap-secrets` File for an Access Server

```
* orange-server "" 172.16.0.2/16+
* purple-server "" 172.17.0.2/16+
```

Here is the `/etc/ppp/chap-secrets` file used for CHAP authentication. Note that only clients joe and mary are listed in the file.

EXAMPLE 35–9 `/etc/ppp/chap-secrets` File for an Access Server

```
joe green-server "joe's secret" joes-pc
mary green-server "mary's secret" marys-pc
```

PPPoE Client Commands and Files

To run PPP over a DSL modem, a machine must become a PPPoE client. You have to plumb an interface to run PPPoE, and then use the `pppoec` utility to "discover" the existence of an access server. Thereafter, the client can create the PPPoE tunnel over the DSL modem and run PPP.

The PPPoE client relates to the access server in the traditional client-server model. The PPPoE tunnel is not a dial-up link, but is configured and operated in much the same manner.

The commands and files that set up a PPPoE client include the following::

- "`/usr/sbin/sppptun` Command" on page 554
- "`/usr/lib/inet/pppoec` Utility" on page 560
- "`pppoe.so` Plugin" on page 560
- "`/etc/ppp/peers/`*peer-name* File" on page 531
- "`/etc/ppp/options` Configuration File" on page 526

`/usr/lib/inet/pppoec` Utility

The `/usr/lib/inet/pppoec` utility is responsible for negotiating the client side of a PPPoE tunnel. `pppoec` is similar to the Solaris PPP 4.0 `chat` utility, in that you do not invoke it directly. Rather, you start `/usr/lib/inet/pppoec` as an argument to the `connect` option of `pppd`.

`pppoe.so` Plugin

`pppoe.so` is the PPPoE shared object that must be loaded by PPPoE to provide PPPoE capability to access servers and clients. This shared object limits MTU and MRU to 1492, filters packets from the driver, and handles runtime PPPoE messages.

On the client side, pppd loads pppoe.so when the user specifies the plugin pppoe.so option.

/etc/ppp/peers/*peer-name* File for Defining an Access Server Peer

When you define an access server to be discovered by pppoec, you use options that apply to both pppoec and the pppd daemon. A /etc/ppp/peers/*peer-name* file for an access server requires the following parameters:

- sppptun – Name for the serial device that is used by the PPPoE tunnel
- plugin pppoe.so – Instructs pppd to load the pppoe.so shared object
- connect "/usr/lib/inet/pppoec *device*" – Starts a connection and invokes the pppoec utility over *device*, the interface that is plumbed for PPPoE

The remaining parameters in the /etc/ppp/peers/*peer-name* file should apply to the PPP link on the server. Use the same options that you would for /etc/ppp/peers/*peer-name* on a dial-out machine. Try to limit the number of options to the minimum you need for the PPP link.

The following example is introduced in "How to Define a PPPoE Access Server Peer" on page 497.

EXAMPLE 35–10 /etc/ppp/peers/*peer-name* to Define a Remote Access Server

```
# vi /etc/ppp/peers/dslserve
sppptun
plugin pppoe.so
connect "/usr/lib/inet/pppoec hme0"
noccp
noauth
user Red
password redsecret
noipdefault
defaultroute
```

This file defines parameters to be used when setting up a PPPoE tunnel and PPP link to access server dslserve. The options included are as follows:

Option	Description
sppptun	Defines sppptun as the name of the serial device.
plugin pppoe.so	Instructs pppd to load the pppoe.so shared object.
connect "/usr/lib/inet/pppoec hme0"	Runs pppoec and designates hme0 as the interface for the PPPoE tunnel and PPP link.

Option	Description
`noccp`	Turns off CCP compression on the link. **Note –** If they use any compression algorithms at all, many ISPs use only proprietary compression algorithms. Turning off the publicly available CCP algorithm saves negotiation time and avoids very occasional interoperability problems.
`noauth`	Stops `pppd` from demanding authentication credentials from the access server. Most ISPs do not provide authentication credentials to customers.
`user Red`	Sets the name `Red` as the user name for the client, required for PAP authentication by the access server.
`password redsecret`	Defines `redsecret` as the password to be provided to the access server for PAP authentication.
`noipdefault`	Assigns 0.0.0.0 as the initial IP address.
`defaultroute`	Tells `pppd` to install a default IPv4 route after IPCP negotiation. You should include `defaultroute` in `/etc/ppp/peers/`*peer-name* when the link is the system's link to the Internet, which true for a PPPoE client.

CHAPTER 36

Migrating From Asynchronous Solaris PPP to Solaris PPP 4.0 (Tasks)

Earlier versions of the Solaris operating system included different PPP implementation, Asynchronous Solaris PPP (`asppp`). If you want to convert peers that run `asppp` to the newer PPP 4.0, you need to run a conversion script. This chapter covers the following topics in PPP conversion:

The appendix uses a sample `asppp` configuration to explain how to accomplish PPP conversion. For a description of the differences between Solaris PPP 4.0 and `asppp`, go to "Which Version of Solaris PPP to Use" on page 424.

Before Converting `asppp` Files

You can use the conversion script `/usr/sbin/asppp2pppd` to convert the files that make up a standard `asppp` configuration:

- `/etc/asppp.cf` – Asynchronous PPP configuration file
- `/etc/uucp/Systems` – UUCP file that describes the characteristics of the remote peer
- `/etc/uucp/Devices` – UUCP file that describes the modem on the local machine
- `/etc/uucp/Dialers` – UUCP file that contains the login sequence to be used by the modem that is described in the `/etc/uucp/Devices` file

For more information about each of these files, see the *Solaris 8 System Administration Collection, Volume 3*, available from `http://www.docs.sun.com`.

Example—/etc/asppp.cf Configuration File

The procedure to be shown in"How to Convert From asppp to Solaris PPP 4.0" on page 567 uses the following /etc/asppp.cf file.

```
#
ifconfig ipdptp0 plumb mojave gobi up

path
 inactivity_timeout 120     # Approx. 2 minutes
    interface ipdptp0
    peer_system_name Pgobi     # The name we log in with (also in
                                 # /etc/uucp/Systems
```

The file contains the following parameters:

- `ifconfig ipdptp0 plumb mojave gobi up`

  Runs the `ifconfig` command to configure a link from PPP interface `ipdptp0` on the local machine `mojave` to the remote peer `gobi`.

- `inactivity_timeout 120`

  Terminates the line after it has been inactive for two minutes.

- `interface ipdptp0`

  Configures the interface `ipdptp0` on the dial-out machine for asynchronous PPP.

- `peer_system_name Pgobi`

  Gives the name of the remote peer, `Pgobi`.

Example—/etc/uucp/Systems File

The procedure to be shown in "How to Convert From asppp to Solaris PPP 4.0" on page 567 uses the following /etc/uucp/Systems file.

```
#ident  "@(#)Systems    1.5  92/07/14 SMI"  /* from SVR4 bnu:Systems 2.4 */
#

# .
# .
Pgobi Any ACU 38400 15551212 in:--in: mojave word: sand
```

The file contains the following parameters:

- `Pgobi`

  Use `Pgobi` as the host name of the remote peer.

- `Any ACU`

  Tells the modem on the dial-out machine mojave to establish a link with a modem on `Pgobi` at any time of the day. (Any ACU means "look for ACU in the

/etc/uucp/Devices file.")

- 38400

  Sets 38400 as the maximum speed of the link.

- 15551212

  Gives the telephone number of Pgobi.

- in:--in: mojave word: sand

  Defines the login script that is required by Pgobi to authenticate dial-out machine mojave.

Example—/etc/uucp/Devices File

The procedure to be shown in "How to Convert From asppp to Solaris PPP 4.0" on page 567 uses the following /etc/uucp/Devices file.

```
#ident  "@(#)Devices  1.6  92/07/14 SMI"  /* from SVR4 bnu:Devices 2.7 */

.

.
#

TCP,et - - Any TCP -
.
.
#
ACU cua/b - Any hayes
# 0-7 are on a Magma 8 port card
Direct cua/0 - Any direct
Direct cua/1 - Any direct
Direct cua/2 - Any direct
Direct cua/3 - Any direct
Direct cua/4 - Any direct
Direct cua/5 - Any direct
Direct cua/6 - Any direct
Direct cua/7 - Any direct
# a is the console port (aka "tip" line)
Direct cua/a - Any direct
# b is the aux port on the motherboard
Direct cua/b - Any direct
# c and d are high speed sync/async ports
Direct cua/c - Any direct
Direct cua/d - Any direct
```

This file supports any Hayes modem connected to serial port cua/b.

Example—/etc/uucp/Dialers File

The procedure to be shown in "How to Convert From asppp to Solaris PPP 4.0" on page 567 uses the following /etc/uucp/Dialers file.

```
#
#<Much information about modems supported by Solaris UUCP>


penril    =W-P    "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK
ventel    =&-%    "" \r\p\r\c $ k\c ONLINE!
vadic    =K-K    "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE
develcon ""    "" \pr\ps\c est:\007 \E\D\e \n\007
micom    ""    "" \s\c NAME? \D\r\c GO
direct
#
#
#
#   Hayes Smartmodem -- modem should be set with the configuration
#   switches as follows:
#
#       S1 - UP        S2 - UP        S3 - DOWN    S4 - UP
#       S5 - UP        S6 - DOWN    S7 - ?        S8 - DOWN
#
hayes    =,-,    "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r \EATDT\T\r\c CONNECT

<much more information about modems supported by Solaris UUCP>
```

This file contains the chat scripts for all types of modems, including the Hayes modems that are supported in the /etc/UUCP/Dialers file.

Running the asppp2pppd Conversion Script (Task)

The /usr/sbin/asppp2pppd script copies the PPP information that is contained in the /etc/asppp.cf and PPP-related UUCP files to appropriate locations in the Solaris PPP 4.0 files.

Prerequisites

Before doing the next task, you should have done the following:

- Installed the Solaris 9 operating environment on the machine that also has the `asppp` and UUCP configuration files
- Become superuser on the machine with the PPP files, for example, the machine `mojave`

▼ How to Convert From `asppp` to Solaris PPP 4.0

1. **Start the conversion script.**

   ```
   # /usr/sbin/asppp2pppd
   ```

   The conversion process starts and gives you the following screen output.

```
This script provides only a suggested translation for your existing aspppd
configuration.  You will need to evaluate for yourself whether the translation
is appropriate for your operating environment.
Continue [Yn]?
```

2. **Type Y to continue. You receive the following output.**

```
Chat cannot do echo checking; requests for this removed.
Adding 'noauth' to /etc/ppp/options

Preparing to write out translated configuration:
    1 chat file:
        1.  /etc/ppp/chat.Pgobi.hayes
    2 option files:
        2.  /etc/ppp/peers/Pgobi
        3.  /etc/ppp/options
    1 script file:
        4.  /etc/ppp/demand
```

The new Solaris PPP 4.0 files have been generated.

▼ How to View the Results of the Conversion

You can view the Solaris PPP 4.0 files that were created by the `/usr/sbin/asppp2pppd` conversion script at the end of the conversion process. The script displays the following list of options.

```
Enter option number:
        1 - view contents of file on standard output
        2 - view contents of file using /usr/bin/less
        3 - edit contents of file using /usr/bin/vi
        4 - delete/undelete file from list
        5 - rename file in list
        6 - show file list again
        7 - escape to shell (or "!")
        8 - abort without saving anything
```

```
        9 - save all files and exit (default)
Option:
```

1. **Type `1` to view the contents of the files on the screen.**

   The script requests the number of the file you want to view.

   ```
   File number (1 .. 4):
   ```

   The numbers refer to the translated files that are listed during the conversion process, as shown in the previous Step 2.

2. **Type `1` to view the chat file `/etc/ppp/chat.Pgobi.hayes`.**

   ```
   File number (1 .. 4):  1
   "" \d\dA\p\pTE1V1X1Q0S2=255S12=255\r\c
   OK\r ATDT\T\r\c
   CONNECT \c
   in:--in: mojave
   word: sand
   ```

   The chat script contains the modem “chat” information that appears on the hayes line in the sample `/etc/UUCP/Dialers`file. `/etc/ppp/chat.Pgobi.hayes` also contains the login sequence for `Pgobi` that appears in the sample `/etc/UUCP/Systems` file. The chat script is now in the `/etc/ppp/chat.Pgobi.hayes` file.

3. **Type `2` to view the peers file, `/etc/ppp/peers/Pgobi`.**

   ```
   File number (1 .. 4):  2
   /dev/cua/b
   38400
   demand
   idle 120
   connect "/usr/bin/chat -f /etc/ppp/chat.Pgobi.hayes -T '15551212'"
   user NeverAuthenticate
   mojave:gobi
   ```

   The serial port information (`/dev/cua/b`) comes from the `/etc/UUCP/Devices` file. The link speed, idle time, authentication information, and peer names come from the `/etc/asppp.cf` file. “demand” refers to the “demand” script, to be called when the dial-out machine tries to connect to peer Pgobi.

4. **Type `3` to view the `/etc/ppp/options` file created for dial-out machine mohave.**

   ```
   File number (1 .. 4):  3
   #lock
   noauth
   ```

   The information in `/etc/ppp/options` comes from the `/etc/asppp.cf` file.

5. **Type `4` to view the contents of the `demand` script.**

   ```
   File number (1 .. 4):  4
   /usr/bin/pppd file /etc/ppp/peers/Pgobi
   ```

This script, when invoked, runs the `pppd` command, which then reads the `/etc/ppp/peers/Pgobi` to initiate the link between `mohave` and `Pgobi`.

6. **Type `9` to save the created files and exit the conversion script.**

CHAPTER 37

Overview of UUCP

This chapter introduces the UNIX-to-UNIX Copy Program (UUCP) and daemons. The following topics are covered:

UUCP enables computers to transfer files and exchange mail with each other. It also enables computers to participate in large networks such as Usenet.

The Solaris environment provides the Basic Network Utilities (BNU) version of UUCP, also known as HoneyDanBer UUCP. The term *UUCP* denotes the complete range of files and utilities that make up the system, of which the program `uucp` is only a part. The UUCP utilities range from those used to copy files between computers (`uucp` and `uuto`) to those used for remote login and command execution (`cu` and `uux`).

UUCP Hardware Configurations

UUCP supports the following hardware configurations:

Direct links	You can create a direct link to another computer by running RS-232 cables between serial ports on the two machines. Direct links are useful where two computers communicate regularly and are physically close—within 50 feet of each other. You can use a limited distance–modem to increase this distance somewhat.

Telephone lines	Using an automatic call unit (ACU), such as a high-speed modem, your machine can communicate with other computers over standard phone lines. The modem dials the telephone number requested by UUCP. The recipient machine must have a modem capable of answering incoming calls.
Network	UUCP can also communicate over a network running TCP/IP or an other protocol family. After your computer has been established as a host on a network, it can contact any other host connected to the network.

This chapter assumes that your UUCP hardware has already been assembled and configured. If you need to set up a modem, refer to *System Administration Guide: Basic Administration* and the manuals that came with the modem for assistance.

UUCP Software

The UUCP software is automatically included when you run the Solaris installation program and select the entire distribution. Alternatively, you can add it using `pkgadd`. The UUCP programs can be divided into three categories: daemons, administrative programs, and user programs.

UUCP Daemons

The UUCP system has four daemons: `uucico`, `uuxqt`, `uusched`, and `in.uucpd`. These daemons handle UUCP file transfers and command executions. You can also run them manually from the shell, if necessary.

`uucico`	Selects the device used for the link, establishes the link to the remote computer, performs the required login sequence and permission checks, transfers data and execute files, logs results, and notifies the user by mail of transfer completions. `uucico` acts as the "login shell" for UUCP login accounts. When the local `uucico` daemon calls a remote machine, it communicates directly with the remote `uucico` daemon during the session. After all the required files have been created,`uucp`, `uuto`, and `uux` programs execute the `uucico` daemon to contact the remote computer. `uusched` and `Uutry` all execute `uucico`. (See the `uucico`(1M) man page for details.)

`uuxqt`	Executes remote execution requests. It searches the spool directory for execute files (always named `X.`*file*) that have been sent from a remote computer. When an `X.`*file* file is found, `uuxqt` opens it to get the list of data files that are required for the execution. It then checks to see if the required data files are available and accessible. If the files are available, `uuxqt` checks the `Permissions` file to verify that it has permission to execute the requested command. The `uuxqt` daemon is executed by the `uudemon.hour` shell script, which is started by `cron`. (See the `uuxqt`(1M) man page for details.)
`uusched`	Schedules the queued work in the spool directory. `uusched` is initially run at boot time by the `uudemon.hour` shell script, which is started by `cron`. (See the `uusched`(1M) man page for details.) Before starting the `uucico` daemon, `uusched` randomizes the order in which remote computers are called.
`in.uucpd`	Supports UUCP connections over networks. The `inetd` on the remote host invokes `in.uucpd` whenever a UUCP connection is established. `uucpd` then prompts for a login name. `uucico` on the calling host must respond with a login name. `in.uucpd` then prompts for a password, unless one is not required. (See the `in.uucpd`(1M) man page for details.)

UUCP Administrative Programs

Most UUCP administrative programs are in `/usr/lib/uucp`. Most basic database files are in `/etc/uucp`. The only exception is `uulog`, which is in `/usr/bin`. The home directory of the `uucp` login ID is `/usr/lib/uucp`. When running the administrative programs through `su` or `login`, use the `uucp` user ID. It owns the programs and spooled data files.

`uulog`	Displays the contents of a specified computer's log files. Log files are created for each remote computer with which your machine communicates. The log files record each use of `uucp`, `uuto`, and `uux`. (See the `uucp`(1C) man page for details.)
`uucleanup`	Cleans up the spool directory. It is normally executed from the `uudemon.cleanup` shell script, which is started by `cron`. (See the `uucleanup`(1M) man page for details.)
`Uutry`	Tests call-processing capabilities and does moderate debugging. It invokes the `uucico` daemon to establish a communication link between your machine and the remote computer you specify. (See the `Uutry`(1M) man page for details.)

uucheck	Checks for the presence of UUCP directories, programs, and support files. It can also check certain parts of the /etc/uucp/Permissions file for obvious syntactic errors. (See the uucheck(1M) man page for details.)

UUCP User Programs

The UUCP user programs are in /usr/bin. You do not need special permission to use these programs.

cu	Connects your machine to a remote computer so that you can log in to both at the same time. cu enables you to transfer files or execute commands on either machine without dropping the initial link. (See the cu(1C) man page for details.)
uucp	Lets you copy a file from one machine to another. It creates work files and data files, queues the job for transfer, and calls the uucico daemon, which in turn attempts to contact the remote computer. (See the uucp(1C) man page for details.)
uuto	Copies files from the local machine to the public spool directory /var/spool/uucppublic/receive on the remote machine. Unlike uucp, which lets you copy a file to any accessible directory on the remote machine, uuto places the file in an appropriate spool directory and tells the remote user to pick it up with uupick. (See the uuto(1C) man page for details.)
uupick	Retrieves files in /var/spool/uucppublic/receive when files are transferred to a computer using uuto. (See the uuto(1C) man page.)
uux	Creates the work, data, and execute files needed to execute commands on a remote machine. (See the uux(1C) man page for details.)
uustat	Displays the status of requested transfers (uucp, uuto, or uux). It also provides a means of controlling queued transfers. (See the uustat(1C) man page for details.)

UUCP Database Files

A major part of UUCP setup is the configuration of the files making up the UUCP database. These files are in the /etc/uucp directory. You need to edit them to set up UUCP or asppp on your machine. The files include:

`Config`	Contains a list of variable parameters. You can manually set these parameters to configure the network.
`Devconfig`	Used to configure network communications.
`Devices`	Used to configure network communications.
`Dialcodes`	Contains dial-code abbreviations that can be used in the phone number field of `Systems` file entries. Though not required, it can be used by `asppp` as well as UUCP.
`Dialers`	Contains character strings required to negotiate with modems to establish connections with remote computers. It is used by asppp as well as UUCP.
`Grades`	Defines job grades, and the permissions associated with each job grade, that users can specify to queue jobs to a remote computer.
`Limits`	Defines the maximum number of simultaneous `uucico`s, `uuxqt`s, and `uusched`s permitted on your machine.
`Permissions`	Defines the level of access granted to remote hosts that attempt to transfer files or execute commands on your machine.
`Poll`	Defines machines that are to be polled by your system and when they are polled.
`Sysfiles`	Assigns different or multiple files to be used by `uucico` and `cu` as `Systems`, `Devices`, and `Dialers` files.
`Sysname`	Enables you to define a unique UUCP name for a machine in addition to its TCP/IP host name.
`Systems`	Contains information needed by the `uucico` daemon, `cu`, and `asppp` to establish a link to a remote computer. This information includes the name of the remote host, the name of the connecting device associated with the remote host, time when the host can be reached, telephone number, login ID, and password.

Several other files can be considered part of the supporting database but are not directly involved in establishing a link and transferring files.

Configuring UUCP Database Files

The UUCP database consists of the files shown in "`UUCP` Database Files" on page 574. However, basic UUCP configuration involves only the following critical files:

- `/etc/uucp/Systems`
- `/etc/uucp/Devices`
- `/etc/uucp/Dialers`

Because asppp uses some of the UUCP databases, you should understand at least these critical database files if you plan to configure asppp. After these databases are configured, UUCP administration is fairly straightforward. Typically, you edit the Systems file first, then edit the Devices file. You can usually use the default /etc/uucp/Dialers file, unless you plan to add dialers that aren't in the default file. In addition, you might also want to use the following files for basic UUCP and asppp configuration:

- /etc/uucp/Sysfiles
- /etc/uucp/Dialcodes
- /etc/uucp/Sysname

Because these files work closely with one another, you should understand the contents of them all before you change any one of them. A change to an entry in one file might require a change to a related entry in another file. The remaining files listed in "UUCP Database Files" on page 574 are not as critically intertwined.

Note – asppp uses only the files described in this section. It does not use the other UUCP database files.

CHAPTER 38

Administering UUCP

This chapter explains how to start UUCP operations after you have modified the database file relevant to your machines. The chapter contains procedures and troubleshooting information for setting up and maintaining UUCP on machines running the Solaris environment, such as:

UUCP Administration Task Map

The following table provides pointers to the procedures covered in this chapter, as well as a short description of each procedure.

TABLE 38–1 Task Map: UUCP Administration

Task...	Description	For Instructions, Go To ...
Allow remote machines to have access to your system	Edit the `/etc/passwd` file to add entries to identify the machines permitted to access your system	"How to Add UUCP Logins" on page 578
Start UUCP	Use the supplied shell scripts to start UUCP	"How to Start UUCP" on page 580

TABLE 38–1 Task Map: UUCP Administration *(Continued)*

Task...	Description	For Instructions, Go To ...
Enable UUCP to work with TCP/IP	Edit `/etc/inetd.conf` and `/etc/uucp/Systems` files to activate UUCP for TCP/IP	"How to Activate UUCP for TCP/IP" on page 581
Troubleshoot some common UUCP problems	Diagnostic steps to use to check for faulty modems or ACUs Diagnostic steps to use for debugging transmissions	"How to Check for Faulty Modems or ACUs" on page 584 "How to Debug Transmissions" on page 584

Adding UUCP Logins

For incoming UUCP (`uucico`) requests from remote machines to be handled properly, each machine has to have a login on your system.

▼ How to Add UUCP Logins

To allow a remote machine to access your system, you need to add an entry to the `/etc/passwd` file as follows:

1. **Edit the `/etc/passwd` file and add the entry to identify the machine permitted to access your system.**

   A typical entry that you might put into the `/etc/passwd` file for a remote machine permitted to access your system with a UUCP connection would be as follows:

   ```
   Ugobi:*:5:5:gobi:/var/spool/uucppublic:/usr/lib/uucp/uucico
   ```

   By convention, the login name of a remote machine is the machine name preceded by the uppercase letter `U`. Note that the name should not exceed eight characters, so that in some cases you might have to truncate or abbreviate it.

   The previous entry shows that a login request by `Ugobi` is answered by `/usr/lib/uucp/uucico`. The home directory is `/var/spool/uucppublic`. The password is obtained from the `/etc/shadow` file. You must coordinate the password and the login name with the UUCP administrator of the remote machine. The remote administrator must then add an appropriate entry, with login name and unencrypted password, in the remote machine's `Systems` file.

2. **Coordinate your machine name with the UUCP administrators on other systems.**

   Similarly, you must coordinate your machine's name and password with the UUCP administrators of all machines that you want to reach through UUCP.

Starting UUCP

UUCP comes with four shell scripts that poll remote machines, reschedule transmissions, and clean up old log files and unsuccessful transmissions. The scripts are:

- `uudemon.poll`
- `uudemon.hour`
- `uudemon.admin`
- `uudemon.cleanup`

These shell scripts should execute regularly to keep UUCP running smoothly. The `crontab` file to run the scripts is automatically created in `/usr/lib/uucp/uudemon.crontab` as part of the Solaris installation process, if you select the full installation. Otherwise, it is created when you install the UUCP package.

You can also run the UUCP shell scripts manually. The following is the prototype `uudemon.crontab` file that you can tailor for a particular machine:

```
#
#ident  "@(#)uudemon.crontab    1.5     97/12/09 SMI"
#
# This crontab is provided as a sample. For systems
# running UUCP edit the time schedule to suit, uncomment
# the following lines, and use crontab(1) to activate the
# new schedule.
#
#48 8,12,16 * * * /usr/lib/uucp/uudemon.admin
#20 3 * * * /usr/lib/uucp/uudemon.cleanup
#0 * * * * /usr/lib/uucp/uudemon.poll
#11,41 * * * * /usr/lib/uucp/uudemon.hour
```

Note – By default, UUCP operations are disabled. To enable UUCP, edit the time schedule and uncomment the appropriate lines in the `uudemon.crontab` file.

▼ How to Start UUCP

To activate the `uudemon.crontab` file, do the following:

1. **Become superuser.**
2. **Edit the** `/usr/lib/uucp/uudemon.crontab` **file and change entries as required**
3. **Issue:**

```
crontab < /usr/lib/uucp/uudemon.crontab
```

`uudemon.poll` Shell Script

The default `uudemon.poll` shell script reads the `/etc/uucp/Poll` file once an hour. If any machines in the `Poll` file are scheduled to be polled, a work file (`C.`*sysnxxxx*) is placed in the `/var/spool/uucp/`*nodename* directory, where *nodename* represents the UUCP node name of the machine.

The shell script is scheduled to run once an hour, before `uudemon.hour`, so that the work files are there when `uudemon.hour` is called.

`uudemon.hour` Shell Script

The default `uudemon.hour` shell script:

- Calls the `uusched` program to search the spool directories for work files (`C.`) that have not been processed and schedules these files for transfer to a remote machine.
- Calls the `uuxqt` daemon to search the spool directories for execute files (`X.`) that have been transferred to your computer and were not processed at the time they were transferred.

By default, `uudemon.hour` runs twice an hour. You might want it to run more often if you expect high failure rates of calls to remote machines.

uudemon.admin Shell Script

The default uudemon.admin shell script does the following:

- Runs the uustat command with p and q options. The q reports on the status of work files (C.), data files (D.), and execute files (X.) that are queued. The p prints process information for networking processes listed in the lock files (/var/spool/locks).
- Sends resulting status information to the uucp administrative login using mail.

uudemon.cleanup Shell Script

The default uudemon.cleanup shell script does the following:

- Takes log files for individual machines from the /var/uucp/.Log directory, merges them, and places them in the /var/uucp/.Old directory with other old log information.
- Removes work files (C.) seven days old or older, data files (D.) seven days old or older, and execute files (X.) two days old or older from the spool files.
- Returns mail that cannot be delivered to the sender.
- Mails a summary of the status information gathered during the current day to the UUCP administrative login (uucp).

Running UUCP Over TCP/IP

To run UUCP on a TCP/IP network, you need to make a few modifications, as described in this section.

▼ How to Activate UUCP for TCP/IP

1. **Edit the /etc/inetd.conf file and make sure that the following entry is not preceded by a comment mark (#):**

   ```
   uucp stream tcp nowait root /usr/sbin/in.uucpd in.uucpd
   ```

2. **Edit the /etc/uucp/Systems file to make sure that the entries have the following fields :**

   System-Name Time TCP Port networkname Standard-Login-Chat

A typical entry would look like this:

```
rochester Any TCP - ur-seneca login: Umachine password: xxx
```

Notice that the *networkname* field permits you to specify explicitly the TCP/IP host name. This is important for some sites. In the previous example, the site has the UUCP node name `rochester` which is different from its TCP/IP host name `ur-seneca`. Moreover, there could easily be a completely different machine running UUCP that has the TCP/IP host name of `rochester`.

The Port field in the `Systems` file should have the entry `-`. This is equivalent to listing it as `uucp`. In almost every case, the *networkname* is the same as the system name, and the Port field is `-`, which says to use the standard uucp port from the `services` database. The `in.uucpd` daemon expects the remote machine to send its login and password for authentication, and it prompts for them much as `getty` and `login` do.

3. **Edit the `/etc/inet/services` file to set up a port for UUCP:**

```
uucp 540/tcp uucpd # uucp daemon
```

You should not have to change the entry. However, if your machine runs NIS or NIS+ as its name service, you should change the `/etc/nsswitch.conf` entry for `/etc/services` to check `files` first, then check `nis` or `nisplus`.

UUCP Security and Maintenance

After you have set up UUCP, maintenance is straightforward. This section explains ongoing UUCP tasks with regard to security, maintenance, and troubleshooting.

Setting Up UUCP Security

The default `/etc/uucp/Permissions` file provides the maximum amount of security for your UUCP links. The default `Permissions` file contains no entries.

You can set additional parameters for each remote machine to define:

- Ways the remote machine can receive files from your machine
- Directories for which the remote machine has read and write permission
- Commands the remote machine can use for remote execution

A typical `Permissions` entry is:

```
MACHINE=datsun LOGNAME=Udatsun VALIDATE=datsun
COMMANDS=rmail REQUEST=yes SENDFILES=yes
```

This entry allows files to be sent and received (to and from the “normal” UUCP directories, not from anywhere in the system) and causes the UUCP user name to be validated at login time.

Regular UUCP Maintenance

UUCP does not require much maintenance. Apart from making sure that the `crontab` file is in place, as described in the section “How to Start UUCP” on page 580, all you have to worry about is the growth of mail files and the public directory.

Email for UUCP

All email messages generated by the UUCP programs and scripts go to the user ID `uucp`. If you do not log in frequently as that user, you might not realize that mail is accumulating (and consuming disk space). To solve this, make an alias in `/etc/mail/aliases` and redirect that email either to `root` or to yourself and others responsible for maintaining UUCP. Remember to run the `newaliases` command after modifying the `aliases` file.

UUCP Public Directory

The directory `/var/spool/uucppublic` is the one place in every system to which UUCP by default is able to copy files. Every user has permission to change to `/var/spool/uucppublic` and read and write files in it. However, its sticky bit is set, so its mode is 01777. As a result, users cannot remove files that have been copied to it and that belong to `uucp`. Only you, as UUCP administrator logged in as `root` or `uucp`, can remove files from this directory. To prevent the uncontrolled accumulation of files in this directory, you should make sure to clean it up periodically.

If this is inconvenient for users, encourage them to use `uuto` and `uupick` rather than removing the sticky bit, which is set for security reasons. (See the `uuto`(1C) man page for instructions for using `uuto` and `uupick`.) You can also restrict the mode of the directory to only one group of people. If you do not want to run the risk of someone filling your disk, you can even deny UUCP access to it.

Troubleshooting UUCP

These procedures describe how to solve common UUCP problems.

▼ How to Check for Faulty Modems or ACUs

You can check if the modems or other ACUs are not working properly in several ways.

1. **Get counts and reasons for contact failure by running:**

   ```
   uustat -q
   ```

2. **Call over a particular line and print debugging information on the attempt.**

   The line must be defined as `direct` in the `/etc/uucp/Devices` file. (You must add a telephone number to the end of the command line if the line is connected to an autodialer or the device must be set up as `direct`.) Type:

   `cu -d -l` *line*

   where *line* is `/dev/cua/a`.

▼ How to Debug Transmissions

If you cannot contact a particular machine, you can check out communications to that machine with `Uutry` and `uucp`.

1. **To try to make contact by typing:**

   `/usr/lib/uucp/Uutry -r` *machine*

   Replace *machine* with the host name of the machine you are having problems contacting. This command:

   a. **Starts the transfer daemon (`uucico`) with debugging. You can get more debugging information if you are `root`.**

   b. **Directs the debugging output to `/tmp/`*machine*.**

   c. **Prints the debugging output to your terminal by issuing:**

      ```
      tail -f
      ```

      Press Control-c to end output. You can copy the output from `/tmp/`*machine* if you want to save it.

2. **If `Uutry` doesn't isolate the problem, try to queue a job by typing:**

   `uucp -r` *file machine*`\!`*/dir/file*

   Replace *file* by the file you want to transfer, *machine* by the machine you want to copy to, and */dir/file* where the file will be placed on the other machine. The `r` option queues a job but does not start the transfer.

3. **Issue:**

   ```
   Uutry
   ```

If you still cannot solve the problem, you might need to call your local support representative. Save the debugging output; it will help diagnose the problem.

You might also want to decrease or increase the level of debugging provided by `Uutry` through the `-x` *n* option, where *n* indicates the debug level. The default debug level for `Uutry` is 5.

Debug level 3 provides basic information as to when and how the connection is established, but not much information about the transmission itself. Debug level 9, on the other hand, provides exhaustive information about the transmission process. Be aware that debugging occurs at both ends of the transmission. If you intend to use a level higher than 5 on a moderately large text, contact the administrator of the other site and agree on a time for doing so.

Checking the UUCP `/etc/uucp/Systems` File

Verify that you have up-to-date information in your `Systems` file if you are having trouble contacting a particular machine. Some things that might be out of date for a machine are its:

- Phone number
- Login ID
- Password

Checking UUCP Error Messages

UUCP has two types of error messages: `ASSERT` and `STATUS`.

- When a process is aborted, `ASSERT` error messages are recorded in `/var/uucp/.Admin/errors`. These messages include the file name, `sccsid`, line number, and text. These messages usually result from system problems.
- `STATUS` error messages are stored in the `/var/uucp/.Status` directory. The directory contains a separate file for each remote machine your computer attempts to communicate with. These files contain status information on the attempted communication and whether it was successful.

Checking Basic Information

Several commands are available for checking basic networking information:

- Use the `uuname` command to list those machines your machine can contact.

- Use the `uulog` command to display the contents of the log directories for particular hosts.
- Use the `uucheck -v` command to check for the presence of files and directories needed by `uucp`. This command also checks the `Permissions` file and outputs information on the permissions you have set up.

CHAPTER 39

UUCP Reference

This chapter provides reference information for working with UUCP. The following topics are covered:

UUCP `/etc/uucp/Systems` File

The `/etc/uucp/Systems` file contains the information needed by the `uucico` daemon to establish a communication link to a remote computer. It is the first file you need to edit to configure UUCP.

Each entry in the `Systems` file represents a remote computer with which your host communicates. A particular host can have more than one entry. The additional entries represent alternative communication paths that are tried in sequential order. In addition, by default UUCP prevents any computer that does not appear in `/etc/uucp/Systems` from logging in to your host.

Using the `Sysfiles` file, you can define several files to be used as `Systems` files. See "UUCP `/etc/uucp/Sysfiles` File" on page 605 for a description of `Sysfiles`.

Each entry in the `Systems` file has the following format:

System-Name	*Time*	*Type*	*Speed*	*Phone*	*Chat-Script*

The following example shows the fields of the `Systems` file.

EXAMPLE 39–1 Fields in `/etc/uucp/Systems`

```
System-Name Time Type  Speed Phone   Chat-Script
Arabian     Any  ACUEC 38400 111222  Login: Puucp ssword:beledi
```

UUCP System-Name Field

This field contains the node name of the remote computer. On TCP/IP networks, this can be the machine's host name or a name created specifically for UUCP communications through the `/etc/uucp/Sysname` file. See "UUCP `/etc/uucp/Systems` File" on page 587. In Example 39–1, the System-Name field contains an entry for remote host `arabian`.

UUCP Time Field

This field specifies the day of week and time of day when the remote computer can be called. The format of the Time field is:

```
daytime[;retry]
```

The *day* portion can be a list containing some of the following entries:

TABLE 39–1 Day Field

`Su Mo Tu We Th Fr Sa`	For individual days.
`Wk`	For any weekday.
`Any`	For any day.
`Never`	Your host never initiates a call to the remote computer; the call must be initiated by the remote computer. Your host is then operating in *passive mode*.

Example 39–1 shows `Any` in the Time field, indicating that host `arabian` can be called at any time.

The *time* portion should be a range of times specified in 24-hour notation. (Example: `0800-1230` for 8:30 AM to 12:30 PM.) If no *time* portion is specified, any time of day is assumed to be allowed for the call.

A time range that spans 0000 is permitted. For example, `0800-0600` means all times are allowed other than times between 6 AM and 8 AM.

UUCP Retry Subfield

The *Retry* subfield enables you to specify the minimum time (in minutes) before a retry, following a failed attempt. The default wait is 60 minutes. The subfield separator is a semicolon (;). For example, `Any;9` is interpreted as call any time, but wait at least 9 minutes before retrying after a failure occurs.

If you do not specify a *retry* entry, an exponential back-off algorithm is used. What this means is that UUCP starts with a default wait time that grows larger as the number of failed attempts increases. For example, suppose the initial retry time is 5 minutes. If there is no response, the next retry is 10 minutes later. The next retry is 20 minutes later, and so on until the maximum retry time of 23 hours is reached. If *retry* is specified, that is always the retry time. Otherwise, the back-off algorithm is used.

UUCP Type Field

This field contains the device type that should be used to establish the communication link to the remote computer. The keyword used in this field is matched against the first field of `Devices` file entries.

EXAMPLE 39–2 Type Field and `/etc/uucp/Devices` File

```
File Name System-Name  Time  Type    Speed  Phone    Chap-Script

Systems   arabian      Any   ACUEC, g 38400  1112222   ogin: Puucp ssword:beledi
```

You can define the protocol used to contact the system by adding it on to the Type field. The previous example shows how to attach the protocol `g` to the device type `ACUEC`. (For information on protocols, see "UUCP Protocol Definitions in the `Devices` File" on page 598.)

UUCP Speed Field

This field (also known as the Class field) specifies the transfer speed of the device used in establishing the communication link. It can contain a letter and speed (for example, `C1200`, `D1200`) to differentiate between classes of dialers (refer to "UUCP Class Field" on page 596).

Some devices can be used at any speed, so the keyword `Any` can be used. This field must match the Class field in the associated `Devices` file entry:

EXAMPLE 39–3 Speed Field and `/etc/uucp/Devices` File

```
File Name System-Name Time  Type   Speed  Phone   Chap-Script

Systems   eagle       Any   ACU, g  D1200  NY3251  ogin: nuucp ssword: Oakgrass
```

If information is not required for this field, use a dash (`-`) as a place holder for the field.

UUCP Phone Field

This field allows you to specify the telephone number (token) of the remote computer for automatic dialers (port selectors). The telephone number consists of an optional alphabetic abbreviation and a numeric part. If an abbreviation is used, it must be one that is listed in the `Dialcodes` file:

EXAMPLE 39–4 Phone Field Correspondence

```
File Name System-Name  Time  Type  Speed  Phone    Chap-Script

Systems   nubian       Any   ACU   2400   NY5551212 ogin: Puucp ssword:Passuan
```

In the `System-Name` string, an equals sign (`=`) tells the ACU to wait for a secondary dial tone before dialing the remaining digits. A dash (`-`) in the string instructs the ACU to pause four seconds before dialing the next digit.

If your computer is connected to a port selector, you can access other computers connected to that selector. The `Systems` file entries for these remote machines should not have a telephone number in the Phone field. Instead, this field should contain the token to be passed on to the switch. In this way, the port selector knows the remote machine with which your host wants to communicate. (This is usually just the system name.) The associated `Devices` file entry should have a `\D` at the end of the entry to ensure that this field is not translated using the `Dialcodes` file.

UUCP Chat-Script Field

This field (also called the Login field) contains a string of characters called a *chat-script*. The chat-script contains the characters the local and remote machines must pass to each other in their initial conversation. Chat-scripts have the format:

expect send [expect send] ....

expect represents the string that the local host expects to get from the remote host to initiate conversation. *send* is the string the local host sends after it receives the *expect* string from the remote host. A chat-script can have more than one expect-send sequence.

A basic chat-script might contain:

- Login prompt that the local host expects to get from the remote machine
- Login name that the local host sends to the remote machine in order to log in
- Password prompt that the local host expects to get from the remote machine
- Password that the local host sends to the remote machine

The *expect* field can be made up of subfields of the form:

expect[-send-expect]...

where *-send* is sent if the prior *expect* is not successfully read, and *-expect* following the *send* is the next expected string.

For example, with strings `login--login`, the UUCP on the local host expects `login`. If UUCP gets `login` from the remote machine, it goes to the next field. If it does not get `login`, it sends a carriage return, then looks for `login` again. If the local computer initially does not expect any characters, use the characters `""` (NULL string) in the *expect* field. All *send* fields are sent followed by a carriage return unless the *send* string is terminated with a `\c`.

Here is an example of a `Systems` file entry that uses an *expect-send* string:

```
System-Name Time Type Speed Phone   Chap-Script
sonora Any ACUEC 9600 2223333 "" \r \r ogin:-BREAK-ogin: Puucpx ssword: xyzzy
```

This example tells UUCP on the local host to send two carriage-returns and wait for `ogin:` (for `Login:`). If `ogin:` is not received, send a `BREAK`. When you do get `ogin:` send the login name `Puucpx`. When you get `ssword:` (for `Password:`), send the password `xyzzy`.

The following table lists some useful escape characters.

TABLE 39–2 Escape Characters Used in `Systems` File Chat-Script

Escape Character	Meaning
`\b`	Sends or expects a backspace character.
`\c`	If at the end of a string, suppresses the carriage return that is normally sent. Ignored otherwise.
`\d`	Delays 1–3 seconds before sending more characters.
`\E`	Starts echo checking. (From this point on, whenever a character is transmitted, it waits for the character to be received before doing anything else.)
`\e`	Echoes check-off.
`\H`	Ignores one hangup. Use this option for dialback modems.
`\K`	Sends a BREAK character.
`\M`	Turns on `CLOCAL` flag.
`\m`	Turns off `CLOCAL` flag.
`\n`	Sends or expects a newline character.
`\N`	Sends a NULL character (ASCII NUL).
`\p`	Pauses for approximately 1/4 to 1/2 second.
`\r`	Sends or expects a carriage return.
`\s`	Sends or expects a space character.
`\t`	Sends or expects a tab character.
`EOT`	Sends an EOT followed by newline twice.
`BREAK`	Sends a break character.
`\`*ddd*	Sends or expects the character represented by the octal digits (*ddd*).

Enabling Dialback Through the Chat-Script

Some companies set up dial-in servers to handle calls from remote computers. For example, your company might have a dial-in server with a dialback modem that employees can call from their home computers. After the dial-in server identifies the remote machine, it disconnects the link to the remote machine and then calls the remote machine back. The communications link is then reestablished.

You can facilitate dialback by using the `\H` option in the `Systems` file chat-script at the place where dialback should occur. Include the `\H` as part of an expect string at the place where the dial-in server is expected to hang up.

For example, suppose the chat-script that calls a dial-in server contains the following string:

```
INITIATED\Hogin:
```

The UUCP dialing facility on the local machine expects to get the characters `INITIATED` from the dial-in server. After the `INITIATED` characters have been matched, the dialing facility flushes any subsequent characters it receives until the dial-in server hangs up. The local dialing facility then waits until it receives the next part of the expect string, the characters `ogin:`, from the dial-in server. When it receives the `ogin:`, the dialing facility then continues through the chat-script.

You need not have a string of characters directly preceding or following the `\H`, as shown in the previous sample string.

UUCP Hardware Flow Control

You can also use the pseudo-send `STTY=value` string to set modem characteristics. For instance, `STTY=crtscts` enables hardware flow control. `STTY` accepts all `stty` modes. See the `stty`(1) and `termio`(7I) man pages for complete details.

The following example would enable hardware flow control in a `Systems` file entry:

```
System-Name Time Type Speed Phone   Chap-Script
unix Any ACU 2400 12015551212 "" \r login:-\r-login:-\r-login:
nuucp password: xxx "" \ STTY=crtscts
```

This pseudo-send string can also be used in entries in the `Dialers` file.

UUCP Setting Parity

In some cases, you have to reset the parity because the system that you are calling checks port parity and drops the line if it is wrong. The expect-send couplet `""` `P_ZERO` sets the high-order bit (parity bit) to 0. For example:

```
System-Name Time Type Speed Phone   Chap-Script
unix Any ACU 2400 12015551212 "" P_ZERO "" \r login:-\r-login:-\r-login:
nuucp password: xxx
```

In the same manner, `P_EVEN` sets parity to even (the default), `P_ODD` sets odd parity, and `P_ONE` sets the parity bit to 1.

The parity couplet can be inserted anywhere in the chat-script. It applies to all information in the chat-script following the `""` `P_ZERO`. It can also be used in entries in the `Dialers` file.

UUCP /etc/uucp/Devices File

The /etc/uucp/Devices file contains information for all the devices that can be used to establish a link to a remote computer. These devices include ACUs—which includes modern, high-speed modems—direct links, and network connections.

Here is an entry in /etc/uucp/Devices for a US Robotics V.32bis modem attached to port A and running at 38,400 bps.

```
Type   Line  Line2 Class Dialer-Token-Pairs
ACUEC  cua/a -     38400 usrv32bis-ec
```

Each field is described in the next section.

UUCP Type Field

This field describes the type of link that the device establishes. It can contain one of the keywords described in the sections that follow.

Direct Keyword

The Direct keyword appears mainly in entries for cu connections. This keyword indicates that the link is a direct link to another computer or a port selector. Make a separate entry for each line that you want to reference through the -l option of cu.

ACU Keyword

The ACU keyword indicates that the link to a remote computer (whether through cu, UUCP, asppp, or Solaris PPP 4.0) is made through a modem. This modem can be connected either directly to your computer or indirectly through a port selector.

Port Selector

This is a variable that is replaced in the Type field by the name of a port selector. Port selectors are devices attached to a network that prompt for the name of a calling modem, then grant access. The file /etc/uucp/Dialers contains caller scripts only for the micom and develcon port selectors. You can add your own port selector entries to the Dialers file. (See "UUCP /etc/uucp/Dialers File" on page 599 for more information.)

Sys-Name

This variable is replaced by the name of a machine in the Type field, indicating that the link is a direct link to this particular computer. This naming scheme is used to associate the line in this `Devices` entry to an entry in `/etc/uucp/Systems` for the computer *Sys-Name*.

Type Field and `/etc/uucp/Systems` File

Example 39–5 shows a comparison between the fields in `/etc/uucp/Devices` and fields in `/etc/uucp/Systems`. The titles of each column apply only to fields in the `Devices` file.

The keyword used in the Type field of the `Devices` file is matched against the third field of the `Systems` file entries. In the `Devices` file, the Type field has the entry ACUEC, indicating an automatic call unit, in this case a V.32bis modem. This value is matched against the third field in the `Systems` file, which also contains the entry ACUEC. (See "UUCP `/etc/uucp/Systems` File" on page 587 for more information.)

EXAMPLE 39–5 Type Field and `/etc/uucp/Systems` File Equivalent

```
File Name Type   Line  Line2 Class Dialer-Token-Pairs

Devices   ACUEC  cua/a -     38400 usrv32bis-ec

System    nubian Any   ACUEC 38400 9998888 "" \d\d\r\n\c-ogin-\r\n\c-ogin.......
```

UUCP Line Field

This field contains the device name of the line (port) associated with the `Devices` entry. For instance, if the modem associated with a particular entry were attached to the `/dev/cua/a` device (serial port A), the name entered in this field would be `cua/a`. An optional modem control flag, `M`, can be used in the Line field to indicate that the device should be opened without waiting for a carrier. For example:

```
cua/a,M
```

UUCP Line2 Field

This field is a placeholder. Always use a dash (`-`) here. 801 type dialers, which are not supported in the Solaris environment, use the Line2 field. Non-801 dialers do not normally use this configuration, but still require a hyphen in this field.

UUCP Class Field

The Class field contains the speed of the device, if the keyword `ACU` or `Direct` is used in the Type field. However, it can contain a letter and a speed (for example, `C1200`, `D1200`) to differentiate between classes of dialers (Centrex or Dimension PBX).

This is necessary because many larger offices can have more than one type of telephone network: one network might be dedicated to serving only internal office communications while another handles the external communications. In such a case, it becomes necessary to distinguish which line(s) should be used for internal communications and which should be used for external communications.

The keyword used in the Class field of the `Devices` file is matched against the Speed field of `Systems` file.

EXAMPLE 39–6 UUCP Class Field

```
File Name  Type   Line  Line2  Class  Dialer-Token-Pairs

Devices    ACU    cua/a -      D2400  hayes
```

Some devices can be used at any speed, so the keyword `Any` can be used in the Class field. If `Any` is used, the line matches any speed requested in the Speed field of the `Systems` file. If this field is `Any` and the `Systems` file Speed field is `Any`, the speed defaults to 2400 bps.

UUCP Dialer-Token-Pairs Field

The Dialer-Token-Pairs (DTP) field contains the name of a dialer and the token to pass it. The DTP field has this syntax:

dialer token [dialer token]

The *dialer* portion can be the name of a modem, a port monitor, or it can be `direct` or `uudirect` for a direct-link device. You can have any number of dialer-token pairs; if not present, it is taken from a related entry in the `Systems` file. The *token* portion can be supplied immediately following the dialer portion.

The last dialer token pair might not be present, depending on the associated dialer. In most cases, the last pair contains only a *dialer* portion. The *token* portion is retrieved from the Phone field of the associated `Systems` file entry.

A valid entry in the *dialer* portion can be defined in the `Dialers` file or can be one of several special dialer types. These special dialer types are compiled into the software and are therefore available without having entries in the `Dialers` file. The following table shows the special dialer types.

TABLE 39–3 Dialer-Token Pairs

`TCP`	TCP/IP network
`TLI`	Transport Level Interface Network (without STREAMS)
`TLIS`	Transport Level Interface Network (with STREAMS)

See "UUCP Protocol Definitions in the `Devices` File" on page 598 for more information.

Structure of the Dialer-Token-Pairs Field

The DTP field can be structured four different ways, depending on the device associated with the entry:

- Directly connected modem

  If a modem is connected directly to a port on your computer, the DTP field of the associated `Devices` file entry has only one pair. This pair would normally be the name of the modem. This name is used to match the particular `Devices` file entry with an entry in the `Dialers` file. Therefore, the Dialer field must match the first field of a `Dialers` file entry.

EXAMPLE 39–7 Dialers Field for Direct Connect Modem

```
Dialers   hayes =,-,  ""          \\dA\pTE1V1X1Q0S2=255S12=255\r\c
                                  \EATDT\T\r\c CONNECT
```

Notice that only the dialer portion (`hayes`) is present in the DTP field of the `Devices` file entry. This means that the *token* to be passed on to the dialer (in this case the phone number) is taken from the `Phone` field of a `Systems` file entry. (`\T` is implied, as described in Example 39–9.)

- Direct link – For a direct link to a particular computer, the DTP field of the associated entry would contain the keyword `direct`. This is true for both types of direct-link entries, `Direct` and *Sys-Name* (refer to "UUCP Type Field" on page 594).
- Computers on the same port selector – If a computer with which you want to communicate is on the same port selector switch as your computer, your computer must first access the switch. The switch then makes the connection to the other computer. This type of entry has only one pair. The *dialer* portion is used to match a `Dialers` file entry.

EXAMPLE 39–8 UUCP Dialers Field for Computers on Same Port Selector

```
Dialers    develcon ,""   ""           \pr\ps\c est:\007 \E\D\e \007
```

As shown, the *token* portion is left blank. This indicates that it is retrieved from the `Systems` file. The `Systems` file entry for this computer contains the token in the Phone field, which is normally reserved for the phone number of the computer. (Refer to "UUCP `/etc/uucp/Systems` File" on page 587.) This type of DTP contains an escape character (`\D`), which ensures that the contents of the Phone field not interpreted as a valid entry in the `Dialcodes` file.

- Modems connected to port selector – If a high-speed modem is connected to a port selector, your computer must first access the port selector switch. The switch makes the connection to the modem. This type of entry requires two dialer-token-pairs. The *dialer* portion of each pair (fifth and seventh fields of entry) is used to match entries in the `Dialers` file, as shown below.

EXAMPLE 39–9 UUCP Dialers Field for Modems Connected to Port Selector

```
Dialers    develcon ""    ""     \pr\ps\c  est:\007    \E\D\e      \007

Dialers    ventel   =&-%   t""   \r\p\r\c  $           <K\T%\r>\c  ONLINE!
```

In the first pair, `develcon` is the dialer and `vent` is the token that is passed to the Develcon switch to tell it which device (such as Ventel modem) to connect to your computer. This token is unique for each port selector, as each switch can be set up differently. After the Ventel modem has been connected, the second pair is accessed, where Ventel is the dialer and the token is retrieved from the `Systems` file.

Two escape characters can appear in a DTP field:

- `\T` – Indicates that the Phone (*token*) field should be translated using the `/etc/uucp/Dialcodes` file. This escape character is normally placed in the `/etc/uucp/Dialers` file for each caller script associated with a modem (Hayes, US Robotics, and so on). Therefore, the translation does not take place until the caller script is accessed.
- `\D` – Indicates that the Phone (*token*) field should not be translated using the `/etc/uucp/Dialcodes` file. If no escape character is specified at the end of a `Devices` entry, the `\D` is assumed (default). A `\D` is also used in the `/etc/uucp/Dialers` file with entries associated with network switches (`develcon` and `micom`).

UUCP Protocol Definitions in the `Devices` File

You can define the protocol to use with each device in `/etc/uucp/Devices`. This is usually unnecessary because you can use the default or define the protocol with the particular system you are calling. (Refer to "UUCP `/etc/uucp/Systems` File" on page 587.) If you do specify the protocol, you must use the form:

Type,Protocol [parameters]

For example, you can use `TCP,te` to specify the TCP/IP protocol.

The following table shows the available protocols for the `Devices` file.

TABLE 39–4 Protocols Used in `/etc/uucp/Devices`

Protocol	Description
`t`	This protocol is commonly used for transmissions over TCP/IP and other reliable connections. It assumes error-free transmissions.
`g`	This is UUCP's native protocol. It is slow, reliable, and good for transmission over noisy telephone lines.
`e`	This protocol assumes transmission over error-free channels that are message oriented (as opposed to byte-stream oriented, like TCP/IP).
`f`	This protocol is used for transmission over X.25 connections. It relies on flow control of the data stream, and is meant for working over links that can (almost) be guaranteed to be error-free, specifically X.25/PAD links. A checksum is carried out over a whole file only. If a transport fails, the receiver can request retransmission(s).

Here is an example showing a protocol designation for a device entry:

```
TCP,te - - Any TCP -
```

This example indicates that, for device `TCP`, try to use the `t` protocol. If the other end refuses, use the `e` protocol.

Neither `e` nor `t` is appropriate for use over modems. Even if the modem assures error-free transmission, data can still be dropped between the modem and the CPU.

UUCP `/etc/uucp/Dialers` File

The `/etc/uucp/Dialers` file contains dialing instructions for many commonly used modems. You probably do not need to change or add entries to this file unless you plan to use a nonstandard modem or plan to customize your UUCP environment. Nevertheless, you should understand what is in the file and how it relates to the `Systems` and `Devices` file.

The text specifies the initial conversation that must take place on a line before it can be made available for transferring data. This conversation, often referred to as a chat-script, is usually a sequence of ASCII strings that is transmitted and expected, and it is often used to dial a phone number.

As shown in the examples in "UUCP /etc/uucp/Devices File" on page 594, the fifth field in a Devices file entry is an index into the Dialers file or a special dialer type (TCP, TLI, or TLIS). The uucico daemon attempts to match the fifth field in the Devices file with the first field of each Dialers file entry. In addition, each odd-numbered Devices field, starting with the seventh position is used as an index into the Dialers file. If the match succeeds, the Dialers entry is interpreted to perform the dialer conversation.

Each entry in the Dialers file has the following format:

dialer	*substitutions*	*expect-send*

The following example shows the entry for a US Robotics V.32bis modem.

EXAMPLE 39–10 /etc/uucp/Dialers File Entry

```
Dialer     Substitution  Expect-Send
usrv32bis-e  =,-,  ""      dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r
                           \EATDT\T\r\c CONNECT\s14400/ARQ STTY=crtscts
```

The Dialer field matches the fifth and additional odd-numbered fields in the Devices file. The Substitutions field is a translate string: the first of each pair of characters is mapped to the second character in the pair. This is usually used to translate = and - into whatever the dialer requires for "wait for dial tone" and "pause."

The remaining expect-send fields are character strings.

The following example shows some sample entries in the Dialers file, as distributed when you install UUCP as part of the Solaris installation program.

EXAMPLE 39–11 Excerpts From /etc/uucp/Dialers

```
penril    =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK

ventel    =&-%    "" \r\p\r\c $ <K\T%%\r>\c ONLINE!

vadic    =K-K    "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE

develcon    ""    "" \pr\ps\c est:\007

\E\D\e \n\007 micom    ""    "" \s\c NAME? \D\r\c GO

hayes    =,-,    "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r \EATDT\T\r\c CONNECT

#   Telebit TrailBlazer
tb1200    =W-,    "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=2\r\c OK\r
\EATDT\T\r\c CONNECT\s1200
tb2400    =W-,    "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=3\r\c OK\r
\EATDT\T\r\c CONNECT\s2400
```

EXAMPLE 39–11 Excerpts From `/etc/uucp/Dialers` *(Continued)*

```
tbfast    =W-,    "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=255\r\c OK\r
\EATDT\T\r\c CONNECT\sFAST

# USrobotics, Codes, and DSI modems

dsi-ec  =,-,    "" \dA\pTE1V1X5Q0S2=255S12=255*E1*F3*M1*S1\r\c OK\r \EATDT\T\r\c
CONNECT\sEC STTY=crtscts,crtsxoff

dsi-nec =,-,    "" \dA\pTE1V1X5Q0S2=255S12=255*E0*F3*M1*S1\r\c OK\r \EATDT\T\r\c CONNECT
STTY=crtscts,crtsxoff

usrv32bis-ec =,-,  "" \dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r \EATDT\T\r\c
CONNECT\s14400/ARQ STTY=crtscts,crtsxoff

usrv32-nec =,-, "" \dA\pT&FE1V1X1Q0S2=255S12=255&A0&H1&M0&B0&W\r\c OK\r \EATDT\T\r\c
CONNECT STTY=crtscts,crtsxoff

codex-fast =,-, "" \dA\pT&C1&D2*MF0*AA1&R1&S1*DE15*FL3S2=255S7=40S10=40*TT5&W\r\c OK\r
\EATDT\T\r\c CONNECT\s38400 STTY=crtscts,crtsxoff

tb9600-ec =W-,  "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=6\r\c OK\r
\EATDT\T\r\cCONNECT\s9600 STTY=crtscts,crtsxoff

tb9600-nec =W-, "" \dA\pA\pA\pTE1V1X1Q0S2=255S12=255S50=6S180=0\r\c OK\r \EATDT\T\r\c
CONNECT\s9600 STTY=crtscts,crtsxoff
```

The following table lists escape characters commonly used in the send strings in the `Dialers` file.

TABLE 39–5 Backslash Characters for `/etc/uucp/Dialers`

Character	Description
`\b`	Sends or expects a backspace character.
`\c`	No newline or carriage return.
`\d`	Delays (approximately 2 seconds).
`\D`	Phone number or token without `Dialcodes` translation.
`\e`	Disables echo checking.
`\E`	Enables echo checking (for slow devices).
`\K`	Insert a Break character
`\n`	Sends newline.

TABLE 39–5 Backslash Characters for `/etc/uucp/Dialers` *(Continued)*

Character	Description
`\nnn`	Sends octal number. Additional escape characters that can be used are listed in the section "UUCP `/etc/uucp/Systems` File" on page 587.
`\N`	Sends or expects a NULL character (ASCII NUL)
`\p`	Pauses (approximately 12–14 seconds).
`\r`	Returns.
`\s`	Sends or expects a space character.
`\T`	Phone number or token with `Dialcodes` translation.

Here is a `penril` entry in the `Dialers` file:

```
penril =W-P "" \d > Q\c : \d- > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK
```

First, the substitution mechanism for the phone number argument is established, so that any `=` is replaced with a `W` (wait for dial tone) and any `-` with a `P` (pause).

The handshake given by the remainder of the line works as listed:

- `""` – Waits for nothing. (that is, proceed to the next step)
- `\d` – Delays 2 seconds, then send a carriage-return
- `>` – Waits for a `>`
- `Q\c` – Sends a `Q` without a carriage return
- `:` – Expects a `:`
- `\d-` – Delays 2 seconds, sends a `-` and a carriage-return
- `>` – Waits for a `>`
- `s\p9\c` – Sends an `s`, pauses, sends a `9` with no carriage return
- `)-W\p\r\ds\p9\c-)` – Waits for a `)`. If it is not received, processes the string between the `-` characters as follows. Sends a `W`, pauses, sends a carriage return, delays, sends an `s`, pauses, sends a `9`, without a carriage return, then waits for the `)`.
- `y\c` – Sends a `y` with no carriage return
- `:` – Waits for a `:`
- `\E\TP` - Enables echo checking. (From this point on, whenever a character is transmitted, it waits for the character to be received before doing anything else.) Then, sends the phone number. The `\T` means take the phone number passed as an argument and applies the `Dialcodes` translation and the modem function translation specified by field 2 of this entry. Then sends a `P` and a carriage return.
- `>` – Waits for a `>`

- `9\c` – Sends a `9` without a newline
- `OK` – Waits for the string `OK`

UUCP Hardware Flow Control

You can also use the pseudo-send `STTY=value` string to set modem characteristics. For instance, `STTY=crtscts` enables outbound hardware flow control; `STTY=crtsxoff` enables inbound hardware flow control; and `STTY=crtscts,crtsxoff` enables both outbound and inbound hardware flow control.

`STTY` accepts all the `stty` modes. See the `stty`(1) and `termio`(7I) man pages.

The following example would enable hardware flow control in a `Dialers` entry:

```
dsi =,-, "" \dA\pTE1V1X5Q0S2=255S12=255*E1*F3*M1*S1\r\c OK\r \EATDT\T\r\c
CONNECT\sEC STTY=crtscts
```

This pseudo-send string can also be used in entries in the `Systems` file.

UUCP Setting Parity

In some cases, you have to reset the parity because the system that you are calling checks port parity and drops the line if it is wrong. The expect-send couplet `P_ZERO` sets parity to zero:

```
foo =,-, "" P_ZERO "" \dA\pTE1V1X1Q0S2=255S12=255\r\c OK\r\EATDT\T\r\c CONNECT
```

In the same manner, `P_EVEN` sets it to even (the default); `P_ODD` sets it to odd; and `P_ONE` sets it to one. This pseudo-send string can also be used in entries in the `Systems` file.

Other Basic UUCP Configuration Files

The files in this section can be used in addition to the `Systems`, `Devices`, and `Dialers` file when doing basic UUCP configuration.

UUCP /etc/uucp/Dialcodes File

The /etc/uucp/Dialcodes file enables you to define dial-code abbreviations that can be used in the Phone field in the /etc/uucp/Systems file. You can use the Dialcodes files to provide additional information about a basic phone number that is used by several systems at the same site.

Each entry has the format:

abbreviation dial-sequence

where *abbreviation* represents the abbreviation used in the Phone field of the Systems file and *dial-sequence* represents the dial sequence passed to the dialer when that particular Systems file entry is accessed. The following table shows the correspondences between the two files.

TABLE 39–6 Correspondences Between Dialcodes and Systems Files

	Field Names					
Dialcodes	*Abbreviation*	Dial-Sequence				
Systems	System-Name	Time	Type	Speed	*Phone*	Chat-Script

The following table contains sample entries in a Dialcodes file.

TABLE 39–7 Entries in the Dialcodes File

Abbreviation	**Dial-sequence**
NY	1=212
jt	9+847

In the first row, NY is the abbreviation to appear in the Phone field of the Systems file. For example, the Systems file might have the entry:

```
NY5551212
```

When uucico reads NY in the Systems file, it searches the Dialcodes file for NY and obtains the dialing sequence 1=212. This is the dialing sequence needed for any phone call to New York City. It includes the number 1, an equal sign (=) meaning pause and wait for a secondary dial tone, and the area code 212. uucico sends this information to the dialer, then returns to the Systems file for the remainder of the phone number, 5551212.

The entry `jt 9=847-` would work with a Phone field in the `Systems` file such as `jt7867`. When `uucico` reads the entry containing `jt7867` in the `Systems` file, it sends the sequence `9=847-7867` to the dialer, if the token in the dialer-token pair is `\T`.

UUCP `/etc/uucp/Sysfiles` File

The `/etc/uucp/Sysfiles` file lets you assign different files to be used by `uucp` and `cu` as `Systems`, `Devices`, and `Dialers` files. (For more information on `cu`, see the `cu`(1C) man page.) You might want to use `Sysfiles` for:

- Different `Systems` files, so that requests for login services can be made to different addresses than `uucp` services.
- Different `Dialers` files, so that you can assign different handshaking for `cu` and `uucp`.
- Multiple `Systems`, `Dialers`, and `Devices` files. The `Systems` file in particular can become large, making it more convenient to split it into several smaller files.

The format of the `Sysfiles` file is:

```
service=w systems=x:x dialers=y:y devices=z:z
```

w represents `uucico`, `cu`, or both separated by a colon. *x* represents one or more files to be used as the `Systems` file, with each file name separated by a colon and read in the order presented. *y* represents one or more files to be used as the `Dialers` file. *z* is one or more files to be used as the `Devices` file.

Each file name is assumed to be relative to the `/etc/uucp` directory, unless a full path is given.

The following sample, `/etc/uucp/Sysfiles` defines a local `Systems` file (`Local_Systems`) in addition to the standard `/etc/uucp/Systems` file:

```
service=uucico:cu systems=Systems :Local_Systems
```

When this entry is in `/etc/uucp/Sysfiles`, both `uucico` and `cu` first check in the standard `/etc/uucp/Systems`. If the system they are trying to call doesn't have an entry in that file, or if the entries in the file fail, then they look in `/etc/uucp/Local_Systems`.

Given the previous entry, `cu` and `uucico` share the `Dialers` and `Devices` files.

When different `Systems` files are defined for `uucico` and `cu` services, your machine stores two different lists of `Systems`. You can print the `uucico` list using the `uuname` command or the `cu` list using the `uuname -C` command. Another example of the file, where the alternate files are consulted first and the default files are consulted in case of need is:

```
service=uucico systems=Systems.cico:Systems
  dialers=Dialers.cico:Dialers \
devices=Devices.cico:Devices
  service=cu systems=Systems.cu:Systems \
dialers=Dialers.cu:Dialers \
  devices=Devices.cu:Devices
```

UUCP /etc/uucp/Sysname File

Every machine that uses UUCP must have an identifying name, often referred to as the *node name*. This is the name that appears in the remote machine's `/etc/uucp/Systems` file, along with the chat-script and other identifying information. Normally, UUCP uses the same node name as is returned by the `uname -n` command, which is also used by TCP/IP.

You can specify a UUCP node name independent of the TCP/IP host name by creating the `/etc/uucp/Sysname` file. The file has a one-line entry containing the UUCP node name for your system.

UUCP /etc/uucp/Permissions File

The `/etc/uucp/Permissions` file specifies the permissions that remote computers have with respect to login, file access, and command execution. Some options restrict the remote computer's ability to request files and its ability to receive files queued by the local machine. Another option is available that specifies the commands that a remote machine can execute on the local computer.

UUCP Structuring Entries

Each entry is a logical line, with physical lines terminated by a backslash (\) to indicate continuation. Entries are made up of options delimited by blank space. Each option is a name-value pair in the following format:

name=value

Values can be colon-separated lists. No blank space is allowed within an option assignment.

Comment lines begin with a pound sign (#), and they occupy the entire line up to a newline character. Blank lines are ignored (even within multiple-line entries).

The types of Permissions file entries are:

- LOGNAME – Specifies the permissions that take effect when a remote computer logs in to (calls) your computer.

Note – When a remote machine calls you, its identity is questionable unless it has a unique login and verifiable password.

- MACHINE – Specifies permissions that take effect when your computer logs in to (calls) a remote computer.

LOGNAME entries contain a LOGNAME option and MACHINE entries contain a MACHINE option. One entry can contain both options.

UUCP Considerations

When using the Permissions file to restrict the level of access granted to remote computers, you should consider the following:

- All login IDs used by remote computers to log in for UUCP communications must appear in one and only one LOGNAME entry.
- Any site that is called having a name that does not appear in a MACHINE entry, has the following default permissions or restrictions:
  - Local send and receive requests are executed.
  - The remote computer can send files to your computer's /var/spool/uucppublic directory.
  - The commands sent by the remote computer for execution on your computer must be one of the default commands, usually rmail.

UUCP REQUEST Option

When a remote computer calls your computer and requests to receive a file, this request can be granted or denied. The REQUEST option specifies whether the remote computer can request to set up file transfers from your computer. The string REQUEST=yes specifies that the remote computer can request to transfer files from your computer. The string REQUEST=no specifies that the remote computer cannot request to receive files from your computer. This is the default value; it is used if the REQUEST option is not specified. The REQUEST option can appear in either a LOGNAME (remote computer calls you) entry or a MACHINE (you call remote computer) entry.

UUCP SENDFILES Option

When a remote computer calls your computer and completes its work, it can attempt to take work your computer has queued for it. The SENDFILES option specifies whether your computer can send the work queued for the remote computer.

The string SENDFILES=yes specifies that your computer can send the work that is queued for the remote computer as long as it is logged in as one of the names in the LOGNAME option. This string is *mandatory* if you have entered Never in the Time field of /etc/uucp/Systems. This designation sets up your local machine in passive mode; it is not allowed to initiate a call to this particular remote computer. (See "UUCP /etc/uucp/Systems File" on page 587 for more information.)

The string SENDFILES=call specifies that files queued in your computer are sent only when your computer calls the remote computer. The call value is the default for the SENDFILES option. This option is only significant in LOGNAME entries because MACHINE entries apply when calls are made out to remote computers. If the option is used with a MACHINE entry, it is ignored.

UUCP MYNAME Option

This option enables you to designate a unique UUCP node name for your computer in addition to its TCP/IP host name, as returned by the hostname command. For instance, if you have unknowingly given your host the same name as that of some other system, you might want to set the MYNAME option of the Permissions file. Or if you want your organization to be known as widget but all your modems are connected to a machine with the host name gadget, you can have an entry in gadget's Permissions file that says:

```
service=uucico systems=Systems.cico:Systems
  dialers=Dialers.cico:Dialers \
  devices=Devices.cico:Devices
service=cu systems=Systems.cu:Systems \
  dialers=Dialers.cu:Dialers \
  devices=Devices.cu:Devices
```

Now the system world can log in to the machine gadget as if it were logging in to widget. In order for machine world to know you also by the aliased name widget when you call it, you can have an entry that says:

```
MACHINE=world MYNAME=widget
```

You can also use the MYNAME option for testing purposes, as it allows your machine to call itself. However, because this option could be used to mask the real identity of a machine, you should use the VALIDATE option, as described in "UUCP VALIDATE Option" on page 611.

UUCP READ and WRITE Options

These options specify the various parts of the file system that uucico can read from or write to. You can designate READ and WRITE options with either MACHINE or LOGNAME entries.

The default for both the READ and WRITE options is the uucppublic directory, as shown in the following strings:

```
READ=/var/spool/uucppublic WRITE=/var/spool/uucppublic
```

The strings READ=/ and WRITE=/ specify permission to access any file that can be accessed by a local user with Other permissions.

The value of these entries is a colon-separated list of path names. The READ option is for requesting files, and the WRITE option is for depositing files. One of the values must be the prefix of any full path name of a file coming in or going out. To grant permission to deposit files in /usr/news as well as the public directory, use the following values with the WRITE option:

```
WRITE=/var/spool/uucppublic:/usr/news
```

If the READ and WRITE options are used, all path names must be specified because the path names are not added to the default list. For instance, if the /usr/news path name were the only one specified in a WRITE option, permission to deposit files in the public directory would be denied.

Be careful which directories you make accessible for reading and writing by remote systems. For example, the /etc directory contains many critical system files; remote users should not have permission to deposit files in this directory.

UUCP NOREAD and NOWRITE Options

The NOREAD and NOWRITE options specify exceptions to the READ and WRITE options or defaults. The entry:

```
READ=/ NOREAD=/etc WRITE=/var/spool/uucppublic
```

permits reading any file except those in the /etc directory (and its subdirectories—remember, these are prefixes). It permits writing only to the default /var/spool/uucppublic directory. NOWRITE works in the same manner as the NOREAD option. You can use the NOREAD and NOWRITE options in both LOGNAME and MACHINE entries.

UUCP CALLBACK Option

You can use the CALLBACK option in LOGNAME entries to specify that no transaction takes place until the calling system is called back. The two reasons to set up CALLBACK are: For security purposes; if you call back a machine, you can be sure it is the right machine. For accounting purposes; if you are doing long data transmissions, you can choose the machine that is billed for the longer call.

The string CALLBACK=yes specifies that your computer must call the remote computer back before any file transfers can take place.

The default for the CALLBACK option is CALLBACK=no. If you set CALLBACK to yes, the permissions that affect the rest of the conversation must be specified in the MACHINE entry corresponding to the caller. Do not specify these permissions in the LOGNAME, or in the LOGNAME entry that the remote machine might have set for your host.

Note – If two sites have the CALLBACK option set for each other, a conversation never gets started.

UUCP COMMANDS Option

Caution – The COMMANDS option can compromise the security of your system. Use it with extreme care.

You can use the COMMANDS option in MACHINE entries to specify the commands that a remote computer can execute on your machine. The uux program generates remote execution requests and queues them to be transferred to the remote computer. Files and commands are sent to the target computer for remote execution. This is an exception to the rule that MACHINE entries apply only when your system calls out.

Note that COMMANDS is not used in a LOGNAME entry; COMMANDS in MACHINE entries defines command permissions, whether you call the remote system or it calls you.

The string COMMANDS=rmail specifies the default commands that a remote computer can execute on your computer. If a command string is used in a MACHINE entry, the default commands are overridden. For instance, the entry:

```
MACHINE=owl:raven:hawk:dove COMMANDS=rmail:rnews:lp
```

overrides the COMMAND default so that the computers named owl, raven, hawk, and dove can now execute rmail, rnews, and lp on your computer.

In addition to the names as just specified, there can be full path names of commands. For example:

```
COMMANDS=rmail:/usr/local/rnews:/usr/local/lp
```

specifies that command `rmail` uses the default search path. The default search path for UUCP is `/bin` and `/usr/bin`. When the remote computer specifies `rnews` or `/usr/local/rnews` for the command to be executed, `/usr/local/rnews` is executed regardless of the default path. Likewise, `/usr/local/lp` is the `lp` command that is executed.

Including the `ALL` value in the list means that any command from the remote computers specified in the entry will be executed. If you use this value, you give the remote computers full access to your machine.

Caution – This allows far more access than normal users have. You should use this value only when both machines are at the same site, are closely connected, and the users are trusted.

The string:

```
COMMANDS=/usr/local/rnews:ALL:/usr/local/lp
```

illustrates two points:

- The `ALL` value can appear anywhere in the string.
- The path names specified for `rnews` and `lp` are used (instead of the default) if the requested command does not contain the full path names for `rnews` or `lp`.

You should use the `VALIDATE` option whenever you specify potentially dangerous commands like `cat` and `uucp` with the `COMMANDS` option. Any command that reads or writes files is potentially dangerous to local security when executed by the UUCP remote execution daemon (`uuxqt`).

UUCP `VALIDATE` Option

Use the `VALIDATE` option in conjunction with the `COMMANDS` option whenever you specify commands that are potentially dangerous to your machine's security. (`VALIDATE` is merely an added level of security on top of the `COMMANDS` option, though it is a more secure way to open command access than ALL.)

`VALIDATE` provides a certain degree of verification of the caller's identity by cross-checking the host name of a calling machine against the login name it uses. The string:

```
LOGNAME=Uwidget VALIDATE=widget:gadget
```

ensures that if any machine other than `widget` or `gadget` tries to log in as `Uwidget`, the connection is refused. The `VALIDATE` option requires privileged computers to have a unique login and password for UUCP transactions. An important aspect of this validation is that the login and password associated with this entry are protected. If an outsider gets that information, that particular `VALIDATE` option can no longer be considered secure.

Carefully consider which remote computers you will grant privileged logins and passwords for UUCP transactions. Giving a remote computer a special login and password with file access and remote execution capability is like giving anyone on that computer a normal login and password on your computer. Therefore, if you cannot trust someone on the remote computer, do not provide that computer with a privileged login and password.

The `LOGNAME` entry:

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

specifies that if one of the remote computers that claims to be `eagle`, `owl`, or `hawk` logs in on your computer, it must have used the login `uucpfriend`. If an outsider gets the `uucpfriend` login and password, masquerading is easy.

But what does this have to do with the `COMMANDS` option, which appears only in `MACHINE` entries? It links the `MACHINE` entry (and `COMMANDS` option) with a `LOGNAME` entry associated with a privileged login. This link is needed because the execution daemon is not running while the remote computer is logged in. In fact, it is an asynchronous process that does not know which computer sent the execution request. Therefore, the real question is, how does your computer know where the execution files came from?

Each remote computer has its own spool directory on your local machine. These spool directories have write permission given only to the UUCP programs. The execution files from the remote computer are put in its spool directory after being transferred to your computer. When the `uuxqt` daemon runs, it can use the spool directory name to find the `MACHINE` entry in the `Permissions` file and get the `COMMANDS` list. Or, if the computer name does not appear in the `Permissions` file, the default list is used.

This example shows the relationship between the `MACHINE` and `LOGNAME` entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=rmail:/usr/local/rnews \
READ=/ WRITE=/
LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

The value in the `COMMANDS` option means that remote users can execute `rmail` and `/usr/local/rnews`.

In the first entry, you must assume that when you want to call one of the computers listed, you are really calling either `eagle`, `owl`, or `hawk`. Therefore, any files put into one of the `eagle`, `owl`, or `hawk` spool directories is put there by one of those computers. If a remote computer logs in and says that it is one of these three computers, its execution files are also put in the privileged spool directory. You therefore have to validate that the computer has the privileged login `uucpz`.

UUCP `MACHINE` Entry for `OTHER`

You might want to specify different option values for remote machines that are not mentioned in specific `MACHINE` entries. The need might arise when many computers are calling your host, and the command set changes from time to time. The name `OTHER` for the computer name is used for this entry as shown in this example:

```
MACHINE=OTHER \
COMMANDS=rmail:rnews:/usr/local/Photo:/usr/local/xp
```

All other options available for the `MACHINE` entry can also be set for the computers that are not mentioned in other `MACHINE` entries.

Combining `MACHINE` and `LOGNAME` Entries for UUCP

You can combine `MACHINE` and `LOGNAME` entries into a single entry where the common options are the same. For example, the two entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
READ=/ WRITE=/
```

and:

```
LOGNAME=uupz REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

share the same `REQUEST`, `READ` , and `WRITE` options. You can merge them, as shown:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
logname=uucpz SENDFILES-yes \
READ=/ WRITE=/
```

Combining `MACHINE` and `LOGNAME` entries makes the `Permissions` file more manageable and efficient.

UUCP Forwarding

When sending files through a series of machines, the intermediary machines must have the command `uucp` among their `COMMANDS` options. If you type the command:

```
% uucp sample.txt oak\!willow\!pine\!/usr/spool/uucppublic
```

the forwarding operation works only if machine `willow` permits machine `oak` to execute the program `uucp`, and if machine `oak` permits your machine to do the same. The machine `pine`, being the last machine designated, does not have to permit the command `uucp` since it is not doing any forwarding operations. Machines are not normally set up this way.

UUCP `/etc/uucp/Poll` File

The `/etc/uucp/Poll` file contains information for polling remote computers. Each entry in the `Poll` file contains the name of a remote computer to call, followed by a tab character or a space, and finally the hours the computer should be called. The format of entries in the `Poll` file are:

sys-name hour ...

For example, the entry

```
eagle 0 4 8 12 16 20
```

provides polling of computer `eagle` every four hours.

The `uudemon.poll` script processes the `Poll` file but does not actually perform the poll. It merely sets up a polling work file (always named `C.`*file*) in the spool directory. The `uudemon.poll` script starts the scheduler, and the scheduler examines all work files in the spool directory.

UUCP `/etc/uucp/Config` File

The `/etc/uucp/Config` file enables you to override certain parameters manually. Each entry in the `Config` file has this format:

parameter=value

See the `Config` file provided with your system for a complete list of configurable parameter names.

The following `Config` entry sets the default protocol ordering to `Gge` and changes the `G` protocol defaults to 7 windows and 512-byte packets.

```
Protocol=G(7,512)ge
```

UUCP `/etc/uucp/Grades` File

The `/etc/uucp/Grades` file contains the definitions for the job grades that can be used to queue jobs to a remote computer. It also contains the permissions for each job grade. Each entry in this file represents a definition of an administrator-defined job grade that lets users queue jobs.

Each entry in the `Grades` file has the following format:

User-job-grade System-job-grade Job-size Permit-type ID-list

Each entry contains fields that are separated by blank space. The last field in the entry is made up of subfields also separated by spaces. If an entry takes up more than one physical line, you can use a backslash to continue the entry onto the following line. Comment lines begin with a pound sign (#) and occupy the entire line. Blank lines are always ignored.

UUCP User-job-grade Field

This field contains an administrative-defined user job grade name of up to 64 characters.

UUCP System-job-grade Field

This field contains a one-character job grade to which *User-job-grade* is mapped. The valid list of characters is A–Z, a–z, with A having the highest priority and z the lowest.

Relationship Between User and System Job Grades

The user job grade can be bound to more than one system job grade. It is important to note that the `Grades` file is searched sequentially for occurrences of a user job grade.

Therefore, any multiple occurrences of a system job grade should be listed according to the restriction on the maximum job size.

While there is no maximum number for the user job grades, the maximum number of system job grades allowed is 52. The reason is that more than one *User-job-grade* can be mapped to a *System-job-grade*, but each *User-job-grade* must be on a separate line in the file. Here is an example:

```
mail N Any User Any netnews N Any User Any
```

Given this configuration in a `Grades` file, these two *User-job-grade* will share the same *System-job-grade*. Because the permissions for a *Job-grade* are associated with a *User-job-grade* and not a *System-job-grade*, two *User-job-grade*s can share the same *System-job-grade*s and have two different sets of permissions.

Default Grade

You can define the binding of a default *User-job-grade* to a system job grade. You must use the keyword default as user job grade in the *User-job-grade* field of the `Grades` file and the system job grade that it is bound to. The Restrictions and ID fields should be defined as `Any` so that any user and any size job can be queued to this grade. Here is an example:

```
default a Any User Any
```

If you do not define the default user job grade, the built-in default grade `Z` is used. Because the restriction field default is `Any`, multiple occurrences of the default grade are not checked.

UUCP Job-size Field

This field specifies the maximum job size that can be entered in the queue. *Job-size* is measured in bytes and can be a list of the options listed shown in the following table:

TABLE 39–8 Job-size Field

nnnn	Integer specifying the maximum job size for this job grade
n`K`	Decimal number representing the number of kilobytes (`K` is an abbreviation for kilobyte)
n`M`	Decimal number representing the number of megabytes (`M` is an abbreviation for megabyte)
`Any`	Keyword specifying that there is no maximum job size

Here are some examples:

- `5000` represents 5000 bytes
- `10K` represents 10 Kbytes
- `2M` represents 2 Mbytes

UUCP Permit-type Field

This field contains a keyword that denotes how to interpret the ID list. The following table lists the keywords and their meanings.

TABLE 39–9 Permit-type Field

Keyword	ID List Contents
`User`	Login names of users permitted to use this job grade
`Non-user`	Login names of users not permitted to use this job grade
`Group`	Group names whose members are permitted to use this group
`Non-group`	Group names whose members are not permitted to use this job grade

UUCP ID-list Field

This field contains a list of login names or group names that are to be permitted or denied queuing to this job grade. The list of names are separated by a blank space and terminated by a newline character. The keyword `Any` is used to denote that anyone is permitted to queue to this job grade.

Other UUCP Configuration Files

This section describes three less-frequently modified files that impact the use of UUCP facilities.

UUCP /etc/uucp/Devconfig File

The /etc/uucp/Devconfig file enables you to configure devices by service—uucp or cu. Devconfig entries define the STREAMS modules that are used for a particular device. They have the format:

service=*x* device=*y* push=*z*[:*z*...]

x can be cu, uucico, or both separated by a colon. *y* is the name of a network and must match an entry in the Devices file. *z* is replaced by the names of STREAMS modules in the order that they are to be pushed onto the Stream. Different modules and devices can be defined for cu and uucp services.

The following entries are for a STARLAN network and would most commonly be used in the file:

```
service=cu       device=STARLAN     push=ntty:tirdwr
service=uucico   device=STARLAN     push=ntty:tirdwr
```

This example pushes ntty, then tirdwr.

UUCP /etc/uucp/Limits File

The /etc/uucp/Limits file controls the maximum number of simultaneous uucicos, uuxqts, and uuscheds that are running in the uucp networking. In most cases, the default values are fine and no changes are needed. If you want to change them, however, use any text editor.

The format of the Limits file is:

service=*x* max=*y*:

x can be uucico, uuxqt or uusched, and *y* is the limit permitted for that service. The fields can be in any order and in lowercase.

The following entries should most commonly be used in the Limits file:

```
service=uucico max=5
service=uuxqt max=5
service=uusched max=2
```

The example allows five uucicos, five uuxqts, and two uuscheds running on your machine.

UUCP `remote.unknown` File

The other file that affects the use of communication facilities is the `remote.unknown` file. This file is a binary program that executes when a machine not found in any of the `Systems` files starts a conversation. It logs the conversation attempt and drops the connection.

Caution – If you change the permissions of the `remote.unknown` file so it cannot execute, your system accepts connections from any system.

This program executes when a machine that is not in any of the `Systems` starts a conversation. It logs the conversation attempt but fails to make a connection. If you change the permissions of this file so it cannot execute (`chmod 000 remote.unknown`), your system accepts any conversation requests. This is not a trivial change, and you should have good reasons for doing it.

UUCP Administrative Files

The UUCP administrative files are described next. These files are created in spool directories to lock devices, hold temporary data, or keep information about remote transfers or executions.

- *Temporary data files* (`TM`) – These data files are created by UUCP processes under the spool directory `/var/spool/uucp/`*x* when a file is received from another computer. The directory *x* has the same name as the remote computer that is sending the file. The names of the temporary data files have the format:

  `TM.`*pid.ddd*

  where *pid* is a process ID and *ddd* is a sequential three-digit number starting at 0.

  When the entire file is received, the `TM.`*pid.ddd* file is moved to the path name specified in the `C.`*sysnxxx* file (discussed subsequently) that caused the transmission. If processing is abnormally terminated, the `TM.`*pid.ddd* file can remain in the *x* directory. These files should be automatically removed by `uucleanup`.

- *Lock files* (`LCK`) – Lock files are created in the `/var/spool/locks` directory for each device in use. Lock files prevent duplicate conversations and multiple attempts to use the same calling device. The following table shows the different types of UUCP lock files.

TABLE 39–10 UUCP Lock Files

File Name	Description
LCK..*sys*	*sys* represents the name of the computer using the file
LCK.*dev*	*dev* represents the name of a device using the file
LCK.*LOG*	*LOG* represents a locked UUCP log file

These files can remain in the spool directory if the communications link is unexpectedly dropped (usually on computer crashes). The lock file is ignored (removed) after the parent process is no longer active. The lock file contains the process ID of the process that created the lock.

- *Work file* (`C.`) – Work files are created in a spool directory when work (file transfers or remote command executions) has been queued for a remote computer. The names of work files have the format:

  `C.`*sysnxxxx*

  where *sys* is the name of the remote computer, *n* is the ASCII character representing the grade (priority) of the work, and *xxxx* is the four-digit job sequence number assigned by UUCP. Work files contain the following information:

  - Full path name of the file to be sent or requested.
  - Full path name of the destination or user or file name.
  - User login name.
  - List of options.
  - Name of associated data file in the spool directory; if the `uucp -C` or `uuto -p` option was specified, a dummy name (`D.0`) is used
  - Mode bits of the source file.
  - Remote user's login name to be notified on completion of the transfer.

- *Data file* (`D.`) – Data files are created when you specify on the command line to copy the source file to the spool directory. The names of data files have the following format:

  `D.`*systmxxxxyyy* – Where *systm* is the first five characters in the name of the remote computer, *xxxx* is a four-digit job sequence number assigned by `uucp`. The four–digit job sequence number can be followed by a subsequence number, *yyy* that is used when there are several `D.` files created for a work (`C.`) file.

- `X.` *(execute file)* – Execute files are created in the spool directory prior to remote command executions. The names of execute files have the following format:

  `X.`*sysnxxxx*

  sys is the name of the remote computer. *n* is the character representing the grade (priority) of the work. *xxxx* is a four-digit sequence number assigned by UUCP. Execute files contain the following information:

- Requester's login and computer name
- Names of files required for execution
- Input to be used as the standard input to the command string
- Computer and file name to receive standard output from the command execution
- Command string
- Option lines for return status requests

UUCP Error Messages

This section lists the error messages associated with UUCP.

UUCP ASSERT Error Messages

The following table lists ASSERT error messages.

TABLE 39–11 ASSERT Error Messages

Error Message	Description/Action
CAN'T OPEN	An `open()` or `fopen()` failed.
CAN'T WRITE	A `write()`, `fwrite()`, `fprint()`, or similar command, failed.
CAN'T READ	A `read()`, `fgets()`, or similar command failed.
CAN'T CREATE	A `creat()` call failed.
CAN'T ALLOCATE	A dynamic allocation failed.
CAN'T LOCK	An attempt to make a `LCK` (lock) file failed. In some cases, this is a fatal error.
CAN'T STAT	A `stat()` call failed.
CAN'T CHMOD	A `chmod()` call failed.
CAN'T LINK	A `link()` call failed.
CAN'T CHDIR	A `chdir()` call failed.
CAN'T UNLINK	An `unlink()` call failed.
WRONG ROLE	This is an internal logic problem.

TABLE 39–11 ASSERT Error Messages *(Continued)*

Error Message	Description/Action
CAN'T MOVE TO CORRUPTDIR	An attempt to move some bad C. or X. files to the /var/spool/uucp/.Corrupt directory failed. The directory is probably missing or has wrong modes or owner.
CAN'T CLOSE	A close() or fclose() call failed.
FILE EXISTS	The creation of a C. or D. file is attempted, but the file exists. This occurs when a problem arises with the sequence file access. Usually indicates a software error.
NO uucp SERVICE NUMBER	A TCP/IP call is attempted, but no entry is in /etc/services for UUCP.
BAD UID	The user ID is not in the password database. Check name service configuration..
BAD LOGIN_UID	Same as previous.
BAD LINE	A bad line is in the Devices file; there are not enough arguments on one or more lines.
SYSLST OVERFLOW	An internal table in gename.c overflowed. A single job attempted to talk to more than 30 systems.
TOO MANY SAVED C FILES	Same as previous.
RETURN FROM fixline ioctl	An ioctl(2), which should never fail, failed. There is a system driver problem.
BAD SPEED	A bad line speed appears in the Devices or Systems file (Class or Speed field).
BAD OPTION	A bad line or option is in the Permissions file. It must be fixed immediately.
PKCGET READ	The remote machine probably hung up. No action need be taken.
PKXSTART	The remote machine aborted in a nonrecoverable way. This can usually be ignored.
TOO MANY LOCKS	An internal problem has occurred. Contact your system vendor.
XMV ERROR	A problem with some file or directory has occurred. It is likely the spool directory, as the modes of the destinations were supposed to be checked before this process was attempted.
CAN'T FORK	An attempt to make a fork and exec failed. The current job should not be lost but will be attempted later (uuxqt). No action is needed.

UUCP STATUS Error Messages

The following table is a list of the most common STATUS error messages.

TABLE 39–12 UUCP STATUS Messages

Error Message	Description/Action
OK	Status is okay.
NO DEVICES AVAILABLE	Currently no device is available for the call. Check whether a valid device is in the Devices file for the particular system. Check the Systems file for the device to be used to call the system.
WRONG TIME TO CALL	A call was placed to the system at a time other than what is specified in the Systems file.
TALKING	Self-explanatory.
LOGIN FAILED	The login for the given machine failed. It could be a wrong login or password, wrong number, a slow machine, or failure in getting through the Dialer-Token-Pairs script.
CONVERSATION FAILED	The conversation failed after successful startup. This usually means that one side went down, the program aborted, or the line (link) was dropped.
DIAL FAILED	The remote machine never answered. It could be a bad dialer or the wrong phone number.
BAD LOGIN/MACHINE COMBINATION	The machine called with a login/machine name that does not agree with the Permissions file. This could be an attempt to masquerade.
DEVICE LOCKED	The calling device to be used is currently locked and in use by another process.
ASSERT ERROR	An ASSERT error occurred. Check the /var/uucp/.Admin/errors file for the error message and refer to the section "UUCP ASSERT Error Messages" on page 621.
SYSTEM NOT IN Systems FILE	The system is not in the Systems file.
CAN'T ACCESS DEVICE	The device tried does not exist or the modes are wrong. Check the appropriate entries in the Systems and Devices files.
DEVICE FAILED	The device could not be opened.
WRONG MACHINE NAME	The called machine is reporting a different name than expected.
CALLBACK REQUIRED	The called machine requires that it call your machine.
REMOTE HAS A LCK FILE FOR ME	The remote machine has a LCK file for your machine. It could be trying to call your machine. If it has an older version of UUCP, the process that was talking to your machine might have failed, leaving the LCK file. If it has the new version of UUCP and is not communicating with your machine, the process that has a LCK file is hung.
REMOTE DOES NOT KNOW ME	The remote machine does not have the node name of your machine in its Systems file.

TABLE 39–12 UUCP STATUS Messages *(Continued)*

Error Message	Description/Action
`REMOTE REJECT AFTER LOGIN`	The login used by your machine to log in does not agree with what the remote machine was expecting.
`REMOTE REJECT, UNKNOWN MESSAGE`	The remote machine rejected the communication with your machine for an unknown reason. The remote machine might not be running a standard version of UUCP.
`STARTUP FAILED`	Login succeeded, but initial handshake failed.
`CALLER SCRIPT FAILED`	This is usually the same as `DIAL FAILED`. However, if it occurs often, suspect the caller script in the `Dialers` file. Use `Uutry` to check.

UUCP Numerical Error Messages

The following table lists the exit code numbers of error status messages produced by the `/usr/include/sysexits.h` file. Not all are currently used by `uucp`.

TABLE 39–13 UUCP Error Messages by Number

Message Number	Description	Meaning
`64`	Base Value for Error Messages	Error messages begin at this value.
`64`	Command–Line Usage Error	The command was used incorrectly, for example, with the wrong number of arguments, a bad flag, or a bad syntax.
`65`	Data Format Error	The input data was incorrect in some way. This should only be used for user's data and not system files.
`66`	Cannot Open Input	An input file (not a system file) did not exist, or was not readable. This could also include errors like "No message" to a mailer.
`67`	Address Unknown	The user specified did not exist. This might be used for mail addresses or remote logins.
`68`	Host Name Unknown	The host did not exist. This is used in mail addresses or network requests.
`69`	Service Unavailable	A service is unavailable. This can occur if a support program or file does not exist. This message also can be a catchall message when something doesn't work and you don't know why.
`70`	Internal Software Error	An internal software error has been detected. This should be limited to non-operating system related errors if possible.
`71`	System Error	An operating system error has been detected. This is intended to be used for conditions like "cannot fork", "cannot create pipe." For instance, it includes `getuid` returning a user that does not exist in the `passwd` file.

TABLE 39–13 UUCP Error Messages by Number *(Continued)*

Message Number	Description	Meaning
72	Critical OS File Missing	Some system file like `/etc/passwd` or `/var/admin/utmpx` does not exist, cannot be opened, or has some error such as syntax error.
73	Can't Create Output File	A user–specified output file cannot be created.
74	Input/Output Error	An error occurred while doing I/O on some file.
75	Temporary Failure. User is invited to retry	Temporary failure, indicating something that is not really an error. In `sendmail`, this means that a mailer, for example, could not create a connection, and the request should be reattempted later.
76	Remote Error in Protocol	The remote system returned something that was "not possible" during a protocol exchange.
77	Permission Denied	You do not have sufficient permission to perform the operation. This is not intended for file system problems, which should use `NOINPUT` or `CANTCREAT` , but rather for higher level permissions. For example, `kre` uses this to restrict students who can send mail to.
78	Configuration Error	The system detected an error in the configuration.
79	Entry Not Found	Entry not found.
79	Maximum Listed Value	Highest value for error messages.

CHAPTER 40

Working With Remote Systems Topics

This section provides instructions for administering an FTP Server and for accessing remote systems in the Solaris environment. The section contains these chapters.

Chapter 42	Step-by-step instructions for administering the FTP Server.
Chapter 43	Step-by-step instructions for accessing remote files.

CHAPTER 41

Working With Remote Systems (Overview)

This section includes information on working with remote files.

What is a Remote System?

For the purpose of this chapter, a remote system is a workstation or server that is connected to the local system with any type of physical network and configured for TCP/IP communication.

On systems running the Solaris 9 release, TCP/IP configuration is established automatically during startup. For more information, see *System Administration Guide, Volume 3*.

CHAPTER 42

Administering the FTP Server (Tasks)

This chapter includes tasks described in the following table to setup and administer an FTP Server.

TABLE 42–1 Task Map: Administering the FTP Server

Task...	Description	For Instructions, Go To ...
Configure access to the FTP Server	Use the `ftpaccess`, `ftpusers`, and the `ftphosts` files in the `/etc/ftpd` directory to establish or restrict access to the FTP Server.	"How to Set User Login Limits" on page 634 "How to Control the Number of Invalid Login Attempts" on page 635 "How to Disallow FTP Server Access to Particular Users" on page 636 "How to Restrict Access to the Default FTP Server" on page 637 "How to Define FTP Server Classes" on page 633
Set up FTP Server Logins	Establish login accounts for real, guest and anonymous users.	"How to Set Up Real FTP Users" on page 638 "How to Set Up Guest FTP Users" on page 639 "How to Set Up Anonymous FTP Users" on page 640 "How to Create the `/etc/shells` file" on page 640
Customize message files	Edit the `/etc/ftpd/ftpaccess` file to configure the FTP Server to return messages to the FTP client related to specific events.	"How to Customize Message Files" on page 642 "How to Create Messages to be Sent to Users" on page 642 "How to Configure the README Option" on page 643

TABLE 42–1 Task Map: Administering the FTP Server *(Continued)*

Task...	Description	For Instructions, Go To ...
Configure access to files on the FTP Server	Use the `/etc/ftpd/ftpaccess` file to specify classes of users allowed to execute certain commands or to download and upload files to the FTP Server.	"How to Configure DA Discovery for Dial-up Networks" on page 262 "Controlling Uploads and Downloads on the FTP Server" on page 646
Enable limited or complete virtual hosting	Use the `/etc/ftpd/ftpaccess` file to configure the FTP Server to support multiple domains on the same machine.	"How to Enable Limited Virtual Hosting" on page 649 "How to Enable Complete Virtual Hosting" on page 651
Start the FTP Server	Edit the `/etc/inet/inetd.conf` file to start the FTP Server in `nowait` or standalone mode.	"How to Start an FTP Server from `inetd.conf`" on page 653 "How to Start a Standalone FTP Server" on page 653
Shut down the FTP Server	Use the `/etc/ftpd/ftpaccess` file and run the `ftpshut` to shut down the FTP Server.	"Shutting Down the FTP Server" on page 654
Troubleshoot some common FTP Server problems	Check `syslogd` and use `greeting text` and `log commands` to debug problems on the FTP Server.	"How to Check `syslogd` for FTP Server Messages" on page 655 "How to Use `greeting text` to Verify `ftpaccess`" on page 656 "How to Check the Commands Executed by FTP Users" on page 656

Controlling FTP Server Access

You can use the following configuration files in the `/etc/ftpd` directory to control access to the FTP Server.

- `ftpusers` is used to list users who are denied access to the FTP Server.
- `ftphosts` is used to allow or deny login from various hosts to various accounts on the FTP Server.
- `ftpaccess` is the main FTP configuration file. The FTP Server only reads the `/etc/ftpd/ftpaccess` file if called with the `-a` option. When the `ftpaccess` file is used, all users must be members of a class to be allowed access to the FTP

Server. You can specify many `ftpaccess` directives that apply only to a particular class.

For further information, see `ftpusers`(4), `ftphosts`(4), and `ftpaccess`(4)

Note – In all FTP Server configuration files, lines beginning with # signs are treated as comments.

▼ How to Define FTP Server Classes

To login to the FTP Server, users must be members of a class when the `ftpaccess` file is used. To add the `class` directive to the `ftpaccess` file, you specify the *class* name, *typelist* of users permitted access from a particular host.

1. **Become superuser.**
2. **Add entries for anonymous, guest, and real users in the `ftpaccess` file.**

   `class` *class typelist addrglob* [*addrglob...*]

`class`	Keyword used to define FTP users.
class	A name defined by the `class` keyword. Each login is compared against a list of defined classes. The logged in user is considered a member of the first class matched.
typelist	A comma-separated list of the keywords that match the three types of users: `anonymous`, `guest`, and `real`.
addrglob	A globbed domain name or a globbed numeric address. The *addrglob* can also be the name of a file, starting with a slash ('/'), which contains additional address globs: `address:netmask` or `address/cidr`. Here are some examples of globbed addresses: ■ numeric IPv4 address: **`10.1.2.3`** ■ globbed domain name **`*.provider.com`** ■ globbed numeric IPv4 address **`10.1.2.*`** ■ numeric IPv4 address:netmask **`10.1.2.0:255.255.255.0`** ■ numeric IPv4 address/CIDR **`10.1.2.0/24`** ■ numeric IPv6 address: **`2000::56:789:21ff:fe8f:ba98`** ■ numeric IPv6 address/CIDR: **`2000::56:789:21ff:fe8f:ba98/120`**

Example—Defining FTP Server Classes

```
class  local  real,guest,anonymous *.provider.com
class  remote real,guest,anonymous *
```

The previous example defines the `local` class as any user of the type `real`, `guest`, or `anonymous` who logs in from `*.provider.com`. The last line defines `remote` as any user that logs in from anywhere other than `*.provider.com`.

▼ How to Set User Login Limits

You can limit the number of simultaneous logins by users of a certain class with directives set in the `ftpaccess` file. Each login limit contains the name of a class, a UUCP-style days-of-week list, and a message file to display if the limit is exceeded.

To set user login limits, follow the steps in the next procedure.

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file:**

   `limit` *class n times* [*message_file*]

`limit`	Keyword used to restrict simultaneous logins by the specified number of users of a defined class at certain connection times.
class	A name defined by the `class` keyword. Each login is compared against a list of defined classes. The logged in user is considered a member of the first class matched.
n	Number of users.
times	Day-of-week and time-of-day when the class can connect. Use `Any` for any day.
message_file	Message file that is displayed if a user is denied access.

Example—Setting User Login Limits

```
limit   anon     50  Wk0800-1800        /etc/ftpd/ftpmsg.deny
limit   anon    100  Any                 /etc/ftpd/ftpmsg.deny
limit   guest   100  Any                 /etc/ftpd/ftpmsg.deny
```

The first line of the preceding example shows a limit of 50 simultaneous logins allowed to users of class `anon` during weekly work hours. The second line limits `anon`

users to 100 simultaneous logins outside of working hours. The last line shows a limit of 100 `guest` logins allowed at any time. For information on how to specify day and time parameters, see `ftpaccess`(4).

The example further indicates that the content of the file `/etc/ftpd/ftpmsg.deny` is returned when a specified login limit is reached, assuming `ftpmsg.deny` exists. For information on using the `/usr/sbin/ftpcount` command to view the number and login limit for each class of user logged in at a given time, see `ftpcount`(1).

Users are allowed login to the FTP Server unless a specified limit is reached. Anonymous users are logged in as the user `ftp`. Real users are logged in as themselves, and guests are logged in as real users with a `chroot` environment to limit access privileges.

For information on using the `/usr/sbin/ftpwho` command to check the identities of the users logged into the FTP Server, see `ftpwho`(1).

▼ How to Control the Number of Invalid Login Attempts

If a login to the FTP Server fails because of a problem such as misspelling required information, login is usually repeated. The user is allowed a specific number of consecutive login attempts before a message is logged to the `syslog` file. At that point, the user is disconnected. You can set a failure limit on the number of login attempts by following steps in the next procedure.

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file.**

```
loginfails n
```

loginfails	Keyword used to assign the number of login failures permitted before the FTP connection is terminated.
n	Number of times a login can fail.

Example—Controlling the Number of Invalid Login Attempts

```
loginfails 10
```

The preceding example states that the user is disconnected from the FTP Server after ten failed login attempts.

▼ How to Disallow FTP Server Access to Particular Users

The `/etc/ftpd/ftpusers` file lists names of users who are not allowed to log in to the FTP Server. When login is attempted, the FTP Server checks the `/etc/ftpd/ftpusers` file to determine whether the user should be denied access. If the user's name is not found in that file, the server then searches the `/etc/ftpusers` file.

If the user's name is matched in `/etc/ftpusers`, a `syslogd` message is written stating that the match was found in a deprecated file. The message also recommends using `/etc/ftpd/ftpusers` instead of `/etc/ftpusers`.

Note – Support for the `/etc/ftpusers` file has been deprecated in this release. If the `/etc/ftpusers` file exists when the FTP Server is installed, the file is moved to `/etc/ftpd/ftpusers`.

For additional information, see`syslogd`(1M), `in.ftpd`(1M), and `ftpusers`(4)

1. **Become superuser.**
2. **Add entries to the `/etc/ftpd/ftpusers` file for users not allowed to login to the FTP Server.**

Example—How to Disallow FTP Server Access

```
root
daemon
bin
sys
adm
lp
uccp
nuucp
listen
nobody
noaccess
nobody4
```

The previous example lists the typical entries in the `ftpusers` file. User names match entries in the `/etc/passwd`. The list generally includes the superuser root and other administrative and system application identities.

The root entry is included in the `ftpusers` file as a security measure. The default security policy is to disallow remote logins for root. The policy is also followed for the default value set as the CONSOLE entry in the `/etc/default/loginfile`. See `login`(1).

▼ How to Restrict Access to the Default FTP Server

In addition to the controls mentioned previously, you can add explicit statements to the `ftpaccess` file to restrict access to the FTP Server.

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file.**

   a. **By default, all users are allowed access to the default (non-virtual) FTP Server. To deny access for specific users (other than `anonymous`), add the entry:**

   ```
   defaultserver deny username [username...]
   ```

`defaultserver`	Keyword used to identify the non-virtual server to which access can be denied or allowed.
username	Login name of a user with restricted access to the `defaultserver`.

   b. **To allow access for users not listed on the `deny` line:**

   ```
   defaultserver allow username [username...]
   ```

   c. **To prevent access by anonymous users, add the entry:**

   ```
   defaultserver private
   ```

Example—Restricting Access to the Default FTP Server

```
defaultserver deny *
defaultserver allow username
```

The previous example states that the FTP Server denies access to all except `anon` users and those listed on the `allow` line.

You can also use the `ftphosts` file to deny access to particular login accounts from various hosts. See `ftphosts`(4) for additional information.

Setting Up FTP Server Logins

To access an FTP Server, you must first log in. The FTP Server supports three types of user login accounts for *real*, *guests*, and *anonymous* users.

- *Real* users have accounts that allow them to establish terminal sessions on systems running the FTP Server. Subject to directory and file access permissions, the entire disk structure is visible to real users.
- *Guest* users also need accounts to log in to the FTP Server. Each guest account is set up with a username and password. Functioning login shells are not assigned to guests to prevent users from establishing terminal sessions. At login, the FTP Server performs a `chroot`(2) operation to restrict a guest's view of the server's disk structure.

Note – Login shells for real and guest users must be listed in the `/etc/shells` file to allow access to the FTP Server.

- *Anonymous* users login to the FTP Server using the either `ftp` or `anonymous` as a username. By convention, anonymous users supply an email address when prompted for a password.

  At login, the FTP Server performs a `chroot`(2) operation that restricts the anonymous user's view of the server's disk structure. A single file area is shared by all anonymous users, unlike the separate areas which can be created for each guest user.

  Real and guest users login using individual accounts with passwords known only to one person. Anonymous users log in to a well known account which is potentially available to anyone. Most large-scale file distribution is made using the anonymous account.

▼ How to Set Up Real FTP Users

To enable access for real users to the FTP Server:

1. **Verify that the user has an account set up with a username and password that can be used to establish a terminal session.**

   For more information, see "Managing Use Accounts and Groups (Overview)" in the *System Administration Guide, Volume 1*.

2. **Confirm that the real user is a member of a class in the `ftpaccess` file.**

   For information on the user classes defined in the `ftpaccess` file, see "How to Define FTP Server Classes" on page 633.

3. **Verify that the user's login shell is listed in the `/etc/shells` file.**

▼ How to Set Up Guest FTP Users

The `ftpconfig` script is used to copy all necessary system files to the home directory. When the guest user and the guest's home directory already exist, the `ftpconfig` script updates the area with the current system files.

For more information, see `ftpconfig`(1M)

Note – Unlike the user name (`anonymous` or `ftp`) set for anonymous users, user names for FTP guests are not fixed. Any name that would work as a real user name can be selected.

To enable access by a guest user to the FTP Server:

1. **Use the `useradd` script to create a guest user account with a login shell of `/bin/true` and a home directory of *`/root_dir/./home_dir`*.**

   For more information, see `useradd`(1M) and "Managing Use Accounts and Groups (Overview)" in the *System Administration Guide, Volume 1*.

Note – In this procedure, **`/home/guests/./guest1`** is used as the home directory name for a user called `guest1`.

```
# /usr/sbin/useradd -m -c "Guest FTP" -d \
  /home/guests/./guest1 -s /bin/true guest1
```

2. **Assign a password to the guest account.**

3. **Add a `guestuser` entry to the `ftpaccess` file.**

```
guestuser guest1
```

Note – You can also use the `guestgroup` capability in the `ftpaccess` file to specify guest users. The `guest-root` capability in `ftpaccess` eliminates the need for the `/./` in the guest users home directory path.

---

4. **Confirm that the guest user is a member of a `class` in the `ftpaccess` file. See "How to Define FTP Server Classes" on page 633 for further information.**

5. **Use the `ftpconfig` script to create the required files in the `chroot` area.**

   ```
   /usr/sbin/ftpconfig -d /home/guests
   ```

6. **Confirm that `/bin/true` listed in the `/etc/shells` file. See "How to Create the `/etc/shells` file" on page 640.**

Example—Setting Up a Guest FTP Server

In this example, the FTP area is set up in the `/home/guests` directory.

```
# /usr/sbin/ftpconfig -d /home/guests
Updating directory /home/guests
```

▼ How to Set Up Anonymous FTP Users

The `ftpconfig` script creates the `anonymous` user account and populates the home directory with the required files.

For more information, see `ftpconfig`(1M).

To enable access by an anonymous user to the FTP Server:

1. **Use the `ftpconfig` script to create the anonymous user account.**

   ```
   # /usr/sbin/ftpconfig anonymous-ftp-directory
   ```

2. **Confirm that the anonymous user is assigned to a `class` in the `ftpaccess` file. See "How to Define FTP Server Classes" on page 633 for further information.**

Example—Setting Up Anonymous FTP Users

In this example, the FTP area is set up in the `/home/ftp` directory.

```
# /usr/sbin/ftpconfig /home/ftp
Creating user ftp
Updating directory /home/ftp
```

▼ How to Create the `/etc/shells` file

1. **Become superuser.**
2. **Create the `/etc/shells` file.**
3. **Edit `/etc/shells`. Add the full path to each shell on a single line.**

Example—Creating the /etc/shells file

The following is an example of an `/etc/shells` file with a `/bin/true` listed for FTP guest users:

```
/sbin/sh
/bin/csh
/bin/jsh
/bin/ksh
/bin/remsh
/bin/rksh
/bin/rsh
/bin/sh
/usr/bin/csh
/usr/bin/ksh
/usr/bin/bash
/usr/bin/tcsh
/usr/bin/zsh
/bin/true
```

Customizing Message Files

You can configure the FTP Server to return messages related to specific events to the FTP client. A welcome message might be set to display when a user logs in to the FTP Server. Another message could appear when the user makes a directory change.

In addition to plain text, message files can contain one or more *magic cookies*. A magic cookie is composed of a `%` (percent sign) followed by a single character. When you embed a cookie in message text, information associated with the cookie appears on screen at the point the message file is called.

For example, message text might contain the cookie `%L`:

```
Welcome to %L!
```

When the message is displayed, the magic cookie `%L` is replaced with the name of the server as defined by the `hostname` statement in the `ftpaccess` file. For a complete list of supported message cookies, see `ftpaccess`(4).

Note – If the hostname is not defined in the `ftpaccess` file, the default hostname for the local machine is used.

▼ How to Customize Message Files

1. **Become superuser.**
2. **Edit your message file to include magic cookies as appropriate.**
   See `ftpaccess`(4) for a list of cookies you can use.

Example—Customizing Message Files

The following is an example of a message file includes magic cookies:

```
Welcome to %L -- local time is %T.

You are number %N out of a maximum of %M.
All transfers are logged.

If your FTP client crashes or hangs shortly after login
please try
using a dash (-) as the first character of your password.
This will
turn off the informational messages that may be confusing
your FTP
client.

Please send any comments to %E.
```

▼ How to Create Messages to be Sent to Users

After the user is logged in, system or application related messages are displayed on screen. The `ftpaccess` file lists the events that trigger associated `message` statements.

1. **Become superuser.**

2. **Add the following entries to the `ftpaccess` file:**

```
message message_file [when [class ...]]
```

`message`	Keyword used to specify the message file to be displayed when a user logs in or executes the command to change the working directory.
message_file	Name of the message file to be displayed.
when	Parameter set as `login` or `cwd=`*dir*. See the following example.
class	The `class` specification allows the message to be displayed only to members of a particular class.

Example: How to Create Messages to be Sent to Users

```
message    /etc/ftpd/Welcome    login   anon guest
message    .message    cwd=*
```

The preceding example states that the file `/etc/ftpd/Welcome` is displayed at login for users of the class `anon` or `guest`. The second line states that the `.message` file in the current working directory is displayed for all users.

Message files are created relative to the `chroot` directory for guest and anonymous users.

▼ How to Configure the README Option

The first time a directory is visited, `README` files can be listed. To configure the README option, add the following entries to the `ftpaccess` file.

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file.**

```
readme message_file [when [class...]]
```

`readme`	Keyword used to specify a message file to be checked when a user logs in or changes the working directory. If the message file exists, the user is notified and is given the date the file was modified.

message_file	Name of the message file to be checked.
when	Parameter set as `login` or `cwd=`*dir*. See the following example.
class	The `class` specification allows the message to be displayed only to members of a particular class.

Note – The `greeting` and `banner` keywords can also be used to output messages to users. See `ftpaccess`(4).

Example—Configuring the README Option

```
readme  README*     login
readme  README*     cwd=*
```

The previous example states that any files which match `README*` are listed at login or when a directory is changed. Here is a sample login that is based on the settings used in that example:

```
% ftp earth
Connected to earth.
220 earth FTP server ready.
Name (earth:rimmer): ftp
331 Guest login ok, send your complete e-mail address as password.
Password:
230-
230-Welcome to earth -- local time is Thu Jul 15 16:13:24
1999.
230-
230-You are number 1 out of a maximum of 10.
230-All transfers are logged.
230-
230-If your FTP client crashes or hangs shortly after login
please try
230-using a dash (-) as the first character of your
password.  This will
230-turn off the informational messages that may be
confusing your FTP
230-client.
230-
230-Please send any comments to ftpadmin@earth.
230-
230 Guest login ok, access restrictions apply.
ftp> cd pub
250-Please read the file README
250-  it was last modified on Thu Jul 15 16:12:25 1999 - 0
days ago
```

```
250 CWD command successful.
ftp> get README /tmp/README
200 PORT command successful.
150 Opening ASCII mode data connection for README (0
bytes).
226 ASCII Transfer complete.
ftp> quit
221 Goodbye.
```

Controlling Access to Files on the FTP Server

The FTP Server access controls in this section supplement the standard file and directory access controls available with Solaris 9. Use the standard Solaris commands to restrict who can access, change or upload files. See `chmod`(1), `chown`(1), and `chgrp`(1).

▼ How to Control File Access Commands

To use the permission capabilities in `ftpaccess` to specify what type of user is allowed to perform which commands:

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` :**

command `yes|no` *typelist*

command	The commands `chmod`, `delete`, `overwrite`, `rename`, or `umask`.
`yes|no`	Allows or disallows a user to issue a command.
typelist	A comma-separated list of any of the keywords "anonymous," "guest," and "real."

Example—How to Control File Access Commands

The following are examples of permissions set for file access functions on FTP Server.

```
chmod no anonymous, guest
delete    no anonymous
overwrite no anonymous
rename    no anonymous
umask     no guest, anonymous
```

The preceding example states:

- Anonymous users are not allowed to delete, overwrite, or rename files.
- Guests and anonymous users are both prevented from changing access modes and resetting the umask.

Controlling Uploads and Downloads on the FTP Server

You can control uploads and downloads made to and from the FTP Server by setting permissions on directories on the server. By default, uploads are not allowed for anonymous users. Be very careful when enabling anonymous uploads.

▼ How to Control Uploads to the FTP Server

Add the directives to the `ftpaccess` file to specify upload permissions and error messages for upload failures.

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file.**

   To enable users to upload files, add the entry:

   ```
   upload [absolute|relative] [class=<classname>]... [-] root-dir \
   dirglob yes|no owner group mode [dirs|nodirs] [<d_mode>]

   path-filter typelist mesg allowed_charset {disallowed regexp...}
   ```

`upload`	Keyword applied to users who have a home directory (the argument to `chroot()`) of the *root-dir* . The *root-dir* may be specified as "*" to match any home directory.

`absolute|relative`	Parameter that specifies whether the *root-dir* directory paths are interpreted as absolute or relative to the current `chroot` directory.
`class`	Keyword used to specify any number of `class=<classname>` restrictions. If restrictions are specified, the upload clause only takes effect if the current user is a member of one of the specified classes.
root-dir	User's root directory and the home directory for anonymous users.
dirglob	A pattern to match a directory name. An asterisk can be used in any place or alone to signify any directory.
`yes|no`	Variable that allows or disallows upload to the FTP Server.
owner	Owner of files uploaded into `dirnames`.
group	Group associated with files uploaded into `dirnames`.
mode	Parameter used to specify access permissions for uploaded files. The default mode `0440` prevents the anonymous account from reading uploaded files.
`dirs|nodirs`	Keyword that allows or disallows users to create subdirectories in a directory listed in `dirnames`.
`d_mode`	Optional mode that determines the permissions for a newly created directory
`path-filter`	Keyword that controls the names of uploaded files.
typelist	A comma-separated list of any of the keywords "anonymous," "guest," and "real."
mesg	Message file that is displayed fails to match the regexp criteria.
allowed_charset {disallowed regexp...}	Alphanumeric characters allowed or disallowed in filenames.

Example—Controlling Uploads to the FTP Server

```
upload /export/home/ftp /incoming yes ftpadm ftpadmin 0440 nodirs
path-filter anonymous /etc/ftpd/filename.msg ^[-A-Za-z0-9._]*$ ^[.-]
```

The preceding example states that:

- FTP user accounts that `chroot` to `/export/home/ftp` can upload to the `/incoming` directory. Uploaded files are owned by user `ftpadm` and the group `ftpadmin`. Mode is set to `0440` with the `nodirs` keyword to prevent anonymous users from creating subdirectories.
- For anonymous users, a file name is any sequence of A-Z, a-z, 0-9, . (dot), - (dash), or _ (underline). File names may not start with a . (dot) or - (dash). If a file name fails this filter, the `/etc/ftpd/filename.msg` message displays if the FTP Administrator has created the message file. This message is followed by an FTP Server error message.

Note – Ownership and permissions on a directory into which anonymous uploads are allowed should be tightly controlled. The FTP Administrator should be the owner of all files uploaded to the FTP Server. You need to create an FTP Administrator when anonymous users are allowed to upload files. The directory should be owned by the user `ftpadm` and group `ftpadm` with permissions set to `3773`.

The access mode for files uploaded to the FTP Server should be `0440`. The `0440` mode prevents the anonymous account from reading uploaded files. This restriction protects your server from becoming a staging area for third party file distribution.

To make uploaded files available for distribution, the FTP Administrator can move files to a public directory.

▼ How to Control Downloads to the FTP Server

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file to prevent users from retrieving files.**

```
noretrieve [absolute|relative] [class=classname]... [-] filename ...
```

`noretrieve`	Keyword used to deny retrieval of a particular file or files.
`absolute|relative`	Parameter that specifies whether the *root-dir* directory paths are interpreted as absolute or relative to the current `chroot` directory.
`class`	Keyword used to specify `class=<classname>` of users to which `noretrieve` restrictions apply.

filename	Name of file the user is not permitted to retrieve.

Example—Controlling Downloads to the FTP Server

```
noretrieve /etc/passwd
```

The preceding example states that all users are prevented from retrieving the `/etc/passwd` file.

Virtual Hosting

Virtual hosting allows the FTP Server to support multiple domains on the same machine. Each virtual host requires a separate logical interface and IP address.

The FTP Server supports two types of virtual hosting: *limited* and *complete*. With limited virtual hosting, the same configuration files are used for all virtual hosts. With complete virtual hosting, separate configuration files can be used for each virtual host.

Note – By default, real and guest users are not allowed to log in to virtual hosts. You can set the following `ftpaccess` directives to override the default.

```
To allow access to specific users:
virtual address allow username
To deny access to anonymous users:
virtual address private username
```

See `ftpaccess`(4) for further information.

▼ How to Enable Limited Virtual Hosting

Limited virtual hosting provides partial support for virtual FTP Servers. You can enable support for limited virtual hosting by specifying the virtual root directory. If required, you can also set the following parameters for the virtual host in the `ftpaccess` file:

- `banner`
- `logfile`
- `email`
- `hostname`

All directives in the `ftpaccess` file are shared globally across all virtual servers.

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file.**

```
virtual address root|banner|logfile path
virtual address hostname|email string
```

`virtual`	Keyword used to enable virtual server capabilities.
address	IP address of the virtual server
`root`	The root directory of the virtual server
`banner`	Banner file displayed when a connection is made to the virtual server
`logfile`	Record of file transfers made to and from the virtual server
path	Variable used to specify the location of directories and files on the virtual server
`email`	Email address used in message files and in the `HELP` command.
`hostname`	Name of the host shown in the greeting message or status command
string	Variable used to specify `email` or `hostname` parameters

Note – While it is possible to use `hostname` as the *address* of the virtual server, you are strongly encouraged use the IPv4 address instead. DNS must be available when the FTP connection is received in order for `hostname` to be matched. For an IPv6 host, use the host name rather than the IPv6 address.

Example—Enabling Limited Virtual Hosting

```
virtual 10.1.2.3 root    /var/ftp/virtual/ftp-serv
virtual 10.1.2.3 banner  /var/ftp/virtual/ftp-serv/banner.msg
virtual 10.1.2.3 logfile /var/log/ftp/virtual/ftp-serv/xferlog
```

The preceding example sets the location of the `root` directory, `banner`, and `logfile` on a virtual FTP Server.

Note – The `ftpaddhost`(1M) script with the `-l` option is provided to configure limited virtual hosts.

In the following example, `ftpaddhost` run with `-l -b -x` options configures limited virtual hosting with a test banner and the logfile `/var/ftp/virtual/10.1.2.3/xferlog` under a virtual root `/var/ftp/virtual/10.1.2.3`.

```
ftpaddhost -l -b -x /var/ftp/virtual/10.1.2.3/xferlog \
/var/ftp/virtual/10.1.2.3
```

▼ How to Enable Complete Virtual Hosting

Complete virtual hosting allows separate configuration files for each virtual domain. To enable complete support for virtual hosting on the FTP Server, you can create or modify the following FTP configuration files for specific domains:

- `ftpaccess`
- `ftpusers`
- `ftpgroups`
- `ftphosts`
- `ftpconversions`

For further information, see `ftpaccess`(4), `ftpusers`(4), `ftpgroups`(4), `ftphosts`(4), and `ftpconversions`(4).

Note – If separate versions of the configuration files are unavailable, master versions of the files in the `/etc/ftpd` directory are used.

1. **Become superuser.**
2. **Add the following entry to the `/etc/ftpd/ftpservers` file.**

   address */config-file-dir*

address	IP address of the virtual server.
config-file-dir	Directory that contains the configuration files customized for the virtual host.

Note – While it is possible to use `hostname` as the *address* of the virtual server, you are strongly encouraged use the IPv4 address instead. DNS must be available when the FTP connection is received in order for `hostname` to be matched. For an IPv6 host, use the host name rather than the IPv6 address.

3. **To create a customized version of an FTP Server configuration file for the virtual host, copy the master version of the file from `/etc/ftpd` to the `/config-file-dir` directory.**

   For further information, see `ftpservers`(4).

Example—Enabling Complete Virtual Hosting

```
#
# FTP Server virtual hosting configuration file
#

10.1.2.3 /net/inet/virtual/somedomain/
10.1.2.4 /net/inet/virtual/anotherdomain/
```

The preceding example specifies the IP addresses for two different domains on the virtual server.

Note – The `ftpaddhost`(1M) script with the `-c` option is provided to configure complete virtual hosts.

In the following example, `ftpaddhost` run with `-l -b -x` options configures limited virtual hosting with a test banner and the logfile `/var/ftp/virtual/10.1.2.3/xferlog` under a virtual root `/var/ftp/virtual/10.1.2.3`.

```
ftpaddhost -l -b -x /var/ftp/virtual/10.1.2.3/xferlog \
/var/ftp/virtual/10.1.2.3
```

Starting the FTP Server Automatically

The FTP Server can be started in one of two ways:

- As a `nowait` server started from the `inetd.conf` file
- As a standalone server started from the command line or by a startup script

Starting an FTP Server from `inetd.conf`

You can add a `nowait` entry in `inetd.conf` file to start the FTP Server. If the site handles many connections, the FTP daemon can also be run in standalone mode. For more information, see `inetd.conf`(4). See also `in.ftpd`(1M) for information on additional command line options.

▼ How to Start an FTP Server from `inetd.conf`

1. **Become superuser.**
2. **Add a `nowait` entry to the `inetd.conf` file:**

```
ftp  stream  tcp6  nowait  root  /usr/sbin/in.ftpd  in.ftpd -a
```

Note – The `-a` option is specified to enable the use of the `ftpaccess` file.

---

3. **Signal `inetd` to reread the `inetd.conf` file.**

```
# pkill -HUP inetd
```

Starting a Standalone FTP Server

The FTP Server can also be run independently of the `inetd.conf` as a *standalone* server.

A standalone server always has the quickest possible response time, and is intended for large servers dedicated to providing FTP service. The standalone server provides low connection latency for dedicated servers because the standalone system never has to be restarted. The standalone server is always running—even during off-peak hours—waiting indefinitely for connections.

▼ How to Start a Standalone FTP Server

1. **Become superuser.**
2. **Add a # sign at the start of the ftp service line in the `inetd.conf` file to comment out the entry.**

3. **Signal `inetd` to reread the `inetd.conf` file.**

```
# pkill -HUP inetd
```

4. **Start the standalone FTP Server.**

```
# /usr/sbin/in.ftpd -a -S
```

Add the line to an FTP Server startup script. See "Run Control Scripts" in *System Administration Guide, Volume 1* for information on creating a system startup script.

Shutting Down the FTP Server

The `ftpshut`(1M) command closes down the FTP Server at a given time.

When you run `ftpshut`, a file is generated from command line options that specify when shutdown will occur, the point at which new connections are refused, and when existing connections are dropped. Users are notified of a server shutdown based on this information. The location of the file created by `ftpshut` is specified by the `shutdown` directive in the `ftpaccess` file.

▼ How to Shut Down the FTP Server

Follow the steps in this procedure to run `ftpshut` and to add the `shutdown` directive to the `ftpaccess` file.

1. **Become superuser.**
2. **Add the following entries to the `ftpaccess` file.**

`shutdown` *path*

`shutdown`	Keyword used to specify the *path* to a file that is checked regularly to see if the FTP Server scheduled to be shut down.
path	Location of the file created by `ftpshut` command

3. **Run the `ftpshut` command.**

```
ftpshut [ -V ] [ -l min] [ -d min] time [warning-message...]
```

`ftpshut`	Command that provides a procedure for notifying users that the FTP Server is shutting down.
`-V`	Option specified to display copyright and version information, then terminate
`-l`	Flag used to adjust the time that new connections to the FTP Server are denied
`-d`	Flag used to adjust the time that existing connections to the FTP Server are disconnected.
`time`	Shutdown time specified by the word `now` for immediate shutdown; or in one of two formats (+ *number* or *HHMM*) for a future shutdown.
`[warning-message...]`	Shutdown notification message.

4. **Use the `ftprestart` command to restart the FTP Server after shutdown.**
   For further information, see `ftpshut`(1M), `ftpaccess`(4), and `ftprestart`(1M).

Debugging the FTP Server

This section describes some of the ways to debug problems with the FTP Server.

▼ How to Check `syslogd` for FTP Server Messages

The FTP Server writes messages which are useful for debugging to the location specified for daemon messages in the `/etc/syslog.conf` file. In the case of a problem with the FTP Server, the first thing to do is to check this file for such messages.

The FTP Server messages are controlled by facility daemon and level information. To send messages from the FTP Server to `/var/adm/message` and have `syslogd` reread its configuration file:

1. **Add an entry such as the following to the** `/etc/syslog.conf` **file.**

   ```
   daemon.info /var/adm/message
   ```

2. **Signal** `syslogd` **to reread its configuration file.**

   ```
   # pkill -HUP syslogd
   ```

   This action causes informational messages from the FTP Sever to be written to `/var/adm/messages`.

▼ How to Use `greeting text` to Verify `ftpaccess`

To use the `greeting text` capability to check that the correct `ftpaccess` file is being used:

1. **Add the following directive to the ftpaccess file.**

   ```
   greeting text message
   ```

2. **Connect to the FTP Server.**

3. **If the message fails to appear:**

   a. **Confirm that the** `ftpaccess` **file is in the correct location. Use the** `strings`**(1) command to get the location of the file from the FTP Server binary.**

      ```
      # strings /usr/sbin/in.ftpd | grep "^/.*ftpaccess"
      ```

   b. **Check the** `ftpservers` **file to see if virtual hosting has been configured.**

      For further information, see `ftpaccess`(4), `ftpserver`(4), `strings`(1), `syslog.conf`(4), and `pkill`(1).

▼ How to Check the Commands Executed by FTP Users

To see what commands are being executed by FTP users, use the `log commands` logging capability in `ftpaccess`.

1. **Add the following directive to the** `ftpaccess` **file to log individual commands by users specified in** *typelist***.**

   ```
   log commands typelist
   ```

2. **Check messages written to** `/etc/syslogd.conf`**.**

CHAPTER 43

Accessing Remote Systems (Tasks)

This chapter describes all the tasks required to log in to remote systems and work with their files. This is a list of the step-by-step instructions in this chapter.

This chapter provides tasks described in the following table to log in and copy files from remote systems.

TABLE 43–1 Task Map: Accessing Remote Systems

Task...	Description	For Instructions, Go To ...
Log in to a remote system (`rlogin`)	■ Remove `.rhosts` files. ■ Use the `rlogin` command to access a remote system.	"How to Search for and Remove `.rhosts` Files" on page 662 "How to Find Out If a Remote System Is Operating" on page 663 "How to Find Who Is Logged In to a Remote System" on page 664 "How to Log In to a Remote System (`rlogin`) " on page 665 "How to Log Out From a Remote System (`exit`)" on page 665

TABLE 43–1 Task Map: Accessing Remote Systems *(Continued)*

Task...	Description	For Instructions, Go To ...
Log in to a remote system (`ftp`)	■ Open and close an`ftp` connection ■ Copy files to and from a remote system.	"How to Open an `ftp` Connection to a Remote System" on page 667 "How to Close an `ftp` Connection to a Remote System" on page 668 "How to Copy Files From a Remote System (`ftp`)" on page 669 "How to Copy Files to a Remote System (`ftp`)" on page 671
Copy remote files with `rcp`	Use the `rcp` command to copy files to and from a remote system.	"How to Copy Files Between a Local and a Remote System (`rcp`)" on page 675

Logging In to a Remote System (`rlogin`)

The `rlogin` command enables you to log in to a remote system. Once logged in, you can navigate through the remote file system and manipulate its contents (subject to authorization), copy files, or execute remote commands.

If the system you are logging into is in a remote domain, be sure to append the domain name to the system name. In this example, `SOLAR` is the name of the remote domain:

```
rlogin pluto.SOLAR
```

Also, you can interrupt a remote login operation at any time by typing Control-d.

Authentication for Remote Logins (`rlogin`)

Authentication (establishing who you are) for `rlogin` operations can be performed either by the remote system or by the network environment.

The main difference between these forms of authentication lies in the type of interaction they require from you and the way they are established. If a remote system tries to authenticate you, you will be prompted for a password, unless you set up the `/etc/hosts.equiv` or `.rhosts` file. If the network tries to authenticate you, you won't be asked for a password, since the network already knows who you are.

When the remote system attempts to authenticate you, it relies on information in its local files; specifically if:

- Your system name and user name appears in the remote system's `/etc/hosts.equiv` file, or
- Your system name and user name appears in the remote user's `.rhosts` file, under the remote user's home directory

Network authentication relies on one of these two methods:

- A "trusting network environment" that has been set up with your local network information service and the automounter
- One of the network information services pointed to by the remote system's `/etc/nsswitch.conf` file contains information about you

---

Note – Network authentication generally supersedes system authentication.

The `/etc/hosts.equiv` File

The `/etc/hosts.equiv` file contains a list of trusted hosts for a remote system, one per line. If a user attempts to log in remotely (using `rlogin`) from one of the hosts listed in this file, and if the remote system can access the user's password entry, the remote system allows the user to log in without a password.

A typical `hosts.equiv` file has the following structure:

```
host1
host2 user_a
+@group1
-@group2
```

When a simple entry for a host is made in `hosts.equiv`, such as the entry above for `host1`, it means that the host is trusted, and so is any user at that machine.

If the user name is also mentioned, as in the second entry in the example, then the host is trusted only if the specified user is attempting access.

A group name preceded by a plus sign (+) means that all the machines in that netgroup are considered trusted.

A group name preceded by a minus sign (–) means that none of the machines in that netgroup are considered trusted.

Security Risks When Using the `/etc/hosts.equiv` *File*

The `/etc/hosts.equiv` file presents a security risk. If you maintain a `/etc/hosts.equiv` file on your system, you should include only trusted hosts in your network. The file should not include any host that belongs to a different network, or any machines that are in public areas. (For example, do not include a host that is located in a terminal room.)

This can create a serious security problem. Either replace the `/etc/hosts.equiv` file with a correctly configured one, or remove the file altogether.

A single line of + in the `/etc/hosts.equiv` file indicates that every known host is trusted.

The `.rhosts` File

The `.rhosts` file is the user equivalent of the `/etc/hosts.equiv` file. It contains a list of host-user combinations, rather than hosts in general. If a host-user combination is listed in this file, the specified user is granted permission to log in remotely from the specified host without having to supply a password.

Note that a `.rhosts` file must reside at the top level of a user's home directory. `.rhost` files located in subdirectories are not consulted.

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

Security Risks When Using the `.rhosts` *File*

Unfortunately, the `.rhosts` file presents a major security problem. While the `/etc/hosts.equiv` file is under the system administrator's control and can be managed effectively, any user can create a `.rhosts` file granting access to whomever the user chooses without the system administrator's knowledge.

In a situation in which all of the users' home directories are on a single server and only certain people have superuser access on that server, a good way to prevent a user from using a `.rhosts` file is to create an empty file as superuser in their home directory. You would then change the permissions in this file to 000 so that it would be difficult to change it, even as superuser. This would effectively prevent a user from risking system security by using a `.rhosts` file irresponsibly. It would not, however, solve anything if the user is able to change the effective path to his or her home directory.

The only secure way to manage `.rhosts` files is to completely disallow them. See "How to Search for and Remove `.rhosts` Files" on page 662 for detailed instructions.

As system administrator, you can check the system often for violations of this policy. One possible exception to this policy is for the root account—you might need to have a `.rhosts` file to perform network backups and other remote services.

Linking Remote Logins

Provided your system is configured properly, you can link remote logins. For example, a user on `earth` logs in to `jupiter`, and from there decides to log in to `pluto`.

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

To link remote logins without having to supply a password, you must have the `/etc/hosts.equiv` or `.rhosts` file set up correctly.

Direct vs. Indirect Remote Logins

The `rlogin` command allows you to log in to a remote system directly or indirectly.

A direct remote login is attempted with the default user name; that is, the user name of the individual currently logged in to the local system. This is the most common form of remote login.

An indirect remote login is attempted with a different user name, which is supplied during the remote login operation. This is the type of remote login you might attempt from a workstation that you borrowed temporarily. For instance, if you were in a coworker's office and needed to examine files in your home directory, you might log in to your system remotely, from your coworker's system, but you would perform an indirect remote login, supplying your own user name.

The dependencies between direct and indirect logins and authentication methods are summarized in the table below.

TABLE 43–2 Dependencies Between Login Method and Authentication Method (`rlogin`)

Type of Login	User Name Supplied By	Authentication	Password
Direct	System	Network	None
		System	Required
Indirect	User	Network	None
		System	Required

What Happens After You Log In Remotely

When you log in to a remote system, the `rlogin` command attempts to find your home directory. If the `rlogin` command can't find your home directory, it will assign you to the remote system's root (/) directory. For example:

```
Unable to find home directory, logging in with /
```

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

For example, if your usual prompt displays your system name and working directory, and when you log in, your working directory is your home directory, your login prompt looks like this:

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

Then when you log in to a remote system, you will see a similar prompt and your working directory will be your home directory, regardless of the directory from which you entered the `rlogin` command:

```
earth(/home/smith): rlogin pluto
.
.
.
pluto(/home/smith):
```

The only difference is that the name of the remote system would take the place of your local system at the beginning of the prompt. The remote file system is parallel to your home directory.

In other words, if you `cd` to `/home` and then run `ls`, this is what you'll see:

```
earth(home/smith): cd ..
earth(/home): ls
smith  jones
```

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

1. **Become superuser.**

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

```
# find home-directories -name .rhosts -print -exec rm{}
```

home-directories	Identifies the path to a directory where users' home directories are located. Note that you can enter multiple paths to search more than one home directory at a time.
`-name .rhosts`	Identifies the filename.
`-print`	Prints the current path name.
`-exec rm {} \;`	Tells the `find` command to apply the `rm` command to all files identified using the matching filename.

The `find` command starts at the designated directory and searches for any file named `.rhosts`. If it finds any, it prints the path on the screen and removes it.

Example—Searching For and Removing `.rhosts` Files

The following example searches and removes `.rhosts` files in all the user's home directories located in the `/export/home` directory.

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

▼ How to Find Out If a Remote System Is Operating

Find out if a remote system is operating by using the `ping` command.

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

system-name	The name of the remote system.
ip-address	The IP address of the remote system.

The `ping` command returns one of three messages:

Status Message	Explanation
system-name `is alive`	The system can be accessed over the network.
`ping:unknown host` *system-name*	The system name is unknown.
`ping:no answer from` *system-name*	The system is known, but is not currently operating.

If the system you "ping" is located in a different domain, the return message can also contain routing information, which you can ignore.

The `ping` command has a time-out of 20 seconds. In other words, if it does not get a response within 20 seconds, it returns the third message. You can force `ping` to wait longer (or less) by entering a *time-out* value, in seconds:

`$` **`ping`** *system-name* | *ip-address time-out*

For more information, see `ping`(1M).

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

Find who is logged in to a remote system by using the `rusers`(1) command.

`$` **`rusers`** [**`-l`**] *remote-system-name*

`rusers`	(No options) Displays the name of the system followed by the name of users currently logged in to it, including root.
`-l`	Displays additional information about each user: the user's login window, login time and date, amount of time logged in, and the name of the remote system from which the user logged on.

Example—Finding Who Is Logged In to a Remote System

The following example shows the short output of `rusers`.

```
$ rusers pluto
pluto    smith  jones
```

In the following example, the long version of `rusers` show that two users are logged in to the remote system `starbug`. The first user logged in from the system console on September 10 and has been logged on for 137 hours and 15 minutes. The second user logged in from a remote system, `mars`, on September 14.

```
$ rusers -l starbug
root         starbug:console           Sep 10 16:13  137:15
rimmer       starbug:pts/0             Sep 14 14:37         (mars)
```

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

Log in to a remote system using the `rlogin`(1) command.

```
$ rlogin [-l user-name] system-name
```

`rlogin`	(No options) Logs you in to the remote system *directly*; in other words, with your current user name.
`-l` *user-name*	Logs you into the remote system *indirectly;* in other words, with the user name you supply.

If the network attempts to authenticate you, you won't be prompted for a password. If the remote system attempts to authenticate you, you will be asked to provide a password.

If the operation succeeds, the `rlogin` command displays brief information about your latest remote login to that system, the version of the operating system running on the remote system, and whether you have mail waiting for you in your home directory.

Example—Logging In to a Remote System (`rlogin`)

The following example shows the output of a direct remote login to `pluto`. The user has been authenticated by the network.

```
$ rlogin starbug
Last login: Mon Jul 12 09:28:39 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

The following example shows the output of an indirect remote login to `pluto`, with the user being authenticated by the remote system.

```
$ rlogin -l smith pluto
password: user-password
Last login: Mon Jul 12 11:51:58 from venus
Sun Microsystems Inc.   SunOS 5.8       February 2000
starbug:
```

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

Log out from a remote system by using the `exit`(1) command.

```
$ exit
```

Example—Logging Out From a Remote System (`exit`)

This example shows the user `smith` logging out from the system `pluto`.

```
$ exit
pluto% logout
Connection closed.
earth%
```

Logging In to a Remote System (`ftp`)

The `ftp` command opens the user interface to the Internet's File Transfer Protocol. This user interface, called the command interpreter, enables you to log in to a remote system and perform a variety of operations with its file system. The principal operations are summarized in the table below.

The main benefit of `ftp` over `rlogin` and `rcp` is that `ftp` does not require the remote system to be running UNIX. (The remote system does, however, need to be configured for TCP/IP communications.) On the other hand, `rlogin` provides access to a richer set of file manipulation commands than `ftp` does.

Authentication for Remote Logins (`ftp`)

Authentication for `ftp` remote login operations can be established either by:

- Including your password entry in the remote system's `/etc/passwd` file or equivalent network information service map or table.
- Establishing an anonymous `ftp` account on the remote system.

Essential `ftp` Commands

TABLE 43–3 Essential `ftp` Commands

Command	Description
`ftp`	Accesses the `ftp` command interpreter
`ftp` *remote-system*	Establishes an `ftp` connection to a remote system. For instructions, see "How to Open an `ftp` Connection to a Remote System" on page 667

TABLE 43–3 Essential `ftp` Commands *(Continued)*

Command	Description
`open`	Logs in to the remote system from the command interpreter
`close`	Logs out of the remote system and returns to the command interpreter
`bye`	Quits the `ftp` command interpreter
`help`	Lists all `ftp` commands or, if a command name is supplied, briefly describes what the command does
`reset`	Re-synchronizes the command-reply sequencing with the remote `ftp` server
`ls`	Lists the contents of the remote working directory
`pwd`	Displays the name of the remote working directory
`cd`	Changes the remote working directory
`lcd`	Changes the local working directory
`mkdir`	Creates a directory on the remote system
`rmdir`	Deletes a directory on the remote system
`get, mget`	Copies a file (or multiple files) from the remote working directory to the local working directory
`put, mput`	Copies a file (or multiple files) from the local working directory to the remote working directory
`delete, mdelete`	Deletes a file (or multiple files) from the remote working directory

For more information, see `ftp`(1).

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

1. **Make sure you have `ftp` authentication.**

   You must have `ftp` authentication, as described in "Authentication for Remote Logins (`ftp`)" on page 666.

2. **Open a connection to a remote system by using the `ftp` command.**

   `$` **`ftp`** *remote-system*

   If the connection succeeds, a confirmation message and prompt is displayed.

3. **Enter your user name.**

   `Name (`*remote-system*`:`*user-name*`):` *user-name*

4. **If prompted, enter your password.**

   `331 Password required for` *user-name*`:`
   `Password:` *password*

   If the system you are accessing has an established anonymous `ftp` account, you will be prompted for an email address for the password. If the `ftp` interface accepts your password, it displays a confirmation message and the (`ftp>`) prompt.

   You can now use any of the commands supplied by the `ftp` interface, including help. The principal commands are summarized in Table 43–3.

Example—Opening an `ftp` Connection to a Remote System

This `ftp` session was established by the user `smith` on the remote system `pluto`:

```
$ ftp pluto
Connected to pluto.
220 pluto FTP server ready.
Name (pluto:smith): smith
331 Password required for smith:
Password: password
230 User smith logged in.
ftp>
```

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

Close an `ftp` connection to a remote system by using the `bye` command.

```
ftp> bye
221-You have  transferred 0 bytes in 0 files.
221-Total traffic for this sessions was 172 bytes in 0 transfers.
221-Thanks you for using the FTP service on spdev.
221 Goodbye.
```

A goodbye message appears, followed by your usual shell prompt.

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

1. **Change to a directory on the local system where you want the files from the remote system to be copied.**

   `$` **`cd`** *target-directory*

2. **Establish an `ftp` connection.**

   See "How to Open an `ftp` Connection to a Remote System" on page 667.

3. **Change to the source directory.**

   `ftp>` **`cd`** *source-directory*

   If your system is using the automounter, the home directory of the remote system's user appears parallel to yours, under `/home`.

4. **Make sure you have read permission for the source files.**

   `ftp>` **`ls -l`**

5. **Set the transfer type to `binary`.**

   `ftp>` **`binary`**

6. **To copy a single file, use the `get` command.**

   `ftp>` **`get`** *filename*

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

   `ftp>` **`mget`** *filename* [*filename* `...`]

   You can supply a series of individual file names and you can use wildcard characters. The `mget` command will copy each file individually, asking you for confirmation each time.

8. **Close the `ftp` connections.**

   `ftp>` **`bye`**

Examples—Copying Files From a Remote System (`ftp`)

In this example, the user `kryten` opens an `ftp` connection to the system `pluto`, and uses the `get` command to copy a single file from the `/tmp` directory:

```
$ cd $HOME
ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
```

```
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34344)
(0 bytes).
dtdbcache_:0
filea
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
53 bytes received in 0.022 seconds (2.39 Kbytes/s)
ftp> get filea
200 PORT command successful.
150 ASCII data connection for filea (129.152.221.238,34331)
(0 bytes).
221 Goodbye.
```

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

```
$ ftp> cd /tmp
250 CWD command successful.
ftp> ls files
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34345)
(0 bytes).
fileb
filec
filed
remote: files
21 bytes received in 0.015 seconds (1.36 Kbytes/s)
ftp> cd files
250 CWD command successful.
ftp> mget file*
mget fileb? y
200 PORT command successful.
150 ASCII data connection for fileb (129.152.221.238,34347)
(0 bytes).
226 ASCII Transfer complete.
mget filec? y
200 PORT command successful.
150 ASCII data connection for filec (129.152.221.238,34348)
(0 bytes).
226 ASCII Transfer complete.
mget filed? y
200 PORT command successful.
150 ASCII data connection for filed (129.152.221.238,34351)
(0 bytes).
226 ASCII Transfer complete.200 PORT command successful.
ftp> bye
```

```
221 Goodbye.
```

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

1. **Change to the source directory on the local system.**

   The directory from which you enter the `ftp` command will be the local working directory, and thus the source directory for this operation.

2. **Establish an `ftp` connection.**

   See "How to Open an `ftp` Connection to a Remote System" on page 667.

3. **Change to the target directory.**

   `ftp>` **`cd`** *target-directory*

   Remember, if your system is using the automounter, the home directory of the remote system's user appears parallel to yours, under `/home`.

4. **Make sure you have write permission to the target directory.**

   `ftp>` **`ls -l`** *target-directory*

5. **Set transfer type to binary.**

   `ftp>` **`binary`**

6. **To copy a single file, use the `put` command.**

   `ftp>` **`put`** *filename*

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

   `ftp>` **`mput`** *filename* [*filename* . . .]

   You can supply a series of individual file names and you can use wildcard characters. The `mput` command will copy each file individually, asking you for confirmation each time.

8. **To close the `ftp` connection, type `bye`.**

   `ftp>` **`bye`**

Examples—Copying Files to a Remote System (`ftp`)

In this example, the user `kryten` opens an `ftp` connection to the system `pluto`, and uses the `put` command to copy a file from his system to the `/tmp` directory on system `pluto`:

```
$ cd /tmp
ftp pluto
```

```
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> put filef
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
226 Transfer complete.
ftp> ls
200 PORT command successful.
150 ASCII data connection for /bin/ls (129.152.221.238,34357) (0 bytes).
dtdbcache_:0
filea
filef
files
ps_data
speckeysd.lock
226 ASCII Transfer complete.
60 bytes received in 0.058 seconds (1.01 Kbytes/s)
ftp> bye
221 Goodbye.
```

In this example, the same user `kryten` uses the `mput` command to copy a set of files from his home directory to `pluto`'s `/tmp` directory. Note that `kryten` can accept or reject individual files in the set.

```
$ cd $HOME/testdir
$ ls
test1   test2   test3
$ ftp pluto
Connected to pluto.
220 pluto FTP server (SunOS 5.8) ready.
Name (pluto:kryten): kryten
331 Password required for kryten.
Password: xxx
230 User kryten logged in.
ftp> cd /tmp
250 CWD command successful.
ftp> mput test*
mput test1? y
200 PORT command successful.
150 ASCII data connection for test1 (129.152.221.238,34365).
226 Transfer complete.
mput test2? y
200 PORT command successful.
150 ASCII data connection for test2 (129.152.221.238,34366).
226 Transfer complete.
mput test3? y
200 PORT command successful.
150 ASCII data connection for filef (129.152.221.238,34356).
```

```
226 Transfer complete.
ftp> bye
221 Goodbye.
```

Remote Copying With rcp

The rcp command copies files or directories between a local and a remote system or between two remote systems. You can use it from a remote system (after logging in with the rlogin command) or from the local system (without logging in to a remote system).

With rcp, you can perform the following remote copy operations:

- Copy a file or directory from your system to a remote system
- Copy a file or directory from a remote system to your local system
- Copy a file or directory between remote systems from your local system

If you have the automounter running, you can perform these remote operations with the cp command. However, the range of cp is constrained to the virtual file system created by the automounter and to operations relative to a user's home directory and, since rcp performs the same operations without these constraints, this section will describe only the rcp versions of these tasks.

Security Considerations for Copy Operations

To copy files or directories between systems, you must have permission to log in and copy files.

Caution – Both the cp and rcp commands can overwrite files without warning. Make sure file names are correct before executing the command.

Specifying Source and Target

With the rcp command in the C-shell, you can specify source (the file or directory you want to copy) and target (the location into which you will copy the file or directory) with either absolute or abbreviated path names.

	Absolute Pathnames	Abbreviated Pathnames
From Local System	`mars:/home/jones/myfile.txt`	`~jones/myfile.txt`
After Remote Login	`/home/jones/myfile.txt`	`~jones/myfile.txt`

Absolute path names identify files or directories mounted on a particular system. In the example above, the first absolute path name identifies a file (`MyFile.txt`) on the `mars` system. Abbreviated path names identify files or directories relative to a user's home directory, wherever that might reside. In the first example above, the abbreviated path name identifies the same file, `MyFile.txt`, but uses "~" symbol to indicate the `jones` home directory. In effect . . .

```
~ = mars:/home/jones
```

The examples on the second line, above, demonstrate the user of absolute and abbreviated path names after a remote login. There is no difference for the abbreviated path name, but because the remote login operation mounted the `jones` home directory onto the local system (parallel to the local user's home directory), the absolute path name no longer requires the system name `mars`. For more information about how a remote login operation mounts another user's home directory, see "What Happens After You Log In Remotely" on page 662.

The table below provides a sample of absolute and abbreviated path names recognized by the C shell. It uses the following terminology:

- Working directory - The directory from which the `rcp` command is entered. Can be remote or local.
- Current user - The user name under which the `rcp` command is entered.

TABLE 43–4 Allowed Syntaxes for Directory and File Names

Logged in to	Syntax	Description
Local system	.	The local working directory
	path/filename	The *path* and *filename* in the local working directory
	~	The current user's home directory
	~/path/filename	The *path* and *filename* beneath the current user's home directory
	~user	The home directory of *user*
	~user/path/filename	The *path* and *filename* beneath the home directory of *user*

TABLE 43–4 Allowed Syntaxes for Directory and File Names *(Continued)*

Logged in to	Syntax	Description
	remote-system:path/filename	The *path* and *filename* in the remote working directory
Remote system	.	The remote working directory
	filename	The *filename* in the remote working directory
	path/filename	The *path* and *filename* in the remote working directory
	~	The current user's home directory
	~/path/filename	The *path* and *filename* in the current user's home directory
	~user	The home directory of *user*
	~/user/path/filename	The *path* and *filename* beneath the home directory of *user*
	local-system:path/filename	The *path* and *filename* in the local working directory

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

1. **Be sure you have permission to copy.**

   You should at least have read permission on the source system and write permission on the target system.

2. **Determine the location of the source and target.**

   If you don't know the path of the source or target, you can first log into the remote system with the `rlogin` command, as described in "How to Log In to a Remote System (`rlogin`) " on page 665. Then, navigate through the remote system until you find the location. You can then perform the next step without logging out.

3. **Copy the file or directory.**

   `$` **`rcp`** `[-r]` *source-file | directory target-file | directory*

`rcp`	(No options) Copies a single file from the source to the target.
`-r`	Copies a directory from the source to the target.

This syntax applies whether you are logged in to the remote system or in to the local system. Only the path name of the file or directory changes, as described in Table 43–4 and as illustrated in the examples below.

You can use the "~" and "." characters to specify the path portions of the local file or directory names. Note, however, that "~" applies to the current user, not the remote system, and that "." applies to system you are logged into. For explanations of these symbols, see Table 43–4.

Examples—Copying Files Between a Local and a Remote System (`rcp`)

Here are several examples of using `rcp` to copy files to and from local and remote systems.

EXAMPLE 43–1 Using `rcp` to Copy a Remote File to a Local System

In this example, `rcp` is used to copy the file `letter.doc` from the `/home/jones` directory of the remote system `pluto` to the working directory (`/home/smith`) on the local system, `earth`:

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

In this instance, the `rcp` operation is performed without a remote login. In this case, the "." symbol at the end of the command line refers to the local system, not the remote system.

The target directory is the also local user's home directory, so it can also be specified with the "~" symbol.

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

EXAMPLE 43–2 Using `rlogin` and `rcp` to Copy a Remote File to a Local System

In this example, the `rcp` operation is run after the `rlogin` command is executed to copy a file from a remote to a local system. Although the flow of the operation is the same as that of the previous example, the paths change to take into account the remote login:

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

Using the "." symbol at the end of the command line would be inappropriate in this instance. Because of the remote login, the symbol would simply refer to the remote system—essentially directing `rcp` to create a duplicate file. The "~" symbol, however, refers to the current user's home directory, even when logged in to a remote system.

EXAMPLE 43–3 Using `rcp` to Copy a Local File to a Remote System

In this example, `rcp` is used to copy the file `notice.doc` from the home directory (`/home/smith`) of the local system `earth` to the `/home/jones` directory of the remote system, `pluto`:

```
earth(/home/smith): rcp notice.doc pluto:/home/jones
```

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

In this instance, the `rcp` operation from the previous example is repeated, but `rcp` is entered from a different working directory on the local system (`/tmp`). Note the use of the "~" symbol to refer to the current user's home directory:

```
earth(/tmp): rcp ~/notice.doc pluto:/home/jones
```

EXAMPLE 43–4 Using `rlogin` and `rcp` to Copy a Local File to a Remote System

In this example, the `rcp` operation is run after the `rlogin` command is executed to copy a local file to a remote directory. Although the flow of the operation is the same as that of the previous example, the paths change to take into account the remote login.

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

In this instance, the "~" symbol can be used to denote the current user's home directory, even though it is on the local system. The "." symbol refers to the working directory on the remote system because the user is logged in to the remote system. Here is an alternative syntax that performs the same operation:

```
pluto(/home/jones): rcp earth:/home/smith/notice.doc /home/jones
```

CHAPTER 44

Monitoring Network Services Topics

Chapter 45	Provides step-by-step instructions for monitoring network services

CHAPTER 45

Monitoring Network Performance (Tasks)

This chapter describes the how to monitor network performance. This is a list of the step-by-step instructions in this chapter.

Monitoring Network Performance

Table 45–1 describes the commands available for monitoring network performance.

TABLE 45–1 Network Monitoring Commands

Command	Use This Command to ...
`ping`	Look at the response of hosts on the network.
`spray`	Test the reliability of your packet sizes. It can tell you whether packets are being delayed or dropped.
`snoop`	Capture packets from the network and trace the calls from each client to each server.
`netstat`	Display network status, including state of the interfaces used for TCP/IP traffic, the IP routing table, and the per-protocol statistics for `UDP`, `TCP`, `ICMP`, and `IGMP`.
`nfsstat`	Display a summary of server and client statistics that can be used to identify NFS problems.

▼ How to Check the Response of Hosts on the Network

Check the response of hosts on the network with the `ping` command.

`$` **`ping`** *hostname*

If you suspect a physical problem, you can use `ping` to find the response time of several hosts on the network. If the response from one host is not what you would expect, you can investigate that host. Physical problems could be caused by:

- Loose cables or connectors
- Improper grounding
- Missing termination
- Signal reflection

For more information about this command, see `ping`(1M).

Examples—Checking the Response of Hosts on the Network

The simplest version of `ping` sends a single packet to a host on the network. If it receives the correct response, it prints the message *host* `is alive`.

```
$ ping elvis
elvis is alive
```

With the `-s` option, `ping` sends one datagram per second to a host. It then prints each response and the time it took for the round trip. For example:

```
$ ping -s pluto
64 bytes from pluto (123.456.78.90): icmp_seq=0. time=10. ms
64 bytes from pluto (123.456.78.90): icmp_seq=5. time=0. ms
64 bytes from pluto (123.456.78.90): icmp_seq=6. time=0. ms
^C
----pluto PING Statistics----
8 packets transmitted, 8 packets received, 0% packet loss

round-trip (ms) min/avg/max = 0/2/10
```

▼ How to Send Packets to Hosts on the Network

Test the reliability of your packet sizes with the `spray` command.

```
$ spray [ -c count -d interval -l packet_size] hostname
```

`-i` *count*	Number of packets to send.
`-d` *interval*	Number of microseconds to pause between sending packets. If you don't use a delay, you might run out of buffers.
`-l` *packet_size*	Is the packet size.
hostname	Is the system to send packets.

For more information about this command, see `spray`(1M).

Example—Sending Packets to Hosts on the Network

The following example sends 100 packets to a host (`-c 100`) with each packet having a size of 2048 bytes (`-l 2048`). The packets are sent with a delay time of 20 microseconds between each burst (`-d 20`).

```
$ spray -c 100 -d 20 -l 2048 pluto
sending 100 packets of length 2048 to pluto ...
no packets dropped by pluto
279 packets/sec, 573043 bytes/sec
```

▼ How to Capture Packets From the Network

To capture packets from the network and trace the calls from each client to each server, use `snoop`. This command provides accurate timestamps that allow some network performance problems to be isolated quickly. For more information, see `snoop`(1M).

```
# snoop
```

Dropped packets could be caused by insufficient buffer space, or an overloaded CPU.

▼ How to Check the Network Status

Display network status information, such as statistics about the state of network interfaces, routing tables, and various protocols, with the `netstat` command.

```
$ netstat [-i] [-r] [-s]
```

`-i`	Displays the state of the TCP/IP interfaces
`-r`	Displays the IP routing table
`-s`	Displays statistics for the `UDP`, `TCP`, `ICMP`, and `IGMP` protocols

For more information, see `netstat`(1M).

Examples—Checking the Network Status

The following example shows output from the `netstat -i` command, which displays the state of the interfaces used for TCP/IP traffic.

```
$ netstat -i
Name  Mtu  Net/Dest    Address      Ipkts  Ierrs Opkts  Oerrs Collis Queue
lo0   8232 software    localhost     1280   0     1280     0       0    0
le0   1500 loopback    venus      1628480   0   347070    16   39354    0
```

This display shows how many packets a machine has transmitted and received on each interface. A machine with active network traffic should show both `Ipkts` and `Opkts` continually increasing.

Calculate the network collisions rate by dividing the number of collision counts (`Collis`) by the number of out packets (`Opkts`). In the previous example, the collision rate is 11 percent. A network-wide collision rate greater than 5 to 10 percent can indicate a problem.

Calculate the input packet error rate by dividing the number of input errors by the total number of input packets (`Ierrs/Ipkts`). The output packet error rate is the number of output errors divided by the total number of output packets (`Oerrs/Opkts`). If the input error rate is high (over 0.25 percent), the host might be dropping packets.

The following example shows output from the `netstat -s` command, which displays the per-protocol statistics for the `UDP`, `TCP`, `ICMP`, and `IGMP` protocols.

```
UDP
    udpInDatagrams      =196543    udpInErrors         =     0
    udpOutDatagrams     =187820

TCP
 tcpRtoAlgorithm     =     4    tcpRtoMin           =   200
    tcpRtoMax           = 60000    tcpMaxConn          =    -1
    tcpActiveOpens      = 26952    tcpPassiveOpens     =   420
    tcpAttemptFails     =  1133    tcpEstabResets      =     9
    tcpCurrEstab        =    31    tcpOutSegs          =3957636
    tcpOutDataSegs      =2731494    tcpOutDataBytes     =1865269594
    tcpRetransSegs      = 36186    tcpRetransBytes     =3762520
    tcpOutAck           =1225849    tcpOutAckDelayed    =165044
    tcpOutUrg           =     7    tcpOutWinUpdate     =   315
```

```
    tcpOutWinProbe      =     0    tcpOutControl       = 56588
    tcpOutRsts          =   803    tcpOutFastRetrans   =   741
    tcpInSegs           =4587678
    tcpInAckSegs        =2087448    tcpInAckBytes       =1865292802
    tcpInDupAck         =109461    tcpInAckUnsent      =     0
    tcpInInorderSegs    =3877639    tcpInInorderBytes   =-598404107
    tcpInUnorderSegs    = 14756    tcpInUnorderBytes   =17985602
    tcpInDupSegs        =    34    tcpInDupBytes       = 32759
    tcpInPartDupSegs    =   212    tcpInPartDupBytes   =134800
    tcpInPastWinSegs    =     0    tcpInPastWinBytes   =     0
    tcpInWinProbe       =   456    tcpInWinUpdate      =     0
    tcpInClosed         =    99    tcpRttNoUpdate      =  6862
    tcpRttUpdate        =435097    tcpTimRetrans       = 15065
    tcpTimRetransDrop   =    67    tcpTimKeepalive     =   763
    tcpTimKeepaliveProbe=     1    tcpTimKeepaliveDrop =     0

IP
 ipForwarding        =     2    ipDefaultTTL        =   255
    ipInReceives        =11757234    ipInHdrErrors       =     0
    ipInAddrErrors      =     0    ipInCksumErrs       =     0
    ipForwDatagrams     =     0    ipForwProhibits     =     0
    ipInUnknownProtos   =     0    ipInDiscards        =     0
    ipInDelivers        =4784901    ipOutRequests       =4195180
    ipOutDiscards       =     0    ipOutNoRoutes       =     0
    ipReasmTimeout      =    60    ipReasmReqds        =  8723
    ipReasmOKs          =  7565    ipReasmFails        =  1158
    ipReasmDuplicates   =     7    ipReasmPartDups     =     0
    ipFragOKs           = 19938    ipFragFails         =     0
    ipFragCreates       =116953    ipRoutingDiscards   =     0
    tcpInErrs           =     0    udpNoPorts          =6426577
    udpInCksumErrs      =     0    udpInOverflows      =   473
    rawipInOverflows    =     0

ICMP
 icmpInMsgs          =490338    icmpInErrors        =     0
    icmpInCksumErrs     =     0    icmpInUnknowns      =     0
    icmpInDestUnreachs  =   618    icmpInTimeExcds     =   314
    icmpInParmProbs     =     0    icmpInSrcQuenchs    =     0
    icmpInRedirects     =   313    icmpInBadRedirects  =     5
    icmpInEchos         =   477    icmpInEchoReps      =    20
    icmpInTimestamps    =     0    icmpInTimestampReps =     0
    icmpInAddrMasks     =     0    icmpInAddrMaskReps  =     0
    icmpInFragNeeded    =     0    icmpOutMsgs         =   827
    icmpOutDrops        =   103    icmpOutErrors       =     0
    icmpOutDestUnreachs =    94    icmpOutTimeExcds    =   256
    icmpOutParmProbs    =     0    icmpOutSrcQuenchs   =     0
    icmpOutRedirects    =     0    icmpOutEchos        =     0
    icmpOutEchoReps     =   477    icmpOutTimestamps   =     0
    icmpOutTimestampReps=     0    icmpOutAddrMasks    =     0
    icmpOutAddrMaskReps =     0    icmpOutFragNeeded   =     0
    icmpInOverflows     =     0
IGMP:
        0 messages received
        0 messages received with too few bytes
```

```
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent
```

The following example shows output from the `netstat -r` command, which displays the IP routing table.

```
Routing Table:
  Destination        Gateway           Flags  Ref   Use    Interface
------------------ -------------------- ----- ----- ------ ---------
localhost          localhost              UH       0   2817  lo0
earth-bb           pluto                  U        3  14293  le0
224.0.0.0          pluto                  U        3      0  le0
default            mars-gate              UG       0  14142
```

The fields in the `netstat -r` report are described in Table 45–2.

TABLE 45–2 Output From the `netstat -r` Command

Field Name		Description
`Flags`	`U`	The route is up
	`G`	The route is through a gateway
	`H`	The route is to a host
	`D`	The route was dynamically created using a redirect
`Ref`		Shows the current number of routes sharing the same link layer
`Use`		Indicates the number of packets sent out
`Interface`		Lists the network interface used for the route

▼ How to Display NFS Server and Client Statistics

The NFS distributed file service uses a remote procedure call (RPC) facility that translates local commands into requests for the remote host. The remote procedure calls are synchronous. That is, the client application is blocked or suspended until the server has completed the call and returned the results. One of the major factors affecting NFS performance is the retransmission rate.

If the file server cannot respond to a client's request, the client retransmits the request a specified number of times before it quits. Each retransmission imposes system

overhead, and increases network traffic. Excessive retransmissions can cause network performance problems. If the retransmission rate is high, you could look for:

- Overloaded servers that take too long to complete requests
- An Ethernet interface dropping packets
- Network congestion, which slows the packet transmission

Table 45–3 describes the `nfsstat` options to display client and server statistics.

TABLE 45–3 Commands for Displaying Client/Server Statistics

Use ...	To Display ...
`nfsstat -c`	Client statistics
`nfsstat -s`	Server statistics
`netstat -m`	Network statistics for each file system

Use `nfsstat -c` to show client statistics, and `nfsstat -s` to show server statistics. Use `netstat -m` to display network statistics for each file system. For more information, see `nfsstat`(1M).

Examples—Displaying NFS Server and Client Statistics

The following example displays RPC and NFS data for the client `pluto`.

```
$ nfsstat -c

        Client rpc:
Connection oriented:
calls    badcalls  badxids  timeouts newcreds  badverfs   timers
1595799  1511      59       297      0         0          0
cantconn nomem     interrupts
1198      0         7
Connectionless:
calls    badcalls  retrans  badxids  timeouts  newcreds   badverfs
80785    3135      25029    193      9543      0          0
timers   nomem     cantsend
17399    0         0

Client nfs:
calls    badcalls  clgets   cltoomany
1640097  3112      1640097  0
Version 2: (46366 calls)
null     getattr   setattr  root     lookup     readlink  read
0 0%     6589 14%  2202 4%  0 0%     11506 24%  0 0%      7654 16%
wrcache  write     create   remove   rename     link      symlink
0 0%     13297 28% 1081 2%  0 0%     0 0%       0 0%      0 0%
mkdir    rmdir     readdir  statfs
24 0%    0 0%      906 1%   3107 6%
```

```
Version 3: (1585571 calls)
null    getattr    setattr  lookup     access     readlink  read
0 0%    508406 32% 10209 0% 263441 16% 400845 25% 3065 0%  117959 7%
write    create     mkdir    symlink    mknod    remove   rmdir
69201 4% 7615 0%    42 0%    16 0%       0 0%     7875 0%  51 0%
rename   link       readdir  readdir+   fsstat   fsinfo   pathconf
929 0%   597 0%     3986 0%  185145 11% 942 0%   300 0%   583 0%
commit
4364 0%

Client nfs_acl:
Version 2: (3105 calls)
null       getacl     setacl     getattr    access
0 0%       0 0%       0 0%       3105 100%  0 0%
Version 3: (5055 calls)
null       getacl     setacl
0 0%       5055 100%  0 0%
```

The output of the `nfsstat -c` command is described in Table 45–4.

TABLE 45–4 Output From the `nfsstat -c` Command

Field	Description
`calls`	The total number of calls sent.
`badcalls`	The total number of calls rejected by RPC.
`retrans`	The total number of retransmissions. For this client, the number of retransmissions is less than 1 percent (10 timeouts out of 6888 calls). These might be caused by temporary failures. Higher rates might indicate a problem.
`badxid`	The number of times that a duplicate acknowledgment was received for a single NFS request.
`timeout`	The number of calls that timed out.
`wait`	The number of times a call had to wait because no client handle was available.
`newcred`	The number of times the authentication information had to be refreshed.
`timers`	The number of times the time-out value was greater than or equal to the specified time-out value for a call.
`readlink`	The number of times a `read` was made to a symbolic link. If this number is high (over 10 percent), it could mean that there are too many symbolic links.

The following example shows output from the `nfsstat -m` command.

```
pluto$ nfsstat -m
/usr/man from pluto:/export/svr4/man
Flags: vers=2,proto=udp,auth=unix,hard,intr,dynamic,
```

```
       rsize=8192, wsize=8192,retrans=5
Lookups: srtt=13 (32ms), dev=10 (50ms), cur=6 (120ms)
All:     srtt=13 (32ms), dev=10 (50ms), cur=6 (120ms)
```

This output of the `nfsstat -m` command, which is displayed in milliseconds, is described in Table 45–5.

TABLE 45–5 Output From the `nfsstat -m` Command

Field	Description
`srtt`	The smoothed average of the round-trip times
`dev`	The average deviations
`cur`	The current “expected” response time

If you suspect that the hardware components of your network are creating problems, you need to look carefully at the cabling and connectors.

Glossary

broadcast A data link layer procedure used to transmit packets to every machine on a subnet. Broadcast packets are typically not routed beyond the submet.

Directory Agent (DA) Optional SLP agent that stores and maintains a cache of service advertisements sent by the service agent (SA). When deployed, the DA resolves user agent (UA) service requests. The DA responds to active solicitations from the SA and UA for directory advertisements. As a result, the SA and UA discover the associated DAs and *scopes*. A DA sends periodic unsolicited advertisements through which UAs and SAs discover the DA within shared scopes.

legacy services A networked service that is not SLP-enabled. You can create a proxy registration to register a legacy service with SLP. SLP-based clients can then discover legacy services (see Chapter 20).

multicast A network layer procedure used to send datagram packets to multiple machines on an IP network. Packets are not handled by every machine as is the case with broadcast routing. Multicast requires that routers be configured with special routing protocols.

scope A grouping of UAs and SAs arranged administratively, topologically, or in some other manner. You can use scopes to tailor how you provision access to services across the enterprise.

service advertisements Information distributed by an SA that describes a service. A service advertisement consists of a URL and a collection of attribute/value list pairs that describe a service. All service advertisements have a lifetime. After the lifetime expires, a service advertisement is no longer valid unless re-registered.

Service Agent (SA) The SLP agent that maintains service advertisements for networked serviced. If no DA is available, the SA answers multicast service requests from UAs. If a DA is available, the SA registers and, optionally, deregisters services with DAs that support its scopes.

service URL A URL that is used to advertise the network location of services. The URL contains the service type, host name, or network address of the service host. The URL might also contain a port number and other information required to use the service.

SLP daemon (slpd) The daemon process that acts as a DA or an SA server in the Solaris implementation of SLP. Service processes on the host register service advertisements with `slpd` rather than maintaining the advertisements individually. Each process contains an SA client library that communicates with `slpd` when the daemon is configured as the SA server. The SLP daemon forwards all registrations and deregistrations to DAs. The daemon times out expired service advertisements and maintains a table of the available DAs by performing active and passive DA discovery. Through such mechanisms, DA information is provided to UA clients. UA clients use `slpd` on a host only for DA information. The SLP daemon is installed on a host as part of the Solaris 9 operating environment.You can optionally configure `slpd` as a DA.

User Agent (UA) The SLP agent that acts on behalf of the user application. The agent queries for the identity of corresponding scopes, directory agents, and service advertisements.