System Administration Guide: Network Services

Part No: 816–4555–21
August 2011

ORACLE

Copyright © 2002, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés.Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. UNIX est une marque déposée concédé sous license par X/Open Company, Ltd.

111219@25097

Contents

Figures

Tables

Examples

Preface

System Administration Guide: Network Services is part of a multivolume set that covers a significant part of the Oracle Solaris system administration information. This book assumes that you have already installed the Oracle Solaris 10 operating system, and you have set up any networking software that you plan to use.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

In this document, these x86 related terms mean the following:

- x86 refers to the larger family of 64-bit and 32-bit x86 compatible products.
- x64 relates specifically to 64-bit x86 compatible CPUs.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Oracle Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems that run the Solaris 10 release. To use this book, you should have one to two years of UNIX system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Guides Are Organized

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

Book Title	Topics
System Administration Guide: Basic Administration	User accounts and groups, server and client support, shutting down and booting a system, managing services, and managing software (packages and patches)
System Administration Guide: Advanced Administration	Terminals and modems, system resources (disk quotas, accounting, and crontabs), system processes, and troubleshooting Oracle Solaris software problems
System Administration Guide: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data
System Administration Guide: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, Solaris IP filter, Mobile IP, IP network multipathing (IPMP), and IPQoS
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP and transitioning from NIS+ to LDAP
System Administration Guide: Naming and Directory Services (NIS+)	NIS+ naming and directory services
System Administration Guide: Network Services	Web cache servers, time-related services, network file systems (NFS and autofs), mail, SLP, and PPP
System Administration Guide: Printing	Printing topics and tasks, using services, tools, protocols, and technologies to set up and administer printing services and printers
System Administration Guide: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Solaris Cryptographic Framework, privileges, RBAC, SASL, and Solaris Secure Shell
System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones	Resource management topics projects and tasks, extended accounting, resource controls, fair share scheduler (FSS), physical memory control using the resource capping daemon (rcapd), and resource pools; virtualization using Solaris Zones software partitioning technology and `lx` branded zones
Oracle Solaris ZFS Administration Guide	ZFS storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, using ZFS on an Oracle Solaris system with zones installed, emulated volumes, and troubleshooting and data recovery
Oracle Solaris Trusted Extensions Administrator's Procedures	System administration that is specific to the Oracle Solaris' Trusted Extensions feature
Oracle Solaris Trusted Extensions Configuration Guide	Starting with the Solaris 10 5/08 release, describes how to plan for, enable, and initially configure the Oracle Solaris' Trusted Extensions feature

Related Books

This is a list of related documentation that is referred 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 (NIS+)*
- *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones*
- *System Administration Guide: Security Services*
- Anderson, Bart, Bryan Costales, and Harry Henderson. *UNIX Communications*. Howard W. Sams & Company, 1987.
- Costales, Bryan. *sendmail, Third Edition*. O'Reilly & Associates, Inc., 2002.
- 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.

Related Information

For information on PPPoE licensing terms, refer to the incorporated material at the following locations:

/var/sadm/pkg/SUNWpppd/install/copyright

/var/sadm/pkg/SUNWpppdu/install/copyright

/var/sadm/pkg/SUNWpppg/install/copyright

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

PART I

Network Services Topics

This section provides an overview of the book, as well as overview, task, and reference information for the NCA and NTP services.

◆ ◆ ◆ CHAPTER 1

Network Service (Overview)

This chapter provides a list of the major topics covered in this book. In addition it includes a description of the PERL service that is included in this release.

Topics for the Oracle Solaris 10 Update 10 Release

The following services or utilities are covered in this book:

"Perl 5" on page 42
: The Practical Extraction and Report Language (Perl) is a tool that can be used to generate scripts to assist with system administration tasks.

Chapter 2, "Managing Web Cache Servers"
: NCA provides improved web server performance by caching web pages.

Chapter 3, "Time-Related Services"
: NTP and time-related utilities can be used to synchronize time for many systems.

Chapter 4, "Managing Network File Systems (Overview)"
: NFS is a protocol that provides the ability to access file systems from a remote host.

Chapter 7, "SLP (Overview)"
: SLP is a dynamic service discovery protocol.

Chapter 12, "Mail Services (Overview)"
: Mail services allow for a message to be sent to one or more people while routing the message over whatever networks are necessary.

Chapter 15, "Solaris PPP 4.0 (Overview)"
: PPP is a protocol that provides point-to-point links between remote hosts.

Chapter 24, "UUCP (Overview)"
: UUCP enables hosts to exchange files.

Chapter 27, "Working With Remote Systems (Overview)"
These commands are used to access files on remote systems. The commands include `ftp`, `rlogin` and `rcp`.

Perl 5

This Solaris release includes Practical Extraction and Report Language (Perl) 5.8.4, 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 task 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. If you wish to build and install add-on modules from CPAN using `gcc`, you can do so using the `/usr/perl5/5.8.4/bin/perlgcc` script. See the `perlgcc`(1) man page for details.

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.

```
% setenv 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 5.8.4 version of Perl is compatible with the previous version. Scripts do not have to be rebuilt or recompiled to function. However, any XSUB-based (.xs) modules require recompilation and reinstallation.

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 v5.8.4:
 .
 .
```

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). Procedures for using NCA and reference material about NCA are included. Also for the Solaris 10 6/06 release, an introduction to using the Secure Sockets Layer (SSL) and procedures for using the SSL kernel proxy to improve the performance of the SSL packet processing are added.

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 improves dramatically. In situations where the requested page is 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. If you run other large processes on a server that runs NCA, problems can result.

NCA provides logging support in that NCA 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 format to common log format (CLF).

The Solaris 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 specified number of records (`-n`).
- `logd_path_name` in `ncalogd.conf` can specify either a raw device, a file, or a combination of the two.
- Support for a web server to open multiple AF_NCA sockets. With multiple sockets, you can have different web servers that run on one server.
- A new configuration file that is called `/etc/nca/ncaport.conf`. The file can be used to manage the IP addresses and ports that NCA uses. Your web server might not provide native support of the AF_NCA socket. If your server lacks this support, use this file and the NCA socket utility library to convert an AF_INET socket to an AF_NCA socket.

Web Servers Using the Secure Sockets Layer Protocol

In the Solaris 10 6/06 release, an Apache 2.0 and a Sun Java System Web Server may be configured to use the Secure Sockets Layer (SSL) Protocol. The protocol provides confidentiality, message integrity and end point authentication between two applications. The kernel has been changed to accelerate the SSL traffic.

The SSL kernel proxy implements the server side of the SSL protocol. The proxy offers better SSL performance for server applications, like web servers, over applications using user-level SSL libraries. The performance improvement may be as high as +35% depending on the workload of the application.

The SSL kernel proxy supports the SSL 3.0 and TLS 1.0 protocols, as well as most common cipher suites. See the `ksslcfg(1M)` man page for the complete list. The proxy can be configured to fallback to the user-level SSL server for any unsupported cipher suites.

The following procedures show how to configure servers to use the SSL kernel proxy:

- "How to Configure an Apache 2.0 Web Server to Use the SSL Kernel Proxy" on page 53
- "How to Configure a Sun Java System Web Server to Use the SSL Kernel Proxy" on page 55
- "Using the SSL Kernel Proxy in Zones" on page 56

Managing Web Cache Servers (Task Map)

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

Task	Description	For Instructions
Planning for NCA	A list of issues to be resolved before you enable the use of NCA.	"Planning for NCA" on page 47
Enabling NCA	Steps to enable in-kernel caching of web pages on a web server.	"How to Enable Caching of Web Pages" on page 48
Disabling NCA	Steps to disable in-kernel caching of web pages on a web server.	"How to Disable Caching of Web Pages" on page 51
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 Socket Utility Library for NCA" on page 52
Using the SSL kernel proxy with an Apache 2.0 web server	Steps to use the SSL kernel proxy with a web server to improve SSL packet processing.	"How to Configure an Apache 2.0 Web Server to Use the SSL Kernel Proxy" on page 53
Using the SSL kernel proxy with a Sun Java System Web Server	Steps to use the SSL kernel proxy with a web server to improve SSL packet processing.	"How to Configure a Sun Java System Web Server to Use the SSL Kernel Proxy" on page 55
Using the SSL kernel proxy with a web server in a local zone	Steps to use the SSL kernel proxy with a web server in a local zone.	"Using the SSL Kernel Proxy in Zones" on page 56

Planning for NCA

The following sections cover the issues that need to be resolved before starting the NCA service.

System Requirements for NCA

To support NCA, the system must meet these requirements:

- 256 Mbytes RAM must be installed.
- The Solaris 10, or 9 release, or one of the Solaris 8 upgrade releases must be installed.
- Support for a web server which has native support for NCA or a web server whose startup script has been modified to use the Socket Utility Library for NCA:
  - Apache web server, ships with the Solaris 8 upgrade, Solaris 9, and Oracle Solaris 10 releases

- Sun Java System Web Server
- Zeus web server available from Zeus Technology, `http://www.zeus.com`

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

NCA Logging

The NCA service can be configured to log web activity. Generally, NCA logging should be enabled if the web server logging is enabled.

Interpositioning Library for Daemon Support of the Door Server

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`. Suppose that the status is enabled in `/etc/nca/ncakmod.conf`. The version of Apache that is included with the Solaris 9 and Solaris 10 release is already set up to call this library. If you are using IWS or Netscape servers, see "How to Load the Socket Utility Library for NCA" on page 52 to use the new library.

Multiple Instance Support

Systems that have NCA installed often need to run multiple instances of a web server. For instance, a single server might need to support a web server for outside access as well as a web administration server. To separate these servers, you would configure each server to use a separate port.

Administering the Caching of Web Pages (Tasks)

The following sections cover the procedures to enable or disable parts of the service.

▼ How to Enable Caching of Web Pages

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Register the interfaces.

Type 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 (Optional) 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 indicated by the `logd_path_name` entry. The log file can be a raw device or a file. See the following examples for samples of NCA log file paths. See the `ncalogd.conf(4)` man page for more information about the configuration file.

5 (Optional) Define ports for multiple instance support.

Add the port numbers in the `/etc/nca/ncaport.conf` file. This entry causes NCA to monitor port 80 on all configured IP addresses.

```
# cat /etc/nca/ncaport.conf
#
# NCA Kernel Module Port Configuration File
#
  .
  .
ncaport=*/80
```

6 For x86 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 – By setting the `kernelbase`, you reduce the amount of virtual memory that user processes can use to less than 3 Gbytes. This restriction means that the system is not ABI compliant. When the system boots, the console 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 the program on a system that does not have NCA enabled.

---

7 Reboot the server.

Example 2–1 Using a Raw Device as the NCA Log File

The `logd_path_name` string in `ncalogd.conf` can define a raw device as the place to store the NCA log file. The advantage to using a raw device is that the service can run faster because the overhead in accessing a raw device is less.

The NCA service tests any raw device that is listed in the file to ensure that no file system is in place. This test ensures that no active file systems are accidentally written over.

To prevent this test from finding a file system, run the following command. This command destroys part of the file system on any disk partition that had been configured as a file system. In this example, `/dev/rdsk/c0t0d0s7` is the raw device that has an old file system in place.

```
# dd if=/dev/zero of=/dev/rdsk/c0t0d0s7 bs=1024 count=1
```

After running `dd`, you can then add the raw device to the `ncalogd.conf` file.

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=enabled
logd_path_name="/dev/rdsk/c0t0d0s7"
logd_file_size=1000000
```

Example 2–2 Using Multiple Files for NCA Logging

The `logd_path_name` string in `ncalogd.conf` can define multiple targets as the place to store the NCA log file. The second file is used when the first file is full. The following example shows how to select to write to the `/var/nca/log` file first and then use a raw partition.

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=enabled
logd_path_name="/var/nca/log /dev/rdsk/c0t0d0s7"
logd_file_size=1000000
```

How to Disable Caching of Web Pages

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 turned off, as needed, after NCA has been enabled. See "How to Enable Caching of Web Pages" on page 48 for more information.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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. **Stop logging.**

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

b. **Start logging.**

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

How to Load the Socket Utility Library for NCA

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

In the startup script for the web server, add a line that causes the library to be preloaded. The line should resemble the following:

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

▼ How to Add a New Port to the NCA Service

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Add a new port.**

Add a new port entry to `/etc/nca/ncaport.conf`. This example adds port `8888` on IP address `192.168.84.71`. See `ncaport.conf(4)` for more information.

```
# cat /etc/nca/ncaport.conf
#
# NCA Kernel Module Port Configuration File
#
  .
  .
ncaport=*/80
ncaport=192.168.84.71/8888
```

3 **Start a new web instance.**

An address needs to be in the file that contains the NCA port configurations before a web server can use the address for NCA. If the web server is running, it must be restarted after the new address is defined.

▼ How to Configure an Apache 2.0 Web Server to Use the SSL Kernel Proxy

This procedure should be used to improve the performance of SSL packet process on an Apache 2.0 web server.

Before You Begin The following procedure requires that an Apache 2.0 web server has been installed and configured. The Apache 2.0 web server is included in the release.

To use the SSL kernel proxy, the server private key and the server certificate need to exist in a single file. If only the `SSLCertificateFile` parameter is specified in the `ssl.conf` file, then the specified file can be used directly for kernel SSL. If the `SSLCertificateKeyFile` parameter is also specified, then the certificate file and the private key file need to be combined. One way to combine the certificate and the key file is to run the following command:

```
# cat cert.pem key.pem >cert-and-key.pem
```

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*. The `ksslcfg` command is included in the `Network Security` profile.

2 Stop the web server.

This command will stop the web server on a system in which the server is configured to run using SMF.

```
# svcadm disable svc:/network/http:apache2
```

If the service has not be converted yet, stop the service with this command syntax:
`/usr/apache2/bin/apachectl stop`

3 Determine what parameters to use with the `ksslcfg` command.

All of the options are listed in the `ksslcfg(1M)` man page. The parameters that you must have information for are:

- `key-format` – Used with the `-f` option to define the certificate and key format. For the SSL kernel proxy the value should be either `pem` or `pkcs12`.
- `key-and-certificate-file` – Used with the `-i` option to set the location of the file that stores to server key and the certificate.
- `password-file` – Used with the `-p` option to select the location of the file that includes the password used to encrypt the private key. This password is used to allow unattended reboots. The permissions on the file should be `0400`.
- `proxy-port` – Used with the `-x` option to set the SSL proxy port. Select a different port than the standard port `80`. The web server listens on the SSL proxy port.

- `ssl-port` – Selects the port for the SSL Kernel Proxy to listen on. Normally this is set to `443`.

Note – The `ssl-port` and the `proxy-port` values can not be configured for NCA since these ports are used exclusively by the SSL kernel proxy. Usually, port 80 is used for NCA, port 8443 for the `proxy-port` and 443 for the `ssl-port`.

---

4 Create the service instance.

The `ksslcfg` command to specify the SSL proxy port and associated parameters.

`ksslcfg create -f` *key-format* `-i` *key-and-certificate-file* `-p` *password-file* `-x` *proxy-port ssl-port*

5 Verify that the instance was created properly.

The service state reported by the following command should be "online".

```
# svcs svc:/network/ssl/proxy
```

6 Configure the web server to listen on the SSL proxy port.

Edit the `/etc/apache2/http.conf` file and add a line to define the SSL proxy port. If you use the servers IP address, then the web server will only listen on that interface. The line should look like:

`Listen 0.0.0.0:`*proxy-port*

7 Set an SMF dependency for the web server.

The web server should only be started after the SSL kernel proxy instance. The following commands establish that dependency.

```
# svccfg -s svc:/network/http:apache2
svc:/network/http:apache2> addpg kssl dependency
svc:/network/http:apache2> setprop kssl/entities = fmri:svc:/network/ssl/proxy:kssl-INADDR_ANY-443
svc:/network/http:apache2> setprop kssl/grouping = astring: require_all
svc:/network/http:apache2> setprop kssl/restart_on = astring: refresh
svc:/network/http:apache2> setprop kssl/type = astring: service
svc:/network/http:apache2> end
```

8 Enable the web server.

```
# svcadm enable svc:/network/http:apache2
```

If the service is not started using SMF, use the following command:
`/usr/apache2/bin/apachectl startssl`

Example 2–3 Configuring an Apache 2.0 Web Server to Use the SSL Kernel Proxy

The following command creates an instance using the `pem` key format.

```
# ksslcfg create -f pem -i cert-and-key.pem -p file -x 8443 443
```

▼ How to Configure a Sun Java System Web Server to Use the SSL Kernel Proxy

This procedure should be used to improve the performance of SSL packet process on a Sun Java System Web Server. See the *Sun Java System Web Server 6.1 SP4 Administrator's Guide* for information about this web server.

Before You Begin The following procedure requires that a Sun Java System Web Server has been installed and configured.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*. The `ksslcfg` command is included in the `Network Security` profile.

2 Stop the web server.

Use the administrator web interface to stop the server. See *Starting and Stopping the Server* in the *Sun Java System Web Server 6.1 SP4 Administrator's Guide*for more information.

3 Disable the cryptographic framework's metaslot.

This step is needed to make sure that the metaslot is disabled when the kernel SSL service instance is created.

```
# cryptoadm disable metaslot
```

4 Determine what parameters to use with the `ksslcfg` command.

All of the options are listed in the `ksslcfg`(1M) man page. The parameters that you must have information for are:

- `key-format` – Used with the `-f` option to define the certificate and key format.
- `token-label` – Used with the `-T` option to specify the PKCS#11 token.
- `certificate-label` – Used with the `-C` option to select the label in the certificate object in the PKCS#11 token.
- `password-file` – Used with the `-p` option to select the location of the file that includes the password used to login the user to the PKCS#11 token used by the web server. This password is used to allow unattended reboots. The permissions on the file should be `0400`.
- `proxy-port`– Used with the `-x` option to set the SSL proxy port. Select a different port than the standard port `80`. The web server listens on the SSL proxy port.
- `ssl-port` – Defines the port for the SSL Kernel Proxy to listen on. Normally this value is set to `443`.

Note – The `ssl-port` and the `proxy-port` values can not be configured for NCA since these ports are used exclusively by the SSL kernel proxy. Usually, port 80 is used for NCA, port 8443 for the `proxy-port` and 443 for the `ssl-port`.

5 Create the service instance.

The `ksslcfg` command to specify the SSL proxy port and associated parameters.

```
ksslcfg create -f key-format -T PKCS#11-token -C certificate-label -p password-file -x proxy-port ssl-port
```

6 Enable the cryptographic framework's metaslot.

```
# cryptoadm enable metaslot
```

7 Verify that the instance was created properly.

The service state reported by the following command should be “online”.

```
# svcs svc:/network/ssl/proxy
```

8 Configure the web server to listen on the SSL proxy port.

See *Adding and Editing Listen Sockets* in the *Sun Java System Web Server 6.1 SP4 Administrator's Guide* for more information.

9 Start the web server.

Example 2–4 Configuring a Sun Java System Web Server to Use the SSL Kernel Proxy

The following command creates an instance using the `pkcs11` key format.

```
# ksslcfg create -f pkcs11 -T "Sun Software PKCS#11 softtoken" -C "Server-Cert" -p file -x 8443 443
```

Using the SSL Kernel Proxy in Zones

The SSL Kernel Proxy works in zones with the following limitations:

- All of the kernel SSL administration must be done from the global zone. The global zone administrator needs access to the local zone certificate and key files. The local zone web server can be started once the service instance is configured using the `ksslcfg` command in the global zone.
- A specific host name or IP address must be specified when running the `ksslcfg` command to configure the instance. In particular, the instance can not use INADDR_ANY.

EXAMPLE 2–5 Configuring an Apache Web Server in a Local Zone to Use the SSL Kernel Proxy

In the local zone, first stop the web server. In the global zone do all of the steps to configure the service. To create a instance for a local zone called `apache-zone`, use the following command:

EXAMPLE 2–5 Configuring an Apache Web Server in a Local Zone to Use the SSL Kernel Proxy *(Continued)*

```
# ksslcfg create -f pem -i /zone/apache-zone/root/keypair.pem -p /zone/apache-zone/root/pass \
    -x 8443 apache-zone 443
```

In the local zone, run the following command to enable the service instance:

```
# svcadm enable svc:/network/http:apache2
```

Caching Web Pages (Reference)

The following sections cover the files and the 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 the files are binary. The following table lists all of the files.

TABLE 2–1 NCA Files

File Name	Function
`/dev/nca`	The path name for the NCA device.
`/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. This script is run when a server is booted.
`/etc/init.d/ncalogd`	Script that starts NCA logging. This script 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.

TABLE 2–1 NCA Files *(Continued)*

File Name	Function
`/etc/nca/ncaport.conf`	File that lists the IP addresses and the ports for NCA. See the `ncaport.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.
`/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.
`/var/run/nca_httpd_1.door`	The door path name.

NCA Architecture

The NCA feature includes the following components.

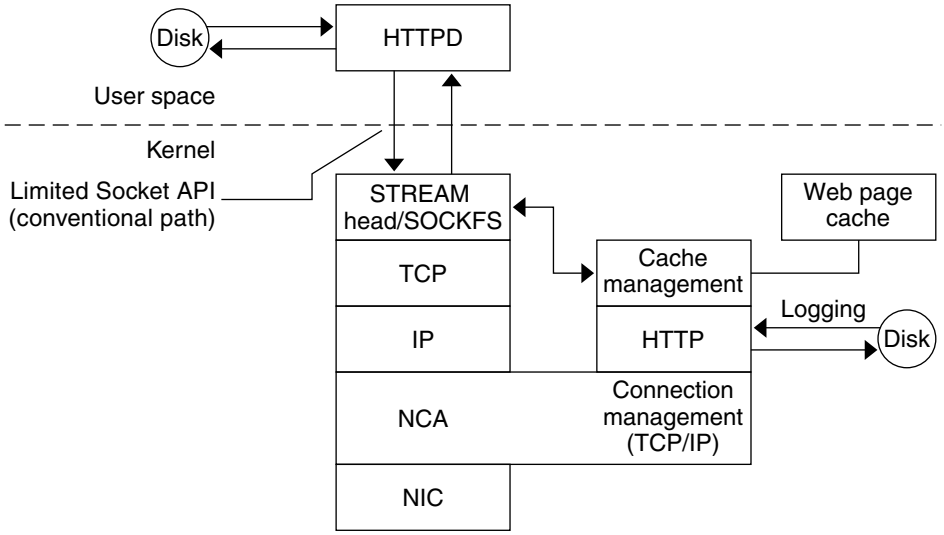
- Kernel module, `ncakmod`
- Web server, `httpd`

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. The family type is 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.

FIGURE 2–1 Data Flow With the NCA Service



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 response, the page is cached or not. Then the page is returned to the client. If the Pragma: No-cache header is included in the HTTP request, the page is not cached.

◆ ◆ ◆ 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. 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.

Remember the following when using NTP in your network:

- The `xntpd` daemon uses minimal system resources.
- An NTP client synchronizes automatically with an NTP server when it boots. If the client becomes unsynchronized, the client resynchronizes again when the client contacts a time server.

Another way to synchronize clocks is to run `rdate` while 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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.**

   ```
   # svcadm enable network/ntp
   ```

How to Set Up an NTP Client

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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.**

   ```
   # svcadm enable network/ntp
   ```

Using Other Time-Related Commands (Tasks)

The following procedure can be used to update the current time when ever needed, without having to setup NTP.

▼ How to Synchronize Date and Time From Another System

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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

`# rdate` *another-system*

another-system Name of the 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 3–1 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.

TABLE 3–1 NTP Files *(Continued)*

File Name	Function
`/etc/inet/ntp.client`	Sample configuration file for NTP clients.
`/etc/inet/ntp.server`	Sample configuration file for NTP servers.
`/etc/inet/ntp.keys`	Contains the NTP authentication keys.
`/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.
`/var/ntp/ntp.drift`	Sets the initial frequency offset on NTP servers.

PART II

Accessing Network File Systems Topics

This section provides overview, task, and reference information for the NFS service.

◆ ◆ ◆ CHAPTER 4

Managing Network File Systems (Overview)

This chapter provides an overview of the NFS service, which can be used to access file systems over the network. The chapter includes a discussion of the concepts necessary to understand the NFS service and a description of the latest features in NFS and autofs.

Note – If your system has zones enabled and you want to use this feature in a non-global zone, see *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones* for more information.

What's New With the NFS Service

This section provides information about new features in releases of the Solaris OS.

Changes in the Solaris 10 11/06 Release

The Solaris 10 11/06 release provides support for a file system monitoring tool. See the following:

- "`fsstat` Command" on page 147 for a description and examples
- `fsstat`(1M) man page for more information

Additionally, this Guide provides a more detailed description of the `nfsmapid` daemon. For information about `nfsmapid`, see the following:

- "`nfsmapid` Daemon" on page 138
- `nfsmapid`(1M) man page

For a complete list of new features, see *Oracle Solaris 10 8/11 What's New*.

Changes in the Solaris 10 Release

Starting in the Solaris 10 release, NFS version 4 is the default. For information about features in NFS version 4 and other changes, refer to the following:

- "NFS Version 4 Protocol" on page 72
- "`/etc/default/autofs` File" on page 130
- "Keywords for the `/etc/default/nfs` File" on page 131
- "`lockd` Daemon" on page 135
- "`nfs4cbd` Daemon" on page 136
- "`nfsmapid` Daemon" on page 138
- "`mount` Options for NFS File Systems" on page 148
- "NFS Over RDMA" on page 167
- "Version Negotiation in NFS" on page 169
- "Features in NFS Version 4" on page 170
- "How Autofs Selects the Nearest Read-Only Files for Clients (Multiple Locations)" on page 199

Also, see the following:

- "Setting Up NFS Services" on page 88 for task information
- *Oracle Solaris 10 8/11 What's New* for a complete list of new features

Additionally, the NFS service is managed by the Service Management Facility. Administrative actions on this service, such as enabling, disabling, or restarting, can be performed by using the `svcadm` command. The service's status can be queried by using the `svcs` command. For more information about the Service Management Facility, refer to the `smf`(5) man page and Chapter 18, "Managing Services (Overview)," in *System Administration Guide: Basic Administration*.

NFS Terminology

This section presents some of the basic terminology that must be understood to work with the NFS service. Expanded coverage of the NFS service is included in Chapter 6, "Accessing Network File Systems (Reference)."

NFS Servers and Clients

The terms *client* and *server* are used to describe the roles that a computer assumes when sharing file systems. Computers that share their file systems over a network are acting as servers. The computers that are accessing the file systems are said to be clients. The NFS service enables any computer to access any other computer's file systems. At the same time, the NFS service provides access to its own file systems. A computer can assume the role of client, server, or both client and server at any particular 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, the client 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 resembles a local mount. Users type commands as if the file systems were local. See "Mounting File Systems" on page 83 for information about tasks that mount file systems.

After a file system has been shared on a server through an NFS operation, the file system can be accessed from a client. You can mount an NFS file system automatically with autofs. See "Automatic File System Sharing" on page 80 and "Task Overview for Autofs Administration" on page 98 for tasks that involve 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 a file hierarchy 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* refers to a file or file hierarchy that can be shared and be mounted with NFS.

About the NFS Service

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

The NFS environment can be implemented on different operating systems because NFS 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 model means that file system operations such as reading and writing function as though the operations are accessing a local file.

The NFS service has the following benefits:

- Enables multiple computers to use the same files so that 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 of 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. All other systems access the files across the network. Under NFS operation, remote file systems are almost indistinguishable from local file systems.

About Autofs

File systems that are shared through the NFS service can be mounted by 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 as necessary.

Whenever a client computer that is running `automountd` tries to access a remote file or remote directory, the daemon mounts the remote file system. This remote file system remains mounted for as long as needed. If the remote file system is not accessed for a certain period of time, the file system 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 do not need to use the `mount` and `umount` commands. The autofs service mounts and unmounts file systems as required without any intervention by the user.

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

"Task Overview for Autofs Administration" on page 98 and "How Autofs Works" on page 195 give more specific information about the autofs service.

Features of the NFS Service

This section describes the important features that are included in the NFS service.

NFS Version 2 Protocol

Version 2 was the first version of the NFS protocol in wide use. Version 2 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 Protocol

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.

Unlike the NFS version 2 protocol, the NFS version 3 protocol can handle files that are larger than 2 Gbytes. The previous limitation has been removed. See "NFS Large File Support" on page 74.

The NFS version 3 protocol enables safe asynchronous writes on the server, which improve 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.

Many 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. Version 2 generated a "write error" message or a "read error" message if users tried to copy a remote file without the appropriate 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 protocol removed the 8-Kbyte transfer size limit. Clients and servers could negotiate whatever transfer size the clients and servers support, rather than conform to the 8-Kbyte limit that version 2 imposed. Note that in the Solaris 2.5 implementation, the protocol defaulted to a 32-Kbyte transfer size. Starting in the Solaris 10 release, restrictions on wire transfer sizes are relaxed. The transfer size is based on the capabilities of the underlying transport.

NFS Version 4 Protocol

NFS version 4 has features that are not available in the previous versions.

The NFS version 4 protocol represents the user ID and the group ID as strings. `nfsmapid` is used by the client and the server to do the following:

- To map these version 4 ID strings to a local numeric ID
- To map the local numeric IDs to version 4 ID strings

For more information, refer to "`nfsmapid` Daemon" on page 138.

Note that in NFS version 4, the ID mapper, `nfsmapid`, is used to map user or group IDs in ACL entries on a server to user or group IDs in ACL entries on a client. The reverse is also true. For more information, see "ACLs and `nfsmapid` in NFS Version 4" on page 177.

With NFS version 4, when you unshare a file system, all the state for any open files or file locks in that file system is destroyed. In NFS version 3 the server maintained any locks that the clients had obtained before the file system was unshared. For more information, refer to "Unsharing and Resharing a File System in NFS Version 4" on page 170.

NFS version 4 servers use a pseudo file system to provide clients with access to exported objects on the server. Prior to NFS version 4 a pseudo file system did not exist. For more information, refer to "File-System Namespace in NFS Version 4" on page 170.

In NFS version 2 and version 3 the server returned persistent file handles. NFS version 4 supports volatile file handles. For more information, refer to "Volatile File Handles in NFS Version 4" on page 172.

Delegation, a technique by which the server delegates the management of a file to a client, is supported on both the client and the server. For example, the server could grant either a read delegation or a write delegation to a client. For more information, refer to "Delegation in NFS Version 4" on page 175.

Starting in the Solaris 10 release, NFS version 4 does not support the LIPKEY/SPKM security flavor.

Also, NFS version 4 does not use the following daemons:

- `mountd`
- `nfslogd`
- `statd`

For a complete list of the features in NFS version 4, refer to "Features in NFS Version 4" on page 170.

For procedural information that is related to using NFS version 4, refer to "Setting Up NFS Services" on page 88.

Controlling NFS Versions

The `/etc/default/nfs` file has keywords to control the NFS protocols that are used by both the client and the server. For example, you can use keywords to manage version negotiation. For more information, refer to "Keywords for the `/etc/default/nfs` File" on page 131 or the `nfs(4)` man page.

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.

The NFS Version 2 and Version 3 protocols support the old POSIX-draft style ACLs. POSIX-draft ACLs are natively supported by UFS. See "Using Access Control Lists to Protect UFS Files" in *System Administration Guide: Security Services* for more information about UFS ACLs.

The NFS Version 4 protocol supports the new NFSv4 style ACLs. NFSv4 ACLs are natively supported by ZFS. For full featured NFSv4 ACL functionality, ZFS must be used as the underlying file system on the NFSv4 server. The NFSv4 ACLs have a rich set of inheritance properties, as well as a set of permission bits beyond the standard read, write and execute. See Chapter 8, "Using ACLs to Protect Oracle Solaris ZFS Files," in *Oracle Solaris ZFS Administration Guide* for an overview of the new ACLs. For more information about support for ACLs in NFS version 4, see "ACLs and `nfsmapid` in NFS Version 4" on page 177.

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. TCP helps performance on slow networks and wide area networks. TCP also provides congestion control and error recovery. NFS over TCP works with version 2, version 3, and version 4. Prior to the Solaris 2.5 release, the default NFS protocol was User Datagram Protocol (UDP).

NFS Over UDP

Starting in the Solaris 10 release, the NFS client no longer uses an excessive number of UDP ports. Previously, NFS transfers over UDP used a separate UDP port for each outstanding request. Now, by default, the NFS client uses only one UDP reserved port. However, this support is configurable. If the use of more simultaneous ports would increase system performance through increased scalability, then the system can be configured to use more

ports. This capability also mirrors the NFS over TCP support, which has had this kind of configurability since its inception. For more information, refer to the *Oracle Solaris Tunable Parameters Reference Manual*.

Note – NFS version 4 does not use UDP. If you mount a file system with the `proto=udp` option, then NFS version 3 is used instead of version 4.

Overview of NFS Over RDMA

The Solaris 10 release includes the Remote Direct Memory Access (RDMA) protocol, which is a technology for memory-to-memory transfer of data over high speed networks. Specifically, RDMA provides remote data transfer directly to and from memory without CPU intervention. To provide this capability, RDMA combines the interconnect I/O technology of InfiniBand-on-SPARC platforms with the Solaris Operating System. For more information, refer to "NFS Over RDMA" on page 167.

Network Lock Manager and NFS

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

Note – The Network Lock Manager is used only for NFS version 2 and version 3 mounts. File locking is built into the NFS version 4 protocol.

NFS Large File Support

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

NFS Client Failover

Dynamic failover of read-only file systems was added in the Solaris 2.6 release. Failover 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 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 86 and "Client-Side Failover" on page 181 for more information.

Kerberos Support for the NFS Service

Support for Kerberos V4 clients was included in the Solaris 2.0 release. In the 2.6 release, the `mount` and `share` commands were altered to support NFS version 3 mounts that use Kerberos V5 authentication. Also, the `share` command was changed to enable multiple authentication flavors for different clients. See "RPCSEC_GSS Security Flavor" on page 75 for more information about changes that involve security flavors. See "Configuring Kerberos 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. This capability was provided by 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. Additionally, 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 76 for a description of more changes that are related to the WebNFS service. See "WebNFS Administration Tasks" on page 96 for more task information.

Note – The NFS version 4 protocol is preferred over the WebNFS service. NFS version 4 fully integrates all the security negotiation that was added to the MOUNT protocol and the WebNFS service.

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 enabling support of multiple security mechanisms. See "Kerberos Support for the NFS Service" on page 75 for more information about support of Kerberos V5 authentication. See *Developer's Guide to Oracle Solaris Security* for more information about GSS-API.

Solaris 7 Extensions for NFS Mounting

The Solaris 7 release includes extensions to the `mount` command and `automountd` command. The extensions enable the mount request to use the public file handle instead of the MOUNT protocol. The MOUNT protocol is the same access method that the WebNFS service uses. By circumventing the MOUNT protocol, the mount can occur through a firewall. Additionally, because fewer transactions need to occur between the server and the client, the mount should occur faster.

The extensions also enable 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 75 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 in the Solaris 8 release. This protocol provides the ability to use secure transactions when using the WebNFS service. See "How WebNFS Security Negotiation Works" on page 185 for more information.

NFS Server Logging

In the Solaris 8 release, NFS server logging enables an NFS server to provide a record of file operations that have been performed on its file systems. The record includes information about which file was accessed, when the file was accessed, and who accessed the file. 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 82 for more information.

Note – NFS version 4 does not support server logging.

Autofs Features

Autofs works with file systems that are specified in the local namespace. 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 enables 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 the file systems were hierarchically related. Now, only the top file system is mounted. Other file systems that are related to this mount point are mounted when needed.

The autofs service supports browsability of indirect maps. This support enables a user to see which directories could be mounted, without having to actually mount each file system. 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`. See "Disabling Autofs Browsability" on page 111 for more information.

◆ ◆ ◆ CHAPTER 5

Network File System Administration (Tasks)

This chapter provides information about how to perform such NFS administration tasks as setting up NFS services, adding new file systems to share, and mounting file systems. The chapter also covers the use of the Secure NFS system and the use of WebNFS functionality. The last part of the chapter includes troubleshooting procedures and a list of some 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 instance you might be responsible for determining these configuration items:

- Which computers 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. So, a computer can be used to share local file systems with remote computers and to mount remote file systems.

Note – If your system has zones enabled and you want to use this feature in a non-global zone, see *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones* for more information.

Automatic File System Sharing

Servers provide access to their file systems by sharing the file systems over the NFS environment. You specify which file systems are to be shared with the `share` command or with 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. This file also controls which clients can mount a file system. You can modify `dfstab` to add or delete a file system or change the way sharing occurs. Just edit the file with any text editor that is supported (such as `vi`). The next time that 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 that you type at the command-line prompt to share the file system. The `share` command is located in `/usr/sbin`.

TABLE 5–1 File-System Sharing Task Map

Task	Description	For Instructions
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 81
Enable WebNFS	Steps to configure a server so that users can access files by using WebNFS	"How to Enable WebNFS Access" on page 81
Enable 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 82

How to Set Up Automatic File-System Sharing

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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

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

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

   See the `dfstab(4)` man page for a description of `/etc/dfs/dfstab` and the `share_nfs(1M)` man page for a complete list of options.

3. **Share the file system.**

   After the entry is in `/etc/dfs/dfstab`, you can share the file system by either rebooting the system or by using the `shareall` command.

   ```
   # shareall
   ```

4. **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  ""
   ```

See Also The next step is to set up your autofs maps so that clients can access the file systems that you have shared on the server. See "Task Overview for Autofs Administration" on page 98.

How to Enable WebNFS Access

Starting with the Solaris 2.6 release, by default all file systems that are available for NFS mounting are automatically available for WebNFS access. The only condition that requires the use of this procedure is one of the following:

- To allow NFS mounting on a server that does not already allow NFS mounting
- To reset the public file handle to shorten NFS URLs by using the `public` option
- To force the loading of a specific HTML file by using the `index` option

See "Planning for WebNFS Access" on page 96 for a list of issues to consider before starting the WebNFS service.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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

   Edit `/etc/dfs/dfstab`. Add one entry to the file for every file system. The `public` and `index` tags that are shown in the following example are optional.

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

   See the `dfstab(4)` man page for a description of `/etc/dfs/dfstab` and the `share_nfs(1M)` man page for a complete list of options.

3. **Share the file system.**

   After the entry is in `/etc/dfs/dfstab`, you can share the file system by either rebooting the system or by using the `shareall` command.

   ```
   # shareall
   ```

4. **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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **(Optional) Change file system configuration settings.**

   In `/etc/nfs/nfslog.conf`, you can change the settings in one of two ways. You can edit the default settings for all file systems by changing the data that is associated with the `global` tag. Alternately, 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 by using NFS server logging.**

   Edit `/etc/dfs/dfstab`. Add one entry to the file for the file system on which you are enabling NFS server logging. The tag that is 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 dfstab(4) man page for a description of /etc/dfs/dfstab and the share_nfs(1M) man page for a complete list of options.

4 **Share the file system.**

After the entry is in /etc/dfs/dfstab, you can share the file system by either rebooting the system or by using the shareall command.

```
# shareall
```

5 **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  ""
```

6 **Check if nfslogd, the NFS log daemon, is running.**

```
# ps -ef | grep nfslogd
```

7 **(Optional) Start nfslogd, if it is not running already.**

- **(Optional) If /etc/nfs/nfslogtab is present, start the NFS log daemon by typing the following:**

  ```
  # svcadm restart network/nfs/server:default
  ```

- **(Optional) If /etc/nfs/nfslogtab is not present, run any of the share commands to create the file and then start the daemon.**

  ```
  # shareall
  # svcadm restart network/nfs/server:default
  ```

Mounting File Systems

You can mount file systems in several ways. File systems 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. However, many situations require a combination of all three methods. Additionally, several ways of enabling or disabling processes exist, depending on the options you use when mounting the file system. See the following table for a complete list of the tasks that are associated with file system mounting.

TABLE 5–2 Task Map for Mounting File Systems

Task	Description	For Instructions
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 84.

TABLE 5–2 Task Map for Mounting File Systems *(Continued)*

Task	Description	For Instructions
Mount a file system by 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 85.
Mount with the automounter	Steps to access a file system on demand without using the command line.	"Mounting With the Automounter" on page 85.
Prevent 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 86.
Start 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 86.
Disable 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 87.
Provide 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 87.
Mount a file system by using an NFS URL	Steps to allow access to a file system by 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 88.

▼ 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. This procedure must be completed on every client that should have access to remote file systems.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 that also have NFS client `vfstab` entries must always specify the `bg` option to avoid a system hang during reboot. For more information, see "`mount` Options for NFS File Systems" on page 148.

Example 5–1 Entry in the Client's `vfstab` File

You want a client machine to mount the `/var/mail` directory from the server `wasp`. You want 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 performed to test a new mount point. This type of mount allows for temporary access to a file system that is not available through the automounter.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Mount the file system.**

   Type the following command:

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

   In this instance, 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 – All versions of the `mount` command do not warn about invalid options. The command silently ignores any options that cannot be interpreted. To prevent unexpected behavior, ensure that you verify all of the options that were used.

Mounting With the Automounter

"Task Overview for Autofs Administration" on page 98 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, type the following:

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

Because the automounter allows all users to mount file systems, `root` access is not required. The automounter also provides for automatic unmounting of file systems, so you do not 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, you might need to disable the ability to create large files.

Note – Versions prior to the 2.6 release of the Solaris release cannot use large files. If the clients need to access large files, check that the clients of the NFS server are running, at minimum, the 2.6 release.

---

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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

   For example:

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

   If large files are on the file system, you must remove or move these files 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 by 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 by using `nolargefiles`.**

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

   You can mount from the command line, but to make the option more permanent, add an entry that resembles 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **On the NFS client, mount the file system by using the `ro` option.**

   You can mount from the command line, through the automounter, or by adding an entry to `/etc/vfstab` that resembles the following:

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

   This syntax has been allowed by the automounter. However, 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 by using a command line or in a `vfstab` entry. Mixing servers that support NFS version 2, version 3, or version 4 protocols can only be performed with autofs. In autofs, the best subset of version 2, version 3, or version 4 servers is used.

▼ How to Disable Mount Access for One Client

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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

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

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

   For additional information about access lists, see "Setting Access Lists With the `share` Command" on page 158. For a description of `/etc/dfs/dfstab`, see `dfstab(4)`.

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

To access file systems through a firewall, use the following procedure.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Manually mount the file system by using a command such as the following:**

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

In this example, the file system `/export/share/local` is mounted on the local client by 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 fails.

Note – This procedure requires that the file system on the NFS server be shared by using the `public` option. Additionally, any firewalls between the client and the server must allow TCP connections on port `2049`. 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **(Optional) If you are using NFS version 2 or version 3, manually mount the file system by using a command such as the following:**

```
# 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` by using NFS port number `3000`. The port number is not required and by default the standard NFS port number of `2049` is used. You can choose to include the `public` option with an NFS URL. Without the `public` option, the MOUNT protocol is used if the public file handle is not supported by the server. The `public` option forces the use of the public file handle, and the mount fails if the public file handle is not supported.

3 **(Optional) If you are using NFS version 4, manually mount the file system by using a command such as the following:**

```
# mount -F nfs -o vers=4 nfs://bee:3000/export/share/local /mnt
```

Setting Up NFS Services

This section describes some of the tasks that are necessary to do the following:

- Start and stop the NFS server
- Start and stop the automounter
- Select a different version of NFS

Note – Starting in the Solaris 10 release, NFS version 4 is the default.

TABLE 5–3 Task Map for NFS Services

Task	Description	For Instructions
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 89
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 90
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 90
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 90
Select a different version of NFS on the server	Steps to select a different version of NFS on the server. If you choose not to use NFS version 4, use this procedure.	"How to Select Different Versions of NFS on a Server" on page 91
Select a different version of NFS on the client	Steps to select a different version of NFS on the client by modifying the `/etc/default/nfs` file. If you choose not to use NFS version 4, use this procedure.	"How to Select Different Versions of NFS on a Client by Modifying the `/etc/default/nfs` File" on page 92
	Alternate steps to select a different version of NFS on the client by using the command line. If you choose not to use NFS version 4, use this alternate procedure.	"How to Use the `mount` Command to Select Different Versions of NFS on a Client" on page 93

▼ How to Start the NFS Services

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Enable the NFS service on the server.

Type the following command.

```
# svcadm enable network/nfs/server
```

This command enables the NFS service.

Note – The NFS server starts automatically when you boot the system. Additionally, any time after the system has been booted, the NFS service daemons can be automatically enabled by sharing the NFS file system. See "How to Set Up Automatic File-System Sharing" on page 81.

How to Stop the NFS Services

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Disable the NFS service on the server.**

   Type the following command.

   ```
   # svcadm disable network/nfs/server
   ```

How to Start the Automounter

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Enable the autofs daemon.**

   Type the following command:

   ```
   # svcadm enable system/filesystem/autofs
   ```

How to Stop the Automounter

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Disable the autofs daemon.**

   Type the following command:

   ```
   # svcadm disable system/filesystem/autofs
   ```

How to Select Different Versions of NFS on a Server

If you choose not to use NFS version 4, use this procedure.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit the `/etc/default/nfs` file.**

   For example, if you want the server to provide only version 3, set the values for both NFS_SERVER_VERSMAX and NFS_SERVER_VERSMIN to 3. For a list of keywords and their values, refer to "Keywords for the `/etc/default/nfs` File" on page 131.

   ```
   NFS_SERVER_VERSMAX=value
   NFS_SERVER_VERSMIN=value
   ```

   value Provide the version number.

   Note – By default, these lines are commented. Remember to remove the pound (#) sign, also.

3. **(Optional) If you want to disable server delegation, include this line in the `/etc/default/nfs` file.**

   ```
   NFS_SERVER_DELEGATION=off
   ```

   Note – In NFS version 4, server delegation is enabled by default. For more information, see "Delegation in NFS Version 4" on page 175.

4. **(Optional) If you want to set a common domain for clients and servers, include this line in the `/etc/default/nfs` file.**

   ```
   NFSMAPID_DOMAIN=my.comany.com
   ```

   my.comany.com Provide the common domain

   For more information, refer to "`nfsmapid` Daemon" on page 138.

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

   Type the following command:

   ```
   # svcs network/nfs/server
   ```

   This command reports whether the NFS server service is online or disabled.

6 **(Optional) If necessary, disable the NFS service.**

If you discovered from the previous step that the NFS service is online, type the following command to disable the service.

```
# svcadm disable network/nfs/server
```

Note – If you need to configure your NFS service, refer to "How to Set Up Automatic File-System Sharing" on page 81.

---

7 **Enable the NFS service.**

Type the following command to enable the service.

```
# svcadm enable network/nfs/server
```

See Also "Version Negotiation in NFS" on page 169

▼ How to Select Different Versions of NFS on a Client by Modifying the `/etc/default/nfs` File

The following procedure shows you how to control which version of NFS is used on the client by modifying the `/etc/default/nfs` file. If you prefer to use the command line, refer to "How to Use the `mount` Command to Select Different Versions of NFS on a Client" on page 93.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Edit the `/etc/default/nfs` file.**

For example, if you want only version 3 on the client, set the values for both NFS_CLIENT_VERSMAX and NFS_CLIENT_VERSMIN to 3. For a list of keywords and their values, refer to "Keywords for the `/etc/default/nfs` File" on page 131.

```
NFS_CLIENT_VERSMAX=value
NFS_CLIENT_VERSMIN=value
```

value Provide the version number.

Note – By default, these lines are commented. Remember to remove the pound (#) sign, also.

---

3 **Mount NFS on the client.**

Type the following command:

```
# mount server-name:/share-point /local-dir
```

server-name Provide the name of the server.

/share-point Provide the path of the remote directory to be shared.

/local-dir Provide the path of the local mount point.

See Also "Version Negotiation in NFS" on page 169

How to Use the `mount` Command to Select Different Versions of NFS on a Client

The following procedure shows you how to use the `mount` command to control which version of NFS is used on a client for a particular mount. If you prefer to modify the NFS version for all file systems mounted by the client, see "How to Select Different Versions of NFS on a Client by Modifying the `/etc/default/nfs` File" on page 92.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Mount the desired version of NFS on the client.**

Type the following command:

```
# mount -o vers=value server-name:/share-point /local-dir
```

value Provide the version number.

server-name Provide the name of the server.

/share-point Provide the path of the remote directory to be shared.

/local-dir Provide the path of the local mount point.

Note – This command uses the NFS protocol to mount the remote directory and overrides the client settings in the `/etc/default/nfs` file.

See Also "Version Negotiation in NFS" on page 169

Administering the Secure NFS System

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

Kerberos V5 authentication is supported by the NFS service. Chapter 21, "Introduction to the Kerberos Service," in *System Administration Guide: Security Services* discusses the Kerberos service.

You can also configure the Secure NFS environment to use Diffie-Hellman authentication. Chapter 16, "Using Authentication Services (Tasks)," in *System Administration Guide: Security Services* discusses this authentication 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 (DNS, NIS, and LDAP)* if you are using NIS+ as your name service.

2 **Establish public keys and secret keys for your clients' users by using the `newkey` or `nisaddcred` command. Have each user establish his or her own secure RPC password by 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 keys and secret keys have been generated, the public keys 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 of 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 is not 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 situation, `keylogin` is not required. If the passwords are different, the users have to log in, and then run `keylogin`. You still need to use the `keylogin -r` command as `root` to store the decrypted secret key in `/etc/.rootkey`.

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

---

6 Update mount options for the file system.

For Diffie-Hellman authentication, edit the `/etc/dfs/dfstab` file and add the `sec=dh` option to the appropriate entries.

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

See the `dfstab(4)` man page for a description of `/etc/dfs/dfstab`.

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 – Releases through Solaris 2.5 have a limitation. If a client does not securely mount a shared file system that is secure, users have access as `nobody` rather than as themselves. For subsequent releases that use version 2, 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=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 the keys for `root`. If you do delete `/etc/.rootkey`, you can always type the following:

```
# keylogin -r
```

WebNFS Administration Tasks

This section provides instructions for administering the WebNFS system. Related tasks follow.

TABLE 5–4 Task Map for WebNFS Administration

Task	Description	For Instructions
Plan for WebNFS	Issues to consider before enabling the WebNFS service.	"Planning for WebNFS Access" on page 96
Enable WebNFS	Steps to enable mounting of an NFS file system by using the WebNFS protocol.	"How to Enable WebNFS Access" on page 81
Enable WebNFS through a firewall	Steps to allow access to files through a firewall by using the WebNFS protocol.	"How to Enable WebNFS Access Through a Firewall" on page 98
Browse by using an NFS URL	Instructions for using an NFS URL within a web browser.	"How to Browse Using an NFS URL" on page 97
Use 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 111
Use an NFS URL with autofs	Steps to add an NFS URL to the automounter maps.	"How to Use NFS URLs With Autofs" on page 111
Provide 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 87
Mount a file system by using an NFS URL	Steps to allow access to a file system by 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 88

Planning for WebNFS Access

To use WebNFS, you first need an application that is capable of running and loading an NFS URL (for example, `nfs://server/path`). The next step is to choose the file system that can be exported for WebNFS access. If the application is web browsing, often the document root for the web server is used. You need to consider several factors 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 server's root file system. The root file system does not allow web access unless the root file system is shared.
2. The WebNFS environment enables users who already have mount privileges to access files through a browser. This capability is enabled regardless of whether the file system is exported by using the `public` option. Because users already have access to these files through the NFS setup, this access should not create any additional security risk. You only need to share a file system by 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. Some examples are 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. Otherwise, you can list 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 provide the correct level of access.

   You need to consider additional factors if both NFS and HTTP URLs are to be used to access one web site. These factors are described in "WebNFS Limitations With Web Browser Use" on page 186.

How to Browse Using an NFS URL

Browsers that are capable of supporting the WebNFS service should provide access to an NFS URL that resembles the following:

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

server	Name of the file server
port	Port number to use (`2049`, default value)
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. The exception occurs when a URL that includes a different service type is loaded. After you use an NFS URL, a reference to an HTTP URL might be loaded. If such a reference is loaded, subsequent pages are loaded by using the HTTP protocol instead of the NFS protocol.

How to Enable 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.

Task Overview for Autofs Administration

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 – Starting in the Solaris 10 release, you can also use the `/etc/default/autofs` file to configure your autofs environment. For task information, refer to "Using the `/etc/default/autofs` File to Configure Your Autofs Environment" on page 100.

Task Map for Autofs Administration

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

TABLE 5–5 Task Map for Autofs Administration

Task	Description	For Instructions
Start autofs	Start the automount service without having to reboot the system	"How to Start the Automounter" on page 90
Stop autofs	Stop the automount service without disabling other network services	"How to Stop the Automounter" on page 90
Configure your autofs environment by using the `/etc/default/autofs` file	Assign values to keywords in the `/etc/default/autofs` file	"Using the `/etc/default/autofs` File to Configure Your Autofs Environment" on page 100

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

Task	Description	For Instructions
Access file systems by using autofs	Access file systems by using the automount service	"Mounting With the Automounter" on page 85
Modify 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 102
	Steps to modify an indirect map, which should be used for most maps	"How to Modify Indirect Maps" on page 102
	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 102
Modify 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 104
	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 104
	Steps to use autofs to access a CacheFS file system	"How to Access NFS File Systems by Using CacheFS" on page 105
Using `/home`	Example of how to set up a common `/home` map	"Setting Up a Common View of `/home`" on page 106
	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 106
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 107
	Steps to set up an autofs map that supports different client architectures	"How to Set Up Different Architectures to Access a Shared Namespace" on page 109
	Steps to set up an autofs map that supports different operating systems	"How to Support Incompatible Client Operating System Versions" on page 110
Replicate file systems with autofs	Provide access to file systems that fail over	"How to Replicate Shared Files Across Several Servers" on page 110
Using security restrictions with autofs	Provide access to file systems while restricting remote `root` access to the files	"How to Apply Autofs Security Restrictions" on page 110
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 111
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 111

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

Task	Description	For Instructions
Disable autofs browsability	Steps to disable browsability 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 112
	Steps to disable browsability so that autofs mount points are not automatically populated on all clients	"How to Disable Autofs Browsability for All Clients" on page 112
	Steps to disable browsability so that a specific autofs mount point is not automatically populated on a client	"How to Disable Autofs Browsability on a Selected File System" on page 112

Using the `/etc/default/autofs` File to Configure Your Autofs Environment

Starting in the Solaris 10 release, you can use the `/etc/default/autofs` file to configure your autofs environment. Specifically, this file provides an additional way to configure your autofs commands and autofs daemons. The same specifications you would make on the command line can be made in this configuration file. You can make your specifications by providing values to keywords. For more information, refer to "`/etc/default/autofs` File" on page 130.

The following procedure shows you how to use the `/etc/default/autofs` file.

▼ How to Configure Your Autofs Environment Using the `/etc/default/autofs` File

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add or modify an entry in the `/etc/default/autofs` file.

For example, if you want to turn off browsing for all autofs mount points, you could add the following line.

```
AUTOMOUNTD_NOBROWSE=ON
```

This keyword is the equivalent of the `-n` argument for `automountd`. For a list of keywords, refer to "`/etc/default/autofs` File" on page 130.

3 Restart the autofs daemon.

Type the following command:

```
# svcadm restart system/filesystem/autofs
```

Administrative Tasks Involving Maps

The following tables describe several of the factors you need to be aware of when administering autofs maps. Your choice of map and name service affect the mechanism that you need to use to make changes to the autofs maps.

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

TABLE 5–6 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 that are based on your name service.

TABLE 5–7 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. By running the command, you make the change effective. However, if you have modified an existing entry, you do not need to run the `automount` command for the change to become effective.

TABLE 5–8 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. **Log in as a user who has permissions to change the maps.**
2. **Using the `nistbladm` command, make your changes to the master map.**

   See the *System Administration Guide: Naming and Directory Services (NIS+)*.
3. **For each client, become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.
4. **For each client, run the `automount` command to ensure that your changes become effective.**
5. **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 that the `automount` command gathers information from the master map whenever it is run.

▼ How to Modify Indirect Maps

1. **Log in as a user who has permissions to change the maps.**
2. **Using the `nistbladm` command, make your changes to the indirect map.**

   See the *System Administration Guide: Naming and Directory Services (NIS+)*. Note that the change becomes effective the next time that the map is used, which is the next time a mount is performed.

▼ How to Modify Direct Maps

1. **Log in as a user who has permissions to change the maps.**
2. **Using the `nistbladm` command, add or delete your changes to the direct map.**

   See the *System Administration Guide: Naming and Directory Services (NIS+)*.

3 **Notify your users of the changes.**

Notification is required so that the users can run the `automount` command as superuser on their own computers, if necessary.

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 becomes effective immediately when you try to access `/usr/src`. If `/usr/src` is mounted now, you can wait until the auto-unmounting occurs, then access the file.

Note – Use indirect maps whenever possible. Indirect maps are easier to construct and less demanding on the computers' file systems. Also, indirect maps do not occupy as much space in the mount table as direct maps.

Avoiding Mount-Point Conflicts

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

You need to mount the partition in some other location, for example, on `/export/src`. You then need an entry in `/etc/vfstab` such as the following:

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

You also need this entry in `auto_src`:

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

`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 by using the Volume Manager. The following examples show how this mounting could be accomplished 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 the file system from the map. If you plan 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Update the autofs map.**

   Add an entry for the CD-ROM file system, which should resemble the following:

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

   The CD-ROM device that you intend to mount must appear as a name that follows the colon.

▼ How to Access PC-DOS Data Diskettes With Autofs

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

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Update the autofs map.**

   Add an entry for the diskette file system such as the following:

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

Accessing NFS File Systems Using CacheFS

The cache file system (CacheFS) is a generic nonvolatile caching mechanism. CacheFS improves the performance of certain file systems by utilizing a small, fast local disk. For example, you can improve the performance of the NFS environment by using CacheFS.

CacheFS works differently with different versions of NFS. For example, if both the client and the back file system are running NFS version 2 or version 3, the files are cached in the front file system for access by the client. However, if both the client and the server are running NFS

version 4, the functionality is as follows. When the client makes the initial request to access a file from a CacheFS file system, the request bypasses the front (or cached) file system and goes directly to the back file system. With NFS version 4, files are no longer cached in a front file system. All file access is provided by the back file system. Also, since no files are being cached in the front file system, CacheFS-specific mount options, which are meant to affect the front file system, are ignored. CacheFS-specific mount options do not apply to the back file system.

Note – The first time you configure your system for NFS version 4, a warning appears on the console to indicate that caching is no longer performed.

▼ How to Access NFS File Systems by Using CacheFS

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 who is 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 which are set in maps that are searched earlier. The last options that are found are the ones that are used. In the previous example, an additional entry to the `auto_home` map only needs to include the options in the master maps if some options required changes.

Customizing the Automounter

You can set up the automounter maps in several ways. The following tasks give details about 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'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
```

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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Install home directory partitions under /export/home.

If the system has several partitions, install the partitions under separate directories, for example, /export/home1 and /export/home2.

3 Use the Solaris Management Console 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. The ampersand 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*. *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.

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, a simple change is sufficient. You need only change `rusty`'s entry in the `auto_home` map to reflect the new location. Other users can continue to use the `/home/rusty` path.

How to Consolidate Project-Related Files Under `/ws`

Assume that you are the administrator of a large software development project. You plan to make all project-related files available under a directory that is 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 resembles 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 the following:

```
compiler        alpha:/export/ws/compiler
```

This first attempt provides a map that appears simple, but the map is 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, the map still contains only the five entries. Each entry is larger because each entry 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 informs autofs that the entry is continued onto the next line. Effectively, the entry is one long line, though line breaks and some indenting have been used to make the entry more readable. The `tools` directory contains software development tools for all subprojects, so this directory 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 typically consume substantial amounts of disk space. Through the life of the project, you might be required to relocate and expand various disk partitions. If 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` namespace 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 Namespace

You need to assemble a shared namespace for local executables, and applications, such as spreadsheet applications and word-processing packages. The clients of this namespace 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.**

   See the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

2. **Choose a single, site-specific name for the shared namespace.**

   This name makes the files and directories that belong to this space easily identifiable. For example, if you choose `/usr/local` as the name, the path `/usr/local/bin` is obviously a part of this namespace.

3. **For ease of user community recognition, create an autofs indirect map.**

   Mount this map at `/usr/local`. Set up the following entry in the NIS `auto_master` map:

   ```
   /usr/local     auto_local     -ro
   ```

   Notice that the `-ro` mount option implies that clients cannot 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 resembles the following:

   ```
   bin     aa:/export/local/bin
   ```

6. **(Optional) 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 x86 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.**

You can combine the autofs `OSREL` variable 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 that are running version 5.6 of the operating system, export the following file systems:

- For SPARC clients – Export `/export/local/bin/sparc5.6`.
- For x86 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 181 for a discussion of failover.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Modify the entry in the autofs maps.**

Create the list of all replica servers as a comma-separated list, such as the following:

```
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 Autofs Security Restrictions

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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. The override happens 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. Therefore, 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Create an entry in the autofs map such as the following:**

   ```
   /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 fails.

How to Use NFS URLs With Autofs

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Create an autofs entry such as the following:**

   ```
   /usr/local      -ro     nfs://bee/export/share/local
   ```

   The service tries to use the public file handle on the NFS server. However, if the server does not support a public file handle, the MOUNT protocol is used.

Disabling Autofs Browsability

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, you might have 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 the feature by using a command-line option to the `automountd` daemon, which completely disables autofs browsability for the client. Or disable browsability for each map entry on all clients by using the autofs maps in either an NIS or NIS+ namespace. You can also disable the feature for each map entry on each client, using local autofs maps if no network-wide namespace is being used.

How to Completely Disable Autofs Browsability on a Single NFS Client

1. **Become superuser or assume an equivalent role on the NFS client.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit the `/etc/default/autofs` file to include the following keyword and value.**

   ```
   AUTOMOUNTD_NOBROWSE=TRUE
   ```

3. **Restart the autofs service.**

   ```
   # svcadm restart system/filesystem/autofs
   ```

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. **Run the `automount` command on all clients.**

   The new behavior becomes effective after you run the `automount` command on the client systems or after a reboot.

   ```
   # /usr/sbin/automount
   ```

How to Disable Autofs Browsability on a Selected File System

In this example, browsability of the `/net` directory is disabled. You can use the same procedure for `/home` or any other autofs mount points.

1. **Check the `automount` entry in `/etc/nsswitch.conf`.**

   For local file entries to have precedence, the entry in the name service switch file should list `files` before the name service. For example:

   ```
   automount:  files nis
   ```

   This entry shows 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 have precedence over the entries in the namespace, the `+auto_master` entry must be moved to follow `/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 placement 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 becomes effective after running the `automount` command on the client systems or after a reboot.

   ```
   # /usr/sbin/automount
   ```

Strategies for NFS Troubleshooting

When tracking an NFS problem, remember the main points of possible failure: the server, the client, and the network. The strategy that is outlined in this section tries to isolate each individual component to find the one that is not working. In all situations, the `mountd` and `nfsd` daemons must be running on the server for remote mounts to succeed.

The `-intr` option is set by default for all mounts. If a program hangs with a `server not responding` message, you can kill the program with the keyboard interrupt Control-c.

When the network or server has problems, programs that access hard-mounted remote files fail differently than those programs 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 the file system hangs if the server fails to respond. In this situation, 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 that accesses 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, you might notice that other portions of the network are not functioning. For example, the name service or the physical network hardware might not be functioning. The *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)* contains debugging procedures for several name services. Also, during the process you might see that the problem is not at the client end. An example is if you get at least one trouble call from every subnet in your work area. In this situation, you should assume that the problem is the server or the network hardware near the server. So, you should 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 115.

2. **If the server is not reachable from the client, ensure 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, ensure 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 117.

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 command fails, check the networking software configuration on the client, for example, `/etc/netmasks` and `/etc/nsswitch.conf`.

6. **(Optional) Check the output of the `rpcinfo` command.**

   If the `rpcinfo` command does not display `program 100003 version 4 ready and waiting`, then NFS version 4 is not enabled on the server. See Table 5–3 for information about enabling NFS version 4.

7. **If the software is correct, check the networking hardware.**

   Try to move the client onto a second net drop.

▼ How to Check the NFS Server Remotely

Note that support for both the UDP and the MOUNT protocols is not necessary if you are using an NFS version 4 server.

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,tcp6,upd6              nfs     superuser
```

```
 100005  3,2,1  ticots,ticotsord,tcp,tcp6,ticlts,udp,upd6  mountd  superuser
```

If the daemons have not been started, see "How to Restart NFS Services" on page 118.

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
```

Note – NFS version 4 does not support UDP.

If the server is running, it prints a list of program and version numbers. Using the `-t` option tests the TCP connection. If this command fails, proceed to "How to Verify the NFS Service on the Server" on page 117.

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 that are associated with the UDP protocol. Using the `-t` option tests the TCP connection. If either attempt fails, proceed to "How to Verify the NFS Service on the Server" on page 117.

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 command fails, then as `root` on the client, type the following to restart the autofs service:

```
# svcadm restart system/filesystem/autofs
```

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 namespace. In this instance, if the first client is not in the `eng` netgroup, that client cannot mount the `/usr/src` file system.

Check all entries that include mounting information in all 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Check that the server can reach the clients.**

```
# ping lilac
lilac is alive
```

3. **If the client is not reachable from the server, ensure 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, for example, `/etc/netmasks` and `/etc/nsswitch.conf`.**

5. **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 that are associated with the UDP protocol. If `rpcbind` seems to be hung, reboot the server.

6. **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
```

Note – NFS version 4 does not support UDP.

   If the server is running, it prints a list of program and version numbers that are 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 118.

7 **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 that are 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 118.

▼ How to Restart NFS Services

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Restart the NFS service on the server.**

Type the following command.

```
# svcadm restart network/nfs/server
```

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

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. However, 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. Therefore, verify that the options that are listed in the file match those options that are listed on the command line. In this way, you can check those options that are 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 improve the problem-solving process. The section 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 that are 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 the cause.

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*

Description: While scanning a direct map, autofs has found an entry key without a prefixed /.

Solution: Keys in direct maps must be full path names.

`bad key` *key* `in indirect map` *mapname*
Description: While scanning an indirect map, autofs has found an entry key that contains a `/`.

Solution: Indirect map keys must be simple names, not path names.

`can't mount` *server*`:`*pathname: reason*
Description: The mount daemon on the server refuses to provide a file handle for *server*:*pathname*.

Solution: Check the export table on the server.

`couldn't create mount point` *mountpoint*`:` *reason*
Description: Autofs was unable to create a mount point that was required for a mount. This problem most frequently occurs when you attempt to hierarchically mount all of a server's exported file systems.

Solution: A required mount point can exist only in a file system that cannot be mounted, which means the file system cannot be exported. The mount point cannot be created because the exported parent file system is exported read-only.

`leading space in map entry` *entry* `text in` *mapname*
Description: Autofs has discovered an entry in an automount map that contains leading spaces. This problem is usually an indication of an improperly continued map entry. For example:

```
fake
/blat           frobz:/usr/frotz
```

Solution: 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`
Description: The required map cannot be located. This message is produced only when the `-v` option is used.

Solution: Check the spelling and path name of the map name.

`remount` *server*`:`*pathname* `on` *mountpoint*`: server not responding`
Description: Autofs has failed to remount a file system that it previously unmounted.

Solution: Contact Sun for assistance. This error message is extremely rare and has no straightforward solution.

`WARNING:` *mountpoint* `already mounted on`
Description: Autofs is attempting to mount over an existing mount point. This message means that an internal error occurred in autofs (an anomaly).

Solution: Contact Sun for assistance. This error message is extremely rare and has no straightforward solution.

Miscellaneous Error Messages

`dir` *mountpoint* `must start with '/'`

Solution: The automounter mount point must be given as a full path name. Check the spelling and path name of the mount point.

`hierarchical mountpoint:` *pathname1* `and` *pathname2*

Solution: 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`

Description: Autofs attempted to contact *server*, but received no response.

Solution: Check the NFS server status.

hostname`: exports:` *rpc-err*

Description: An error occurred while getting the export list from *hostname*. This message indicates a server or network problem.

Solution: Check the NFS server status.

`map` *mapname*`, key` *key*`: bad`

Description: The map entry is malformed, and autofs cannot interpret the entry.

Solution: Recheck the entry. Perhaps the entry has characters that need to be escaped.

mapname`:` *nis-err*

Description: An error occurred when looking up an entry in a NIS map. This message can indicate NIS problems.

Solution: Check the NIS server status.

`mount of` *server*`:`*pathname* `on` *mountpoint*:*reason*

Description: Autofs failed to do a mount. This occurrence can indicate a server or network problem. The *reason* string defines the problem.

Solution: Contact Sun for assistance. This error message is extremely rare and has no straightforward solution.

mountpoint: Not a directory
Description: Autofs cannot mount itself on *mountpoint* because it is not a directory.

Solution: Check the spelling and path name of the mount point.

nfscast: cannot send packet: *reason*
Description: Autofs cannot send a query packet to a server in a list of replicated file system locations. The *reason* string defines the problem.

Solution: Contact Sun for assistance. This error message is extremely rare and has no straightforward solution.

nfscast: cannot receive reply: *reason*
Description: Autofs cannot receive replies from any of the servers in a list of replicated file system locations. The *reason* string defines the problem.

Solution: Contact Sun for assistance. This error message is extremely rare and has no straightforward solution.

nfscast: select: *reason*
Description: All these error messages indicate problems in attempting to check servers for a replicated file system. This message can indicate a network problem. The *reason* string defines the problem.

Solution: Contact Sun for assistance. This error message is extremely rare and has no straightforward solution.

pathconf: no info for *server*:*pathname*
Description: Autofs failed to get `pathconf` information for the path name.

Solution: See the `fpathconf(2)` man page.

pathconf: *server*: server not responding
Description: Autofs is unable to contact the mount daemon on *server* that provides the information to `pathconf()`.

Solution: Avoid using the `POSIX` mount option with this server.

Other Errors With Autofs

If the `/etc/auto*` files have the execute bit set, the automounter tries to execute the maps, which creates messages such as the following:

```
/etc/auto_home: +auto_home: not found
```

In this situation, the `auto_home` file has incorrect permissions. Each entry in the file generates an error message that is similar to this message. 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 that is followed by a description of the conditions that should create the error and at minimum one remedy.

`Bad argument specified with index option - must be a file`
Solution: 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`
Description: This message is often created when the services information in the namespace has not been updated. The message can also be reported for UDP.

Solution: To fix this problem, you must update the services data in the namespace.

For NIS+, the entries should be as follows:

```
nfsd nfsd tcp 2049 NFS server daemon
nfsd nfsd udp 2049 NFS server daemon
```

For NIS and `/etc/services`, the entries should be as follows:

```
nfsd    2049/tcp    nfs    # NFS server daemon
nfsd    2049/udp    nfs    # NFS server daemon
```

`Cannot use index option without public option`
Solution: Include the `public` option with the `index` option in the share command. You must define the public file handle in order for the `index` option to work.

Note – The Solaris 2.5.1 release required that the public file handle be set by using the `share` command. A change in the Solaris 2.6 release sets the public file handle to be `root` (`/`) by default. This error message is no longer relevant.

`Could not start `*`daemon`*`: `*`error`*
Description: This message is displayed if the daemon terminates abnormally or if a system call error occurs. The *error* string defines the problem.

Solution: Contact Sun for assistance. This error message is rare and has no straightforward solution.

`Could not use public filehandle in request to` *server*

Description: This message is displayed if the `public` option is specified but the NFS server does not support the public file handle. In this situation, the mount fails.

Solution: 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.

daemon `running already with pid` *pid*

Description: The daemon is already running.

Solution: If you want to run a new copy, kill the current version and start a new version.

`error locking` *lock file*

Description: This message is displayed when the *lock file* that is associated with a daemon cannot be locked properly.

Solution: Contact Sun for assistance. This error message is rare and has no straightforward solution.

`error checking` *lock file*`:` *error*

Description: This message is displayed when the *lock file* that is associated with a daemon cannot be opened properly.

Solution: Contact Sun for assistance. This error message is rare and has no straightforward solution.

`NOTICE: NFS3: failing over from` *host1* `to` *host2*

Description: This message is displayed on the console when a failover occurs. The message is advisory only.

Solution: No action required.

filename`: File too large`

Description: An NFS version 2 client is trying to access a file that is over 2 Gbytes.

Solution: Avoid using NFS version 2. Mount the file system with version 3 or version 4. Also, see the description of the `nolargefiles` option in "`mount` Options for NFS File Systems" on page 148.

`mount: ... server not responding:RPC_PMAP_FAILURE - RPC_TIMED_OUT`

Description: The server that is 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.

Solution: Wait for the server to reboot. If the server is hung, reboot the server.

`mount: ... server not responding: RPC_PROG_NOT_REGISTERED`
Description: The mount request registered with `rpcbind`, but the NFS mount daemon `mountd` is not registered.

Solution: Wait for the server to reboot. If the server is hung, reboot the server.

`mount: ... No such file or directory`
Description: Either the remote directory or the local directory does not exist.

Solution: Check the spelling of the directory names. Run `ls` on both directories.

`mount: ...: Permission denied`
Description: Your computer name might not be in the list of clients or netgroup that is allowed access to the file system you tried to mount.

Solution: Use `showmount -e` to verify the access list.

`NFS file temporarily unavailable on the server, retrying ...`
Description: An NFS version 4 server can delegate the management of a file to a client. This message indicates that the server is recalling a delegation for another client that conflicts with a request from your client.

Solution: The recall must occur before the server can process your client's request. For more information about delegation, refer to "Delegation in NFS Version 4" on page 175.

`NFS fsstat failed for server` *hostname*`: RPC: Authentication error`
Description: This error can be caused by many situations. One of the most difficult situations to debug is when this problem occurs because a user is in too many groups. Currently, a user can be in no more than 16 groups if the user is accessing files through NFS mounts.

Solution: An alternate does exist for users who need to be in more than 16 groups. You can use access control lists to provide the needed access privileges.

`nfs mount: ignoring invalid option "-`*option*`"`
Description: The *-option* flag is not valid.

Solution: 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 that is 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"`
Description: An NFS client has attempted to mount a file system from an NFS server by using the `-nolargefiles` option.

Solution: This option is not supported for NFS file system types.

`nfs mount: NFS V2 can't support "largefiles"`

Description: The NFS version 2 protocol cannot handle large files.

Solution: You must use version 3 or version 4 if access to large files is required.

`NFS server` *hostname* `not responding still trying`

Description: If programs hang while doing file-related work, your NFS server might have failed. This message indicates that NFS server *hostname* is down or that a problem has occurred with the server or the network.

Solution: If failover is being used, *hostname* is a list of servers. Start troubleshooting with "How to Check Connectivity on an NFS Client" on page 114.

`NFS server recovering`

Description: During part of the NFS version 4 server reboot, some operations were not permitted. This message indicates that the client is waiting for the server to permit this operation to proceed.

Solution: No action required. Wait for the server to permit the operation.

`Permission denied`

Description: This message is displayed by the `ls -l`, `getfacl`, and `setfacl` commands for the following reasons:

- If the user or group that exists in an access control list (ACL) entry on an NFS version 4 server cannot be mapped to a valid user or group on an NFS version 4 client, the user is not allowed to read the ACL on the client.
- If the user or group that exists in an ACL entry that is being set on an NFS version 4 client cannot be mapped to a valid user or group on an NFS version 4 server, the user is not allowed to write or modify an ACL on the client.
- If an NFS version 4 client and server have mismatched NFSMAPID_DOMAIN values, ID mapping fails.

For more information, see "ACLs and `nfsmapid` in NFS Version 4" on page 177.

Solution: Do the following:

- Make sure that all user and group IDs in the ACL entries exist on both the client and server.
- Make sure that the value for NFSMAPID_DOMAIN is set correctly in the `/etc/default/nfs` file. For more information, see "Keywords for the `/etc/default/nfs` File" on page 131.

To determine if any user or group cannot be mapped on the server or client, use the script that is provided in "Checking for Unmapped User or Group IDs" on page 178.

`port` *number* `in nfs URL not the same as port` *number* `in port option`
Description: The port number that is included in the NFS URL must match the port number that is included with the `-port` option to mount. If the port numbers do not match, the mount fails.

Solution: Either change the command to make the port numbers identical or do not specify the port number that is incorrect. Usually, you do not need to specify the port number with both the NFS URL and the `-port` option.

`replicas must have the same version`
Description: For NFS failover to function properly, the NFS servers that are replicas must support the same version of the NFS protocol.

Solution: Running multiple versions is not allowed.

`replicated mounts must be read-only`
Description: 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 could change.

Solution: NFS failover depends on the file systems being identical.

`replicated mounts must not be soft`
Description: Replicated mounts require that you wait for a timeout before failover occurs.

Solution: 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`
Solution: 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`
Description: 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 does not work.

Solution: Upgrade the server with a new version of the OS that provides complete lock manager support.

◆ ◆ ◆ CHAPTER 6

Accessing Network File Systems (Reference)

This chapter describes the NFS commands, as well as the different parts of the NFS environment and how these parts work together.

Note – If your system has zones enabled and you want to use this feature in a non-global zone, see *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones* for more information.

NFS Files

You need several files to support NFS activities on any computer. Many of these files are ASCII, but some of the files are data files. Table 6–1 lists these files and their functions.

TABLE 6–1 NFS Files

File Name	Function
`/etc/default/autofs`	Lists configuration information for the autofs environment.
`/etc/default/fs`	Lists the default file-system type for local file systems.

TABLE 6–1 NFS Files *(Continued)*

File Name	Function
`/etc/default/nfs`	Lists configuration information for `lockd` and `nfsd`. For more information, refer to "Keywords for the `/etc/default/nfs` File" on page 131 and the `nfs(4)` man page.
`/etc/default/nfslogd`	Lists configuration information for the NFS server logging daemon, `nfslogd`.
`/etc/dfs/dfstab`	Lists the local resources to be shared.
`/etc/dfs/fstypes`	Lists the default file-system types for remote file systems.
`/etc/dfs/sharetab`	Lists the local and remote resources 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 postprocessing by `nfslogd`. Do not edit this file.
`/etc/nfssec.conf`	Lists NFS security services.
`/etc/rmtab`	Lists file systems that are 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/autofs` File

Starting in the Solaris 10 release, you can use the `/etc/default/autofs` file to configure your autofs environment. Specifically, this file provides an additional way to configure your autofs commands and autofs daemons. The same specifications you would make on the command line can be made in this configuration file. However, unlike the specifications you would make on the command line, this file preserves your specifications, even during upgrades to your operating system. Additionally, you are no longer required to update critical startup files to

ensure that the existing behavior of your autofs environment is preserved. You can make your specifications by providing values for the following keywords:

AUTOMOUNT_TIMEOUT
: Sets the duration for a file system to remain idle before the file system is unmounted. This keyword is the equivalent of the `-t` argument for the `automount` command. The default value is 600.

AUTOMOUNT_VERBOSE
: Provides notification of autofs mounts, unmounts, and other nonessential events. This keyword is the equivalent of the `-v` argument for `automount`. The default value is FALSE.

AUTOMOUNTD_VERBOSE
: Logs status messages to the console and is the equivalent of the `-v` argument for the `automountd` daemon. The default value is FALSE.

AUTOMOUNTD_NOBROWSE
: Turns browsing on or off for all autofs mount points and is the equivalent of the `-n` argument for `automountd`. The default value is FALSE.

AUTOMOUNTD_TRACE
: Expands each remote procedure call (RPC) and displays the expanded RPC on standard output. This keyword is the equivalent of the `-T` argument for `automountd`. The default value is 0. Values can range from 0 to 5.

AUTOMOUNTD_ENV
: Permits you to assign different values to different environments. This keyword is the equivalent of the `-D` argument for `automountd`. The AUTOMOUNTD_ENV keyword can be used multiple times. However, you must use separate lines for each environment assignment.

For more information, refer to the man pages for `automount`(1M) and `automountd`(1M). For procedural information, refer to "How to Configure Your Autofs Environment Using the `/etc/default/autofs` File" on page 100.

Keywords for the `/etc/default/nfs` File

In NFS version 4, the following keywords can be set in the `/etc/default/nfs` file. These keywords control the NFS protocols that are used by both the client and server.

NFS_SERVER_VERSMIN
: Sets the minimum version of the NFS protocol to be registered and offered by the server. Starting in the Solaris 10 release, the default is 2. Other valid values include 3 or 4. Refer to "Setting Up NFS Services" on page 88.

NFS_SERVER_VERSMAX
: Sets the maximum version of the NFS protocol to be registered and offered by the server. Starting in the Solaris 10 release, the default is 4. Other valid values include 2 or 3. Refer to "Setting Up NFS Services" on page 88.

NFS_CLIENT_VERSMIN
: Sets the minimum version of the NFS protocol to be used by the NFS client. Starting in the Solaris 10 release, the default is 2. Other valid values include 3 or 4. Refer to "Setting Up NFS Services" on page 88.

NFS_CLIENT_VERSMAX
: Sets the maximum version of the NFS protocol to be used by the NFS client. Starting in the Solaris 10 release, the default is 4. Other valid values include 2 or 3. Refer to "Setting Up NFS Services" on page 88.

NFS_SERVER_DELEGATION
: Controls whether the NFS version 4 delegation feature is enabled for the server. If this feature is enabled, the server attempts to provide delegations to the NFS version 4 client. By default, server delegation is enabled. To disable server delegation, see "How to Select Different Versions of NFS on a Server" on page 91. For more information, refer to "Delegation in NFS Version 4" on page 175.

NFSMAPID_DOMAIN
: Sets a common domain for clients and servers. Overrides the default behavior of using the local DNS domain name. For task information, refer to "Setting Up NFS Services" on page 88. Also, see "`nfsmapid` Daemon" on page 138.

`/etc/default/nfslogd` File

This file defines some of the parameters that are 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. This parameter 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 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 reduced. The default value is 168 hours or 7 days.

UMASK
: Specifies the file mode creation mask for the log files that are created by `nfslogd`. The default value is 0137.

`/etc/nfs/nfslog.conf` File

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. You can use the following parameters with each tag as needed.

defaultdir=*path*
: Specifies the default directory path for the logging files. Unless you specify differently, the default directory is `/var/nfs`.

log=*path/filename*
: Sets the path and file name for the log files. The default is `/var/nfs/nfslog`.

fhtable=*path/filename*
: Selects the path and file name for the file-handle-to-path database files. The default is `/var/nfs/fhtable`.

buffer=*path/filename*
: Determines the path and file name for the buffer files. The default is `/var/nfs/nfslog_workbuffer`.

logformat=*basic*|*extended*
: Selects the format to be used when creating user-readable log files. The basic format produces a log file that is similar to some `ftpd` daemons. The extended format gives a more

detailed view.

If the path is not specified, the path that is defined by `defaultdir` is used. Also, you can override `defaultdir` by using an absolute path.

To identify the files more easily, place the files in separate directories. Here is an example of the changes that are 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 that is shared with `log=publicftp` uses the following values:

- The default directory is `/var/nfs`.
- Log files are stored in `/var/nfs/logs/nfslog*`.
- File-handle-to-path database tables are stored in `/var/nfs/fh/fhtables`.
- Buffer files are stored in `/var/nfs/buffers/workbuffer`.

For procedural information, refer to "How to Enable NFS Server Logging" on page 82.

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 servers. The automatic startup of the server daemons depends on the existence of entries that are labeled with the NFS file-system type in `/etc/dfs/sharetab`. To support NFS file locking, the `lockd` and `statd` daemons are run on NFS clients and servers. However, unlike previous versions of NFS, in NFS version 4, the daemons `lockd`, `statd`, `mountd`, and `nfslogd` are not used.

This section describes the following daemons.

- "statd Daemon" on page 145

automountd Daemon

This daemon handles the mounting and unmounting requests from the autofs service. The syntax of the command is as follows:

`automountd [ -Tnv ] [ -D` *name*`=`*value* `]`

The command behaves in the following ways:

- `-T` enables tracing.
- `-n` disables browsing on all autofs nodes.
- `-v` selects to log all status messages to the console.
- `-D` *name*`=`*value* substitutes *value* for the automount map variable that is indicated by *name*.

The default value for the automount map is `/etc/auto_master`. Use the `-T` option for troubleshooting.

lockd Daemon

This daemon supports record-locking operations on NFS files. The `lockd` daemon 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. These options can either be used from the command line or by editing the appropriate string in `/etc/default/nfs`. The following are descriptions of keywords that can be set in the `/etc/default/nfs` file.

Note – Starting in the Solaris 10 release, the `LOCKD_GRACE_PERIOD` keyword and the `-g` option have been deprecated. The deprecated keyword is replaced with the new keyword `GRACE_PERIOD`. If both keywords are set, the value for `GRACE_PERIOD` overrides the value for `LOCKD_GRACE_PERIOD`. See the description of `GRACE_PERIOD` that follows.

Like `LOCKD_GRACE_PERIOD`, `GRACE_PERIOD`=*graceperiod* in `/etc/default/nfs` sets the number of seconds after a server reboot that the clients have to reclaim both NFS version 3 locks, provided by NLM, and version 4 locks. Thus, the value for `GRACE_PERIOD` controls the length of the grace period for lock recovery, for both NFS version 3 and NFS version 4.

The `LOCKD_RETRANSMIT_TIMEOUT`=*timeout* parameter in `/etc/default/nfs` 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. However, this change can cause additional server load by increasing the frequency of lock requests. The same parameter can be used from the command line by starting the daemon with the `-t` *timeout* option.

The `LOCKD_SERVERS=`*nthreads* parameter in `/etc/default/nfs` specifies the maximum number of concurrent threads that the server handles per connection. Base the value for *nthreads* on the load that is expected on the NFS server. The default value is 20. Each NFS client that uses TCP uses a single connection with the NFS server. Therefore, each client can use a maximum of 20 concurrent threads on the server.

All NFS clients that use UDP share a single connection with the NFS server. Under these conditions, you might have to increase the number of threads that are available for the UDP connection. A minimum calculation would be to allow two threads for each UDP client. However, this number 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. If the threads are never used, however, increasing *nthreads* has no effect. The same parameter can be used from the command line by starting the daemon with the `nthreads` option.

`mountd` Daemon

This daemon handles file-system mount requests from remote systems and provides access control. The `mountd` daemon 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 the `-v` option and the `-r` option with this command. See the `mountd(1M)` man page.

The `-v` option runs the command in verbose mode. Every time an NFS server determines the access that a client should be granted, a message is printed on the console. The information that is 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 option does not affect clients that already have a file system mounted.

Note – NFS version 4 does not use this daemon.

`nfs4cbd` Daemon

`nfs4cbd`, which is for the exclusive use of the NFS version 4 client, manages the communication endpoints for the NFS version 4 callback program. The daemon has no user-accessible interface. For more information, see the `nfs4cbd(1M)` man page.

`nfsd` Daemon

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. These options can either be used from the command line or by editing the appropriate string in `/etc/default/nfs`.

The `NFSD_LISTEN_BACKLOG=`*length* parameter in `/etc/default/nfs` sets the length of the connection queue over connection-oriented transports for NFS and TCP. The default value is 32 entries. The same selection can be made from the command line by starting `nfsd` with the `-l` option.

The `NFSD_MAX_CONNECTIONS=`*#-conn* parameter in `/etc/default/nfs` selects the maximum number of connections per connection-oriented transport. The default value for *#-conn* is unlimited. The same parameter can be used from the command line by starting the daemon with the `-c` *#-conn* option.

The `NFSD_SERVER=`*nservers* parameter in `/etc/default/nfs` selects the maximum number of concurrent requests that a server can handle. The default value for *nservers* is 16. The same selection can be made from the command line by starting `nfsd` with the `nservers` option.

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` Daemon

This daemon provides operational logging. NFS operations that are logged against a server are based on the configuration options that are defined in `/etc/default/nfslogd`. When NFS server logging is enabled, records of all RPC operations on a selected file system are written to a buffer file by the kernel. Then `nfslogd` postprocesses 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 tracks these mappings in a file-handle-to-path mapping table. One mapping table exists for each tag that is identified in `/etc/nfs/nfslogd`. After post-processing, the records are written to ASCII log files.

Note – NFS version 4 does not use this daemon.

`nfsmapid` Daemon

Version 4 of the NFS protocol (RFC3530) changed the way user or group identifiers (UID or GID) are exchanged between the client and server. The protocol requires that a file's owner and group attributes be exchanged between an NFS version 4 client and an NFS version 4 server as strings in the form of `user@nfsv4_domain` or `group@nfsv4_domain`, respectively.

For example, user `known_user` has a UID 123456 on an NFS version 4 client whose fully qualified hostname is `system.example.com`. For the client to make requests to the NFS version 4 server, the client must map the UID 123456 to `known_user@example.com` and then send this attribute to the NFS version 4 server. The NFS version 4 server expects to receive user and group file attributes in the `user_or_group@nfsv4_domain` format. After the server receives `known_user@example.com` from the client, the server maps the string to the local UID 123456, which is understood by the underlying file system. This functionality assumes that every UID and GID in the network is unique and that the NFS version 4 domains on the client match the NFS version 4 domains on the server.

Note – If the server does not recognize the given user or group name, even if the NFS version 4 domains match, the server is unable to map the user or group name to its unique ID, an integer value. Under such circumstances, the server maps the inbound user or group name to the `nobody` user. To prevent such occurrences, administrators should avoid making special accounts that only exist on the NFS version 4 client.

The NFS version 4 client and server are both capable of performing integer-to-string and string-to-integer conversions. For example, in response to a GETATTR operation, the NFS version 4 server maps UIDs and GIDs obtained from the underlying file system into their respective string representation and sends this information to the client. Alternately, the client must also map UIDs and GIDs into string representations. For example, in response to the `chown` command, the client maps the new UID or GID to a string representation before sending a SETATTR operation to the server.

Note, however, that the client and server respond differently to unrecognized strings:

- If the user does not exist on the server, even within the same NFS version 4 domain configuration, the server rejects the remote procedure call (RPC) and returns an error message to the client. This situation limits the operations that can be performed by the remote user.
- If the user exists on both the client and server, but they have mismatched domains, the server rejects the attribute modifying operations (such as SETATTR) that require the server to map the inbound user string to an integer value that the underlying file system can understand. For NFS version 4 clients and servers to function properly, their NFS version 4 domains, the portion of the string after the `@` sign, should match.

- If the NFS version 4 client does not recognize a user or group name from the server, the client is unable to map the string to its unique ID, an integer value. Under such circumstances, the client maps the inbound user or group string to the `nobody` user. This mapping to `nobody` creates varied problems for different applications. As for NFS version 4 functionality, operations that modify file attributes will fail.

You can change the domain name for the clients and servers using the `sharectl` command with the following option.

nfsmapid_domain
: Sets a common domain for clients and servers. Overrides the default behavior of using the local DNS domain name. For task information, refer to "Setting Up NFS Services" on page 88.

Configuration Files and `nfsmapid`

The following describes how the `nfsmapid` daemon uses the `/etc/nsswitch.conf` and `/etc/resolv.conf` files:

- `nfsmapid` uses standard C library functions to request password and group information from back-end name services. These name services are controlled by the settings in the `/etc/nsswitch.conf` file. Any changes to the `nsswitch.conf` file affect `nfsmapid` operations. For more information about the `nsswitch.conf` file, see the `nsswitch.conf(4)` man page.
- To ensure that the NFS version 4 clients are capable of mounting file systems from different domains, `nfsmapid` relies on the configuration of the DNS TXT resource record (RR), `_nfsv4idmapdomain`. For more information about configuring the `_nfsv4idmapdomain` resource record, see "`nfsmapid` and DNS TXT Records" on page 140. Also, note the following:
  - The DNS TXT RR should be explicitly configured on the DNS server with the desired domain information.
  - The `/etc/resolv.conf` file should be configured with the desired parameters to enable the `resolver` to find the DNS server and search the TXT records for client and server NFS version 4 domains.

  For more information, see the following:

  - "Precedence Rules" on page 140
  - "Configuring the NFS Version 4 Default Domain" on page 142
  - `resolv.conf(4)` man page

Precedence Rules

For `nfsmapid` to work properly, NFS version 4 clients and servers must have the same domain. To ensure matching NFS version 4 domains, `nfsmapid` follows these strict precedence rules:

1. The daemon first checks the `/etc/default/nfs` file for a value that has been assigned to the NFSMAPID_DOMAIN keyword. If a value is found, the assigned value takes precedence over any other settings. The assigned value is appended to the outbound attribute strings and is compared against inbound attribute strings. For more information about keywords in the `/etc/default/nfs` file, see "Keywords for the `/etc/default/nfs` File" on page 131. For procedural information, see "Setting Up NFS Services" on page 88.

   Note – The use of the NFSMAPID_DOMAIN setting is not scalable and is not recommended for large deployments.

2. If no value has been assigned to NFSMAPID_DOMAIN, then the daemon checks for a domain name from a DNS TXT RR. `nfsmapid` relies on directives in the `/etc/resolv.conf` file that are used by the set of routines in the `resolver`. The `resolver` searches through the configured DNS servers for the `_nfsv4idmapdomain` TXT RR. Note that the use of DNS TXT records is more scalable. For this reason, continued use of TXT records is much preferred over setting the keyword in the `/etc/default/nfs` file.
3. If no DNS TXT record is configured to provide a domain name, then the `nfsmapid` daemon uses the value specified by the `domain` or `search` directive in the `/etc/resolv.conf` file, with the directive specified last taking precedence.

   In the following example, where both the `domain` and `search` directives are used, the `nfsmapid` daemon uses the first domain listed after the `search` directive, which is `company.com`.

   ```
   domain example.company.com
   search company.com foo.bar.com
   ```

4. If the `/etc/resolv.conf` file does not exist, `nfsmapid` obtains the NFS version 4 domain name by following the behavior of the `domainname` command. Specifically, if the `/etc/defaultdomain` file exists, `nfsmapid` uses the contents of that file for the NFS version 4 domain. If the `/etc/defaultdomain` file does not exist, `nfsmapid` uses the domain name that is provided by the network's configured naming service. For more information, see the `domainname(1M)` man page.

`nfsmapid` and DNS TXT Records

The ubiquitous nature of DNS provides an efficient storage and distribution mechanism for the NFS version 4 domain name. Additionally, because of the inherent scalability of DNS, the use of DNS TXT resource records is the preferred method for configuring the NFS version 4 domain name for large deployments. You should configure the `_nfsv4idmapdomain` TXT record on

enterprise-level DNS servers. Such configurations ensure that any NFS version 4 client or server can find its NFS version 4 domain by traversing the DNS tree.

The following is an example of a preferred entry for enabling the DNS server to provide the NFS version 4 domain name:

```
_nfsv4idmapdomain        IN        TXT            "foo.bar"
```

In this example, the domain name to configure is the value that is enclosed in double-quotes. Note that no `ttl` field is specified and that no domain is appended to `_nfsv4idmapdomain`, which is the value in the `owner` field. This configuration enables the TXT record to use the zone's `${ORIGIN}` entry from the Start-Of-Authority (SOA) record. For example, at different levels of the domain namespace, the record could read as follows:

```
_nfsv4idmapdomain.subnet.yourcorp.com.    IN    TXT    "foo.bar"
_nfsv4idmapdomain.yourcorp.com.           IN    TXT    "foo.bar"
```

This configuration provides DNS clients with the added flexibility of using the `resolv.conf` file to search up the DNS tree hierarchy. See the `resolv.conf(4)` man page. This capability provides a higher probability of finding the TXT record. For even more flexibility, lower level DNS sub-domains can define their own DNS TXT resource records (RRs). This capability enables lower level DNS sub-domains to override the TXT record that is defined by the top level DNS domain.

Note – The domain that is specified by the TXT record can be an arbitrary string that does not necessarily match the DNS domain for clients and servers that use NFS version 4. You have the option of not sharing NFS version 4 data with other DNS domains.

Checking for the NFS Version 4 Domain

Before assigning a value for your network's NFS version 4 domain, check to see if an NFS version 4 domain has already been configured for your network. The following examples provide ways of identifying your network's NFS version 4 domain.

- To identify the NFS version 4 domain from a DNS TXT RR, use either the `nslookup` or the `dig` command:

  The following provides sample output for the `nslookup` command:

  ```
  # nslookup -q=txt _nfsv4idmapdomain
  Server:         10.255.255.255
  Address:        10.255.255.255#53

  _nfsv4idmapdomain.example.company.com text = "company.com"
  ```

  See this sample output for the `dig` command:

  ```
  # dig +domain=example.company.com -t TXT _nfsv4idmapdomain
  ...
  ;; QUESTION SECTION:
  ```

```
;_nfsv4idmapdomain.example.company.com. IN    TXT

;; ANSWER SECTION:
_nfsv4idmapdomain.example.company.com. 21600 IN TXT   "company.com"

;; AUTHORITY SECTION:
...
```

For information about setting up a DNS TXT RR, see "nfsmapid and DNS TXT Records" on page 140.

- If your network is not setup with a NFS version 4 DNS TXT RR, use the following command to identify your NFS version 4 domain from the DNS domain name:

```
# egrep domain /etc/resolv.conf
domain example.company.com
```

- If the `/etc/resolv.conf` file is not configured to provide a DNS domain name for the client, use the following command to identify the domain from the network's NFS version 4 domain configuration:

```
# cat /var/run/nfs4_domain
company.com
```

- If you are using a different naming service, such as NIS, use the following command to identify the domain for the naming service configured for your network:

```
# domainname
it.example.company.com
```

For more information, see the following man pages:

- `nslookup`(1M)
- `dig`(1M)
- `resolv.conf`(4)
- `domainname`(1M)

Configuring the NFS Version 4 Default Domain

This section describes how the network obtains the desired default domain:

- For most current releases, see "Configuring an NFS Version 4 Default Domain" on page 143.
- For the initial Solaris 10 release, see "Configuring an NFS Version 4 Default Domain in the Solaris 10 Release" on page 144.

Configuring an NFS Version 4 Default Domain

In the initial Solaris 10 release, the domain was defined during the first system reboot after installing the OS. In later releases, the NFS version 4 domain is defined during the installation of the OS. To provide this functionality, the following features have been added:

- The `sysidtool` command includes the `sysidnfs4` program. This program runs during the installation process to determine whether an NFS version 4 domain has been configured for the network. See the man pages for `sysidtool(1M)` and `sysidnfs4(1M)`.
- The `sysidcfg` file has a new keyword, `nfs4_domain`. This keyword can be used to define the NFS version 4 domain. Note that other keywords can also be defined in the `sysidcfg` file. See the `sysidcfg(4)` man page.

The following describes how the functionality operates:

1. The `sysidnfs4` program checks the `/etc/.sysIDtool.state` file to determine whether an NFS version 4 domain has been identified.
   - If the `.sysIDtool.state` file shows that an NFS version 4 domain has been configured for the network, the `sysidnfs4` program makes no further checks. See the following example of a `.sysIDtool.state` file:

     ```
     1       # System previously configured?
     1       # Bootparams succeeded?
     1       # System is on a network?
     1       # Extended network information gathered?
     1       # Autobinder succeeded?
     1       # Network has subnets?
     1       # root password prompted for?
     1       # locale and term prompted for?
     1       # security policy in place
     1       # NFSv4 domain configured
     xterms
     ```

     The `1` that appears before `# NFSv4 domain configured` confirms that the NFS version 4 domain has been configured.
   - If the `.sysIDtool.state` file shows that no NFS version 4 domain has been configured for the network, the `sysidnfs4` program must make further checks. See the following example of a `.sysIDtool.state` file:

     ```
     1       # System previously configured?
     1       # Bootparams succeeded?
     1       # System is on a network?
     1       # Extended network information gathered?
     1       # Autobinder succeeded?
     1       # Network has subnets?
     1       # root password prompted for?
     1       # locale and term prompted for?
     1       # security policy in place
     0       # NFSv4 domain configured
     xterms
     ```

     The `0` that appears before `# NFSv4 domain configured` confirms that no NFS version 4 domain has been configured.

2. If no NFS version 4 domain has been identified, the `sysidnfs4` program checks the `nfs4_domain` keyword in the `sysidcfg` file.
   - If a value for `nfs4_domain` exists, that value is assigned to the NFSMAPID_DOMAIN keyword in the `/etc/default/nfs` file. Note that any value assigned to NFSMAPID_DOMAIN overrides the dynamic domain selection capability of the `nfsmapid` daemon. For more information about the dynamic domain selection capability of `nfsmapid`, see "Precedence Rules" on page 140.
   - If no value for `nfs4_domain` exists, the `sysidnfs4` program identifies the domain that `nfsmapid` derives from the operating system's configured name services. This derived value is presented as a default domain at an interactive prompt that gives you the option of accepting the default value or assigning a different NFS version 4 domain.

This functionality makes the following obsolete:

- The sample JumpStart script, `set_nfs4_domain`, which was provided in the initial Solaris 10 media distribution is no longer required and is discouraged.
- The `/etc/.NFS4inst_state.domain` file, which was created by the previous implementation of the `sysidnfs4` program, is no longer required.

Note – Because of the inherent ubiquitous and scalable nature of DNS, the use of DNS TXT records for configuring the domain of large NFS version 4 deployments continues to be preferred and strongly encouraged. See "`nfsmapid` and DNS TXT Records" on page 140.

For specific information about the Solaris installation process, see the following:

- *Oracle Solaris 10 8/11 Installation Guide: Basic Installations*
- *Oracle Solaris 10 8/11 Installation Guide: Network-Based Installations*

Configuring an NFS Version 4 Default Domain in the Solaris 10 Release

In the initial Solaris 10 release of NFS version 4, if your network includes multiple DNS domains, but only has a single UID and GID namespace, all clients must use one value for NFSMAPID_DOMAIN. For sites that use DNS, `nfsmapid` resolves this issue by obtaining the domain name from the value that you assigned to `_nfsv4idmapdomain`. For more information, see "`nfsmapid` and DNS TXT Records" on page 140. If your network is not configured to use DNS, during the first system boot the OS uses the `sysidconfig(1M)` utility to provide the following prompts for an NFS version 4 domain name:

```
This system is configured with NFS version 4, which uses a
domain name that is automatically derived from the system's
name services. The derived domain name is sufficient for most
configurations. In a few cases, mounts that cross different
domains might cause files to be owned by nobody due to the
lack of a common domain name.
```

```
Do you need to override the system's default NFS verion 4 domain
name (yes/no)? [no]
```

The default response is `[no]`. If you choose `[no]`, you see the following:

```
For more information about how the NFS version 4 default domain name is
derived and its impact, refer to the man pages for nfsmapid(1M) and
nfs(4), and the System Administration Guide: Network Services.
```

If you choose `[yes]`, you see this prompt:

```
Enter the domain to be used as the NFS version 4 domain name.
NFS version 4 domain name []:
```

Note – If a value for NFSMAPID_DOMAIN exists in `/etc/default/nfs`, the `[domain_name]` that you provide overrides that value.

Additional Information About `nfsmapid`

For more information about `nfsmapid`, see the following:

- `nfsmapid(1M)` man page
- `nfs(4)` man page
- `http://www.ietf.org/rfc/rfc1464.txt`
- "ACLs and `nfsmapid` in NFS Version 4" on page 177

`statd` Daemon

This daemon works with `lockd` to provide crash and recovery functions for the lock manager. The `statd` daemon 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. You have 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` tracks the clients has been improved. In all earlier Solaris releases, `statd` created files in `/var/statmon/sm` for each client by using the client's unqualified host name. This file naming caused problems if you had two clients in different domains that shared a host name, or if clients 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 by using the IP address of the client. The new link resembles the following:

```
# ls -l /var/statmon/sm
lrwxrwxrwx   1 daemon          11 Apr 29 16:32 ipv4.192.168.255.255 -> myhost
lrwxrwxrwx   1 daemon          11 Apr 29 16:32 ipv6.fec0::56:a00:20ff:feb9:2734 -> v6host
--w-------   1 daemon          11 Apr 29 16:32 myhost
--w-------   1 daemon          11 Apr 29 16:32 v6host
```

In this example, the client host name is `myhost` and the client's IP address is `192.168.255.255`. If another host with the name `myhost` were mounting a file system, two symbolic links would lead to the host name.

Note – NFS version 4 does not use this daemon.

NFS Commands

These commands must be run as `root` to be fully effective, but requests for information can be made by all users:

`automount` Command

This command installs autofs mount points and associates the information in the automaster files with each mount point. The syntax of the command is as follows:

`automount [ -t` *duration* `] [ -v ]`

`-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 value is good. 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, which is 30 minutes, could be more optimal. By not unmounting the file systems every 5 minutes, `/etc/mnttab` can become large. To reduce the output when `df` checks each entry in `/etc/mnttab`, you can filter the output from `df` by using the `-F` option (see the `df(1M)` man page) or by using `egrep`.

You should consider that adjusting the duration also changes how quickly changes to the automounter maps are reflected. Changes cannot be seen until the file system is unmounted. Refer to "Modifying the Maps" on page 102 for instructions on how to modify automounter maps.

`clear_locks` Command

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. 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 that is 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. You must run this option from the NFS client, which created the locks. In this situation, the locks from the client would be removed from the NFS server that is 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.

`fsstat` Command

Starting in the Solaris 10 11/06 release, the `fsstat` utility enables you to monitor file system operations by file system type and by mount point. Various options allow you to customize the output. See the following examples.

This example shows output for NFS version 3, version 4, and the `root` mount point.

```
% fsstat nfs3 nfs4 /
  new     name   name    attr    attr   lookup   rddir   read   read   write   write
 file    remov   chng     get     set      ops     ops    ops  bytes     ops   bytes
3.81K       90  3.65K   5.89M   11.9K    35.5M   26.6K   109K   118M   35.0K   8.16G  nfs3
  759      503    457   93.6K   1.44K     454K   8.82K  65.4K   827M     292    223K  nfs4
25.2K    18.1K  1.12K   54.7M    1017     259M   1.76M  22.4M  20.1G   1.43M   3.77G  /
```

This example uses the `-i` option to provide statistics about the I/O operations for NFS version 3, version 4, and the `root` mount point.

```
% fsstat -i nfs3 nfs4 /
 read    read    write   write   rddir   rddir   rwlock   rwulock
  ops   bytes      ops   bytes     ops   bytes      ops       ops
 109K    118M    35.0K   8.16G   26.6K   4.45M     170K      170K  nfs3
65.4K    827M      292    223K   8.82K   2.62M    74.1K     74.1K  nfs4
22.4M   20.1G    1.43M   3.77G   1.76M   3.29G    25.5M     25.5M  /
```

This example uses the `-n` option to provide statistics about the naming operations for NFS version 3, version 4, and the `root` mount point.

```
% fsstat -n nfs3 nfs4 /
lookup   creat   remov  link   renam  mkdir  rmdir   rddir  symlnk  rdlnk
 35.5M   3.79K      90     2   3.64K      5      0   26.6K      11   136K  nfs3
  454K     403     503     0     101      0      0   8.82K     356  1.20K  nfs4
  259M   25.2K   18.1K   114    1017     10      2   1.76M      12  8.23M  /
```

For more information, see the `fsstat(1M)` man page.

`mount` Command

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, see `mount_ufs(1M)`.

The Solaris 7 release includes the ability to select a path name to mount from an NFS server by 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 88 for further information.

Caution – The version of the `mount` command does not warn about invalid options. The command silently ignores any options that cannot be interpreted. Ensure that you verify all of the options that were used so that you can prevent unexpected behavior.

`mount` Options for NFS File Systems

The subsequent text lists some of the options that can follow the `-o` flag when you are mounting an NFS file system. For a complete list of options, refer to the `mount_nfs(1M)` man page.

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. This option prevents further processing until the mount is complete. `bg` is

a good selection for noncritical file systems because the client can do other processing while waiting for the mount request to be completed.

forcedirectio
: This option improves performance of large sequential data transfers. Data is copied directly to a user buffer. No caching is performed in the kernel on the client. This option is off by default.

  Previously, all write requests were serialized by both the NFS client and the NFS server. The NFS client has been modified to permit an application to issue concurrent writes, as well as concurrent reads and writes, to a single file. You can enable this functionality on the client by using the `forcedirectio` mount option. When you use this option, you are enabling this functionality for all files within the mounted file system. You could also enable this functionality on a single file on the client by using the `directio()` interface. Unless this functionality has been enabled, writes to files are serialized. Also, if concurrent writes or concurrent reads and writes are occurring, then POSIX semantics are no longer being supported for that file.

  For an example of how to use this option, refer to "Using the `mount` Command" on page 151.

largefiles
: With this option, you can access files that are larger than 2 Gbytes. 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. By default, all UFS file systems are mounted with `largefiles`. For mounts that use the NFS version 2 protocol, the `largefiles` option causes the mount to fail with an error.

nolargefiles
: This option for UFS mounts guarantees that no large files can exist 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, no option for `nolargefiles` exists when using NFS mounts. Attempts to NFS-mount a file system by using this option are rejected with an error.

nosuid|suid
: Starting in the Solaris 10 release, the `nosuid` option is the equivalent of specifying the `nodevices` option with the `nosetuid` option. When the `nodevices` option is specified, the opening of device-special files on the mounted file system is disallowed. When the `nosetuid` option is specified, the `setuid` bit and `setgid` bit in binary files that are located in the file system are ignored. The processes run with the privileges of the user who executes the binary file.

  The `suid` option is the equivalent of specifying the `devices` option with the `setuid` option. When the `devices` option is specified, the opening of device-special files on the mounted file system is allowed. When the `setuid` option is specified, the `setuid` bit and the `setgid` bit in binary files that are located in the file system are honored by the kernel.

  If neither option is specified, the default option is `suid`, which provides the default behavior of specifying the `devices` option with the `setuid` option.

The following table describes the effect of combining `nosuid` or `suid` with `devices` or `nodevices`, and `setuid` or `nosetuid`. Note that in each combination of options, the most restrictive option determines the behavior.

Behavior From the Combined Options	Option	Option	Option
The equivalent of `nosetuid` with `nodevices`	`nosuid`	`nosetuid`	`nodevices`
The equivalent of `nosetuid` with `nodevices`	`nosuid`	`nosetuid`	`devices`
The equivalent of `nosetuid` with `nodevices`	`nosuid`	`setuid`	`nodevices`
The equivalent of `nosetuid` with `nodevices`	`nosuid`	`setuid`	`devices`
The equivalent of `nosetuid` with `nodevices`	`suid`	`nosetuid`	`nodevices`
The equivalent of `nosetuid` with `devices`	`suid`	`nosetuid`	`devices`
The equivalent of `setuid` with `nodevices`	`suid`	`setuid`	`nodevices`
The equivalent of `setuid` with `devices`	`suid`	`setuid`	`devices`

The `nosuid` option provides additional security for NFS clients that access potentially untrusted servers. The mounting of remote file systems with this option reduces the chance of privilege escalation through importing untrusted devices or importing untrusted `setuid` binary files. All these options are available in all Solaris file systems.

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 following.

- Use `krb5` for Kerberos version 5 authentication service.
- Use `krb5i` for Kerberos version 5 with integrity.
- Use `krb5p` for Kerberos version 5 with privacy.
- Use `none` for no authentication.
- Use `dh` for Diffie-Hellman (DH) authentication.
- Use `sys` for standard UNIX authentication.

The modes are also defined in `/etc/nfssec.conf`.

soft|hard
An NFS file system that is 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. If the application does check the return values, routing problems and other conditions can still confuse the application or lead to file corruption if the `soft` option is used. In most situations, the `soft` option should not be used. If a file system is mounted by using the `hard` option and becomes unavailable, an application that uses this file system hangs until the file system becomes available.

Using the `mount` Command

Refer to the following examples.

- In NFS version 2 or version 3, 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
```

In NFS version 4, the following command line would accomplish the same mount.

```
# mount -F nfs -o vers=4 -r bee:/export/share/man /usr/man
```

- In NFS version 2 or version 3, 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. See the following.

```
# mount -F nfs -O bee:/export/share/man /usr/man
```

In NFS version 4, the following command line would accomplish the same mount.

```
# mount -F nfs -o vers=4 -O bee:/export/share/man /usr/man
```

- In NFS version 2 or version 3, this command uses client failover.

  ```
  # mount -F nfs -r bee,wasp:/export/share/man /usr/man
  ```

  In NFS version 4, the following command line uses client failover.

  ```
  # mount -F nfs -o vers=4 -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 use both version 2 and version 3 servers when running `mount` from the command line. You can use both servers with autofs. Autofs automatically selects the best subset of version 2 or version 3 servers.

- Here is an example of using an NFS URL with the `mount` command in NFS version 2 or version 3.

  ```
  # mount -F nfs nfs://bee//export/share/man /usr/man
  ```

  Here is an example of using an NFS URL with the `mount` command in NFS version 4.

  ```
  # mount -F nfs -o vers=4 nfs://bee//export/share/man /usr/man
  ```

- Use the `forcedirectio` mount option to enable the client to permit concurrent writes, as well as concurrent reads and writes, to a file. Here is an example.

  ```
  # mount -F nfs -o forcedirectio bee:/home/somebody /mnt
  ```

  In this example, the command mounts an NFS file system from the server bee and enables concurrent reads and writes for each file in the directory `/mnt`. When support for concurrent reads and writes is enabled, the following occurs.

  - The client permits applications to write to a file in parallel.
  - Caching is disabled on the client. Consequently, data from reads and writes is kept on the server. More explicitly, because the client does not cache the data that is read or written, any data that the application does not already have cached for itself is read from the server. The client's operating system does not have a copy of this data. Normally, the NFS client caches data in the kernel for applications to use.

    Because caching is disabled on the client, the read-ahead and write-behind processes are disabled. A read-ahead process occurs when the kernel anticipates the data that an application might request next. The kernel then starts the process of gathering that data in advance. The kernel's goal is to have the data ready before the application makes a request for the data.

    The client uses the write-behind process to increase write throughput. Instead of immediately starting an I/O operation every time an application writes data to a file, the data is cached in memory. Later, the data is written to the disk.

    Potentially, the write-behind process permits the data to be written in larger chunks or to be written asynchronously from the application. Typically, the result of using larger chunks is increased throughput. Asynchronous writes permit overlap between

application processing and I/O processing. Also, asynchronous writes permit the storage subsystem to optimize the I/O by providing a better sequencing of the I/O. Synchronous writes force a sequence of I/O on the storage subsystem that might not be optimal.

- Significant performance degradation can occur if the application is not prepared to handle the semantics of data that is not being cached. Multithreaded applications avoid this problem.

Note – If support for concurrent writes is not enabled, all write requests are serialized. When requests are serialized, the following occurs. When a write request is in progress, a second write request has to wait for the first write request to be completed before proceeding.

- Use the `mount` command with no arguments to display file systems that are mounted on a client. See the following.

```
% mount
/ on /dev/dsk/c0t3d0s0 read/write/setuid on Wed Apr 7 13:20:47 2004
/usr on /dev/dsk/c0t3d0s6 read/write/setuid on Wed Apr 7 13:20:47 20041995
/proc on /proc read/write/setuid on Wed Apr 7 13:20:47 2004
/dev/fd on fd read/write/setuid on Wed Apr 7 13:20:47 2004
/tmp on swap read/write on Wed Apr 7 13:20:51 2004
/opt on /dev/dsk/c0t3d0s5 setuid/read/write on Wed Apr 7 13:20:51 20041995
/home/kathys on bee:/export/home/bee7/kathys
  intr/noquota/nosuid/remote on Wed Apr 24 13:22:13 2004
```

`umount` Command

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 unmount 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 that are 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 use this option to unhang a client that is hung while trying to mount an unmountable file system.

Caution – By forcing an unmount of a file system, you can cause data loss if files are being written to.

See the following examples.

EXAMPLE 6–1 Unmounting a File System

This example unmounts a file system that is mounted on `/usr/man`:

```
# umount /usr/man
```

EXAMPLE 6–2 Using Options with `umount`

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` Command

Use this command to mount all file systems or a specific group of file systems that are listed in a file-system table. The command provides a way of doing the following:

- Selecting the file-system type to be accessed with the `-F` *FSType* option
- Selecting all the remote file systems that are listed in a file-system table with the `-r` option
- Selecting all the local file systems with the `-l` option

Because all file systems that are 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.

Note that the following two examples of user input are equivalent:

```
# mountall -F nfs
```

```
# mountall -F nfs -r
```

`umountall` Command

Use this command to unmount a group of file systems. The `-k` option runs the `fuser -k` *mount-point* command to kill any processes that are 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`.

The following is an example of unmounting all file systems that are mounted from remote hosts:

```
# umountall -r
```

The following is an example of unmounting all file systems that are currently mounted from the server bee:

```
# umountall -h bee
```

`share` Command

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 an entry is in `/etc/dfs/dfstab`. The command does not report an error if the NFS server software is not running, so you must verify that the software is running.

The objects that can be shared include any directory tree. However, 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 these directories 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 if that file system is part of a larger file system that is 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. However, if both directories need to be shared with different share options, `/usr/local` must be moved to a separate disk slice.

You can gain access to a file system that is read-only shared through the file handle of a file system that is read-write shared. However, the two file systems have to be on the same disk slice. You can create a more secure situation. Place those file systems that need to be read-write on a separate partition or separate disk slice from the file systems that you need to share as read-only.

Note – For information about how NFS version 4 functions when a file system is unshared and then reshared, refer to "Unsharing and Resharing a File System in NFS Version 4" on page 170.

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 for all clients.

rw=*accesslist*
: The file system is shared read-write for the clients that are listed only. All other requests are denied. Starting with the Solaris 2.6 release, the list of clients that are defined in *accesslist* has been expanded. See "Setting Access Lists With the `share` Command" on page 158 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 that supports 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 that are 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. A decision to permit too much access or too little access depends on the security systems already in place. See "Using Access Control Lists to Protect UFS Files" in *System Administration Guide: Security Services* for more information about access control lists (ACLs).

Note – To use ACLs, ensure that 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 obtain correct access but cannot manipulate the ACLs. If the software supports the NFS_ACL protocol, the clients obtain correct access and can manipulate the ACLs.

---

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 option allows unauthenticated users to have root access, so use the `root` option instead.

index=*filename*
: When a user accesses an NFS URL, the `-index=`*filename* option forces the HTML file to load, instead of displaying a list of the directory. 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 option is the equivalent of setting the `DirectoryIndex` option for `httpd`. For instance, suppose that the `dfstab` file entry resembles the following:

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

These URLs then 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 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 that are defined in *accesslist* is expanded. See "Setting Access Lists With the `share` Command" on page 158 for more information.

Caution – Granting root access to other hosts has wide security implications. Use the `-root=` option with extreme caution.

root=*client-name*
: The *client-name* value is used with AUTH_SYS authentication to check the client's IP address against a list of addresses provided by `exportfs(1B)`. If a match is found, `root` access is given to the file systems being shared.

root=*host-name*
: For secure NFS modes, such as AUTH_SYS or RPCSEC_GSS, the server checks the clients' principal names against a list of host-based principal names that are derived from an access list. The generic syntax for the client's principal name is `root@`*hostname*. For Kerberos V the syntax is `root/`*hostname.fully.qualified*`@REALM`. When you use the *host-name* value, the clients on the access list must have the credentials for a principal name. For Kerberos V, the client must have a valid keytab entry for its `root/`*hostname.fully.qualified*`@REALM` principal name. For more information, see "Configuring Kerberos Clients" in *System Administration Guide: Security Services*.

sec=*mode*[:*mode*]
: *mode* selects the security modes that are needed to obtain 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. The use of `-sec=none` maps all users to user `nobody`.

window=*value*
value selects the maximum lifetime 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* that was included with either the `-ro=`, `-rw=`, or `-root=` option of the `share` command was 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 simplify file access control on a single server without having to change the namespace 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 type all options on the same line. Multiple invocations of the `share` command on the same object "remember" only the last command that is 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 that uses multiple authentication mechanisms, ensure that you 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 that is 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 these hosts use Diffie-Hellman authentication. With these options, `tulip` and `lilac` can mount the file system read-only even if these hosts are not using DH authentication. However, the host names must be 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, UNIX authentication is not included if the `-sec` option is used. Therefore, you must 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 string that follows the dot is a domain name, not a fully qualified host name. The following entry allows mount access to all hosts in the `eng.example.com` domain:

```
# share -F nfs -o ro=.:.eng.example.com /export/share/man
```

In this example, the single "`.`" matches all hosts that are matched through the NIS or NIS+ namespaces. The results that are returned from these name services do not include the domain name. The "`.eng.example.com`" entry matches all hosts that use DNS for namespace 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 namespaces.

You can use a subnet number in an access list by preceding the actual network number or the network name with "`@`". This character differentiates the network name from a netgroup or a fully qualified host name. You must identify the subnet in either `/etc/networks` or in an NIS or NIS+ namespace. The following entries have the same effect if the `192.168` subnet has been identified as the `eng` network:

```
# share -F nfs -o ro=@eng /export/share/man
# share -F nfs -o ro=@192.168 /export/share/man
# share -F nfs -o ro=@192.168.0.0 /export/share/man
```

The last two entries show that 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=@192.168.0/17 /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 about CIDR, look up RFC 1519.

You can also select negative access by placing a "`-`" before the entry. Note that the entries are read from left to right. Therefore, you must place the negative access entries before the entry that the negative access entries apply to:

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

This example would allow access to any hosts in the `eng.example.com` domain except the host that is named `rose`.

`unshare` Command

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, be careful. Remember that the file system is 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 are not accessible.

Note – For information about how NFS version 4 functions when a file system is unshared and then reshared, refer to "Unsharing and Resharing a File System in NFS Version 4" on page 170.

The following is an example of unsharing a specific file system:

```
# unshare /usr/src
```

`shareall` Command

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.

The following is an example of sharing all file systems that are listed in a local file:

```
# shareall /etc/dfs/special_dfstab
```

`unshareall` Command

This command makes all currently shared resources unavailable. The `-F` *FSType* option selects a list of file-system types that are 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.

The following is an example of unsharing all NFS-type file systems:

```
# unshareall -F nfs
```

`showmount` Command

This command displays one of the following:

- All clients that have remotely mounted file systems that are shared from an NFS server
- Only the file systems that are mounted by clients
- The shared file systems with the client access information

Note – The `showmount` command only shows NFS version 2 and version 3 exports. This command does not show NFS version 4 exports.

The command syntax is as follows:

`showmount [ -ade ] [` *hostname* `]`

`-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 that are shared or are exported.
hostname	Selects the NFS server to gather the information from.

If *hostname* is not specified, the local host is queried.

The following command lists all clients and the local directories that the clients have mounted:

```
# showmount -a bee
lilac:/export/share/man
lilac:/usr/src
rose:/usr/src
tulip:/export/share/man
```

The following command lists the directories that have been mounted:

```
# showmount -d bee
/export/share/man
/usr/src
```

The following command lists file systems that have been shared:

```
# showmount -e bee
/usr/src                              (everyone)
/export/share/man                   eng
```

`setmnt` Command

This command creates an `/etc/mnttab` table. The `mount` and `umount` commands consult the table. Generally, you do not have to run this command manually, as this command runs automatically when a system is booted.

Commands for Troubleshooting NFS Problems

These commands can be useful when troubleshooting NFS problems.

`nfsstat` Command

You can use this command to gather statistical information about NFS and RPC connections. The syntax of the command is as follows:

`nfsstat [ -cmnrsz ]`

`-c` Displays client-side information

`-m` Displays statistics for each NFS-mounted file system

`-n` Specifies that NFS information is to be displayed on both the client side and the server side

`-r` Displays RPC statistics

`-s` Displays the server-side information

`-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 new hardware is added to the computing environment. Running this command a minimum of once a week, and storing the numbers, provides a good history of previous performance.

Refer to the following example:

```
# nfsstat -s

Server rpc:
Connection oriented:
calls      badcalls   nullrecv   badlen     xdrcall    dupchecks  dupreqs
719949194  0          0          0          0          58478624   33
Connectionless:
calls      badcalls   nullrecv   badlen     xdrcall    dupchecks  dupreqs
73753609   0          0          0          0          987278     7254
```

```
Server nfs:
calls                badcalls
787783794            3516
Version 2: (746607 calls)
null        getattr     setattr     root        lookup      readlink    read
883 0%      60 0%       45 0%       0 0%        177446 23%  1489 0%     537366 71%
wrcache     write       create      remove      rename      link        symlink
0 0%        1105 0%     47 0%       59 0%       28 0%       10 0%       9 0%
mkdir       rmdir       readdir     statfs
26 0%       0 0%        27926 3%    108 0%
Version 3: (728863853 calls)
null           getattr       setattr       lookup        access
1365467 0%     496667075 68% 8864191 1%    66510206 9%   19131659 2%
readlink       read          write         create        mkdir
414705 0%      80123469 10%  18740690 2%   4135195 0%    327059 0%
symlink        mknod         remove        rmdir         rename
101415 0%      9605 0%       6533288 0%    111810 0%     366267 0%
link           readdir       readdirplus   fsstat        fsinfo
2572965 0%     519346 0%     2726631 0%    13320640 1%   60161 0%
pathconf       commit
13181 0%       6248828 0%
Version 4: (54871870 calls)
null                compound
266963 0%           54604907 99%
Version 4: (167573814 operations)
reserved            access              close               commit
0 0%                2663957 1%          2692328 1%          1166001 0%
create              delegpurge          delegreturn         getattr
167423 0%           0 0%                1802019 1%          26405254 15%
getfh               link                lock                lockt
11534581 6%         113212 0%           207723 0%           265 0%
locku               lookup              lookupp             nverify
230430 0%           11059722 6%         423514 0%           21386866 12%
open                openattr            open_confirm        open_downgrade
2835459 1%          4138 0%             18959 0%            3106 0%
putfh               putpubfh            putrootfh           read
52606920 31%        0 0%                35776 0%            4325432 2%
readdir             readlink            remove              rename
606651 0%           38043 0%            560797 0%           248990 0%
renew               restorefh           savefh              secinfo
2330092 1%          8711358 5%          11639329 6%         19384 0%
setattr             setclientid         setclientid_confirm verify
453126 0%           16349 0%            16356 0%            2484 0%
write               release_lockowner   illegal
3247770 1%          0 0%                0 0%

Server nfs_acl:
Version 2: (694979 calls)
null        getacl      setacl      getattr     access      getxattrdir
0 0%        42358 6%    0 0%        584553 84%  68068 9%    0 0%
Version 3: (2465011 calls)
null        getacl      setacl      getxattrdir
0 0%        1293312 52% 1131 0%     1170568 47%
```

The previous listing is an example of NFS server statistics. The first five lines relate to 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 a problem. The `badcalls` value reports the number of bad messages from a client. This value can indicate 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 and version 4 statistics, the value to watch is `commit`. If the `commit` level is high in one NFS server, compared to another almost identical server, check that the NFS clients have enough memory. The number of `commit` operations on the server grows when clients do not have available resources.

`pstack` Command

This command displays a stack trace for each process. The `pstack` command must be run by the owner of the process or by `root`. You can use `pstack` to determine where a process is hung. The only option that is 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/bin/pgrep nfsd
243
# /usr/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
```

The example shows that the process is waiting for a new connection request, which 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 118 to fix this problem. Review the instructions in "NFS Troubleshooting Procedures" on page 114 to fully verify that your problem is a hung program.

`rpcinfo` Command

This command generates information about the RPC service that is running on a system. You can also use this command to change the RPC service. Many options are available with this command. See the `rpcinfo(1M)` man page. The following 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*]

`-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 that use specific transports or protocols

`-t` Probes the RPC programs that use TCP

`-u` Probes the RPC programs that use UDP

transport Selects the transport or protocol for the services

hostname Selects the host name of the server that you need information from

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 can 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 that is generated by this command can include the following:

- 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

The following example gathers information about the RPC services that are running on a server. The text that is generated by the command is filtered by the `sort` command to make the output more readable. Several lines that list RPC services have been deleted from the example.

```
% rpcinfo -s bee |sort -n
   program version(s) netid(s)                         service     owner
    100000  2,3,4     udp6,tcp6,udp,tcp,ticlts,ticotsord,ticots rpcbind     superuser
    100001  4,3,2     ticlts,udp,udp6                   rstatd      superuser
    100002  3,2       ticots,ticotsord,tcp,tcp6,ticlts,udp,udp6 rusersd     superuser
    100003  3,2       tcp,udp,tcp6,udp6                 nfs         superuser
    100005  3,2,1     ticots,ticotsord,tcp,tcp6,ticlts,udp,udp6 mountd      superuser
    100007  1,2,3     ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 ypbind      superuser
    100008  1         ticlts,udp,udp6                   walld       superuser
    100011  1         ticlts,udp,udp6                   rquotad     superuser
    100012  1         ticlts,udp,udp6                   sprayd      superuser
    100021  4,3,2,1   tcp,udp,tcp6,udp6                 nlockmgr    superuser
    100024  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 status      superuser
    100029  3,2,1     ticots,ticotsord,ticlts           keyserv     superuser
    100068  5         tcp,udp                           cmsd        superuser
    100083  1         tcp,tcp6                          ttdbserverd superuser
```

```
    100099  3         ticotsord                        autofs      superuser
    100133  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -          superuser
    100134  1         ticotsord                        tokenring   superuser
    100155  1         ticots,ticotsord,tcp,tcp6        smserverd   superuser
    100221  1         tcp,tcp6                         -           superuser
    100227  3,2       tcp,udp,tcp6,udp6                nfs_acl     superuser
    100229  1         tcp,tcp6                         metad       superuser
    100230  1         tcp,tcp6                         metamhd     superuser
    100231  1         ticots,ticotsord,ticlts          -           superuser
    100234  1         ticotsord                        gssd        superuser
    100235  1         tcp,tcp6                         -           superuser
    100242  1         tcp,tcp6                         metamedd    superuser
    100249  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -          superuser
    300326  4         tcp,tcp6                         -           superuser
    300598  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -          superuser
    390113  1         tcp                              -           unknown
 805306368  1         ticots,ticotsord,ticlts,tcp,udp,tcp6,udp6 -          superuser
1289637086  1,5       tcp                              -           26069
```

The following two examples show how to gather information about a particular RPC service by selecting a particular transport on a server. The first example checks the `mountd` service that is running over TCP. The second example checks the NFS service that is running over UDP.

```
% 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
```

`snoop` Command

This command is often used to watch for packets on the network. The `snoop` command must be run as `root`. The use of this command is a good way to ensure 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* `]`

`-d` *device*	Specifies the local network interface
`-o` *filename*	Stores all the captured packets into the named file
hostname	Displays packets going to and from a specific host only

The `-d` *device* option is useful on those servers that have multiple network interfaces. You can use many expressions other than 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 simplify the review of the data.

`truss` Command

You can use this command to check if a process is hung. The `truss` command 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 follows.

`truss [ -t` *syscall* `] -p` *pid*

`-t` *syscall* Selects system calls to trace

`-p` *pid* Indicates the PID of the process to be traced

The *syscall* can be a comma-separated list of system calls to be traced. Also, starting *syscall* with an `!` 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.

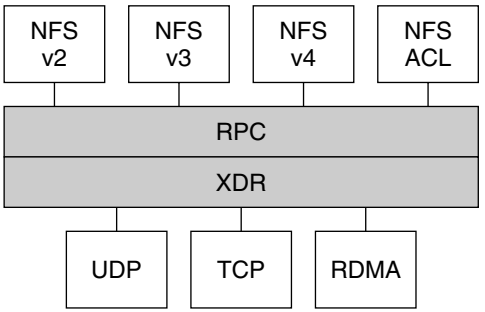
```
# /usr/bin/truss -p 243
poll(0x00024D50, 2, -1)         (sleeping...)
```

The previous example shows 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 118 to fix the hung program. Review the instructions in "NFS Troubleshooting Procedures" on page 114 to fully verify that your problem is a hung program.

NFS Over RDMA

The Solaris 10 release includes the Remote Direct Memory Access (RDMA) protocol, which is a technology for memory-to-memory transfer of data over high-speed networks. Specifically, RDMA provides remote data transfer directly to and from memory without CPU intervention. RDMA also provides direct data placement, which eliminates data copies and, therefore, further eliminates CPU intervention. Thus, RDMA relieves not only the host CPU, but also reduces contention for the host memory and I/O buses. To provide this capability, RDMA combines the interconnect I/O technology of InfiniBand on SPARC platforms with the Solaris operating system. The following figure shows the relationship of RDMA to other protocols, such as UDP and TCP.

FIGURE 6–1 Relationship of RDMA to Other Protocols



NFS is a family of protocols layered over RPC. The XDR (eXternal Data Representation) layer encodes RPC arguments and RPC results onto one of several RPC transports, such as UDP, TCP, and RDMA.

If the RDMA transport is not available on both the client and the server, the TCP transport is the initial fallback, followed by UDP if TCP is unavailable. Note, however, that if you use the `proto=rdma` mount option, NFS mounts are forced to use RDMA only.

For more information about NFS mount options, see the `mount_nfs(1M)` man page and "`mount` Command" on page 148.

Note – RDMA for InfiniBand uses the IP addressing format and the IP lookup infrastructure to specify peers. However, because RDMA is a separate protocol stack, it does not fully implement all IP semantics. For example, RDMA does not use IP addressing to communicate with peers. Therefore, RDMA might bypass configurations for various security policies that are based on IP addresses. However, the NFS and RPC administrative policies, such as `mount` restrictions and secure RPC, are not bypassed.

How the NFS Service Works

The following sections describe some of the complex functions of the NFS software. Note that some of the feature descriptions in this section are exclusive to NFS version 4.

- "Client-Side Failover" on page 181
- "Large Files" on page 183
- "How NFS Server Logging Works" on page 184
- "How the WebNFS Service Works" on page 184
- "WebNFS Limitations With Web Browser Use" on page 186
- "Secure NFS System" on page 186
- "Secure RPC" on page 187

Note – If your system has zones enabled and you want to use this feature in a non-global zone, see *System Administration Guide: Oracle Solaris Containers-Resource Management and Oracle Solaris Zones* for more information.

Version Negotiation in NFS

The NFS initiation process includes negotiating the protocol levels for servers and clients. If you do not specify the version level, then the best level is selected by default. For example, if both the client and the server can support version 3, then version 3 is used. If the client or the server can only support version 2, then version 2 is used.

Starting in the Solaris 10 release, you can set the keywords NFS_CLIENT_VERSMIN, NFS_CLIENT_VERSMAX, NFS_SERVER_VERSMIN, NFS_SERVER_VERSMAX in the `/etc/default/nfs` file. Your specified minimum and maximum values for the server and the client would replace the default values for these keywords. For both the client and the server the default minimum value is 2 and the default maximum value is 4. See "Keywords for the `/etc/default/nfs` File" on page 131. To find the version supported by the server, the NFS client begins with the setting for NFS_CLIENT_VERSMAX and continues to try each version until reaching the version setting for NFS_CLIENT_VERSMIN. As soon as the supported version is found, the process terminates. For example, if NFS_CLIENT_VERSMAX=4 and NFS_CLIENT_VERSMIN=2, then the client attempts version 4 first, then version 3, and finally version 2. If NFS_CLIENT_VERSMIN and NFS_CLIENT_VERSMAX are set to the same value, then the client always uses this version and does not attempt any other version. If the server does not offer this version, the mount fails.

Note – You can override the values that are determined by the negotiation by using the `vers` option with the `mount` command. See the `mount_nfs(1M)` man page.

For procedural information, refer to "Setting Up NFS Services" on page 88.

Features in NFS Version 4

Many changes have been made to NFS in version 4. This section provides descriptions of these new features.

Note – Starting in the Solaris 10 release, NFS version 4 does not support the LIPKEY/SPKM security flavor. Also, NFS version 4 does not use the `mountd`, `nfslogd`, and `statd` daemons.

For procedural information related to using NFS version 4, refer to "Setting Up NFS Services" on page 88.

Unsharing and Resharing a File System in NFS Version 4

With both NFS version 3 and version 4, if a client attempts to access a file system that has been unshared, the server responds with an error code. However, with NFS version 3 the server maintains any locks that the clients had obtained before the file system was unshared. Thus, when the file system is reshared, NFS version 3 clients can access the file system as though that file system had never been unshared.

With NFS version 4, when a file system is unshared, all the state for any open files or file locks in that file system is destroyed. If the client attempts to access these files or locks, the client receives an error. This error is usually reported as an `I/O` error to the application. Note, however, that resharing a currently shared file system to change options does not destroy any of the state on the server.

For related information, refer to "Client Recovery in NFS Version 4" on page 173 or see the `unshare_nfs(1M)` man page.

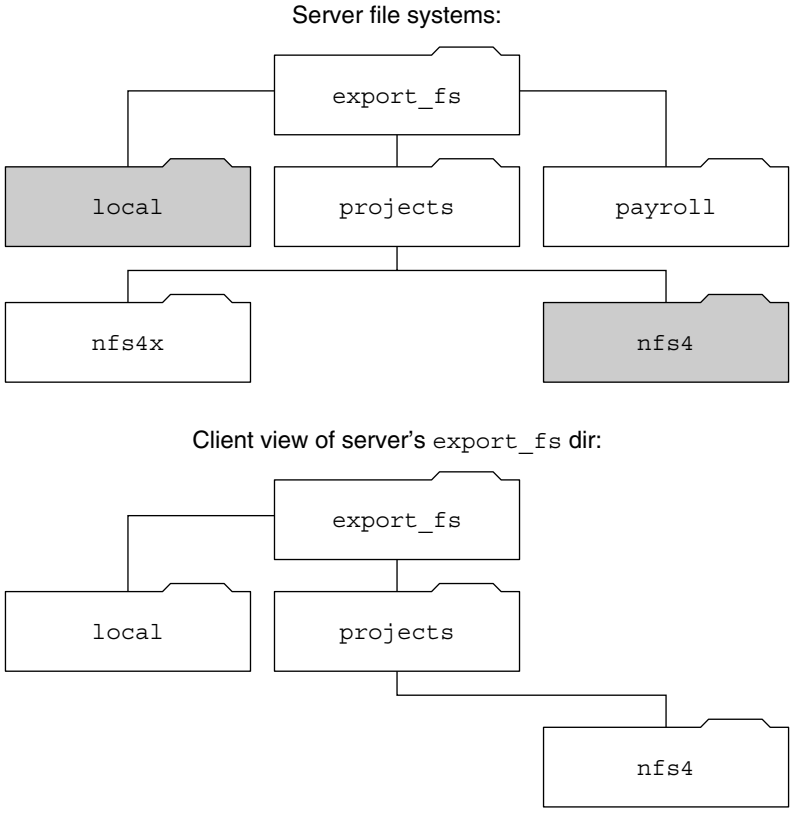
File-System Namespace in NFS Version 4

NFS version 4 servers create and maintain a pseudo-file system, which provides clients with seamless access to all exported objects on the server. Prior to NFS version 4, the pseudo-file system did not exist. Clients were forced to mount each shared server file system for access. Consider the following example.

FIGURE 6–2 Views of the Server File System and the Client File System

Server exports:
/export_fs/local
/export_fs/projects/nfs4

Server file systems:
/
/export_fs



Note that the client cannot see the `payroll` directory and the `nfs4x` directory, because these directories are not exported and do not lead to exported directories. However, the `local` directory is visible to the client, because `local` is an exported directory. The `projects` directory is visible to the client, because `projects` leads to the exported directory, `nfs4`. Thus, portions of the server namespace that are not explicitly exported are bridged with a pseudo-file system that views only the exported directories and those directories that lead to server exports.

A pseudo-file system is a structure that contains only directories and is created by the server. The pseudo-file system permits a client to browse the hierarchy of exported file systems. Thus, the client's view of the pseudo-file system is limited to paths that lead to exported file systems.

Previous versions of NFS did not permit a client to traverse server file systems without mounting each file system. However, in NFS version 4, the server namespace does the following:

- Restricts the client's file-system view to directories that lead to server exports.
- Provides clients with seamless access to server exports without requiring that the client mount each underlying file system. See the previous example. Note, however, that different operating systems might require the client to mount each server file system.

For POSIX-related reasons, the Solaris NFS version 4 client does not cross server file-system boundaries. When such attempts are made, the client makes the directory appear to be empty. To remedy this situation, you must perform a mount for each of the server's file systems.

Volatile File Handles in NFS Version 4

File handles are created on the server and contain information that uniquely identifies files and directories. In NFS versions 2 and 3 the server returned persistent file handles. Thus, the client could guarantee that the server would generate a file handle that always referred to the same file. For example:

- If a file was deleted and replaced with a file of the same name, the server would generate a new file handle for the new file. If the client used the old file handle, the server would return an error that the file handle was stale.
- If a file was renamed, the file handle would remain the same.
- If you had to reboot the server, the file handles would remain the same.

Thus, when the server received a request from a client that included a file handle, the resolution was straightforward and the file handle always referred to the correct file.

This method of identifying files and directories for NFS operations was fine for most UNIX-based servers. However, the method could not be implemented on servers that relied on other methods of identification, such as a file's path name. To resolve this problem, the NFS version 4 protocol permits a server to declare that its file handles are volatile. Thus, a file handle could change. If the file handle does change, the client must find the new file handle.

Like NFS versions 2 and 3, the Solaris NFS version 4 server always provides persistent file handles. However, Solaris NFS version 4 clients that access non-Solaris NFS version 4 servers must support volatile file handles if the server uses them. Specifically, when the server tells the client that the file handle is volatile, the client must cache the mapping between path name and file handle. The client uses the volatile file handle until it expires. On expiration, the client does the following:

- Flushes the cached information that refers to that file handle
- Searches for that file's new file handle
- Retries the operation

Note – The server always tells the client which file handles are persistent and which file handles are volatile.

Volatile file handles might expire for any of these reasons:

- When you close a file
- When the filehandle's file system migrates
- When a client renames a file
- When the server reboots

Note that if the client is unable to find the new file handle, an error message is put in the `syslog` file. Further attempts to access this file fail with an `I/O` error.

Client Recovery in NFS Version 4

The NFS version 4 protocol is a stateful protocol. A protocol is stateful when both the client and the server maintain current information about the following.

- Open files
- File locks

When a failure occurs, such as a server crash, the client and the server work together to reestablish the open and lock states that existed prior to the failure.

When a server crashes and is rebooted, the server loses its state. The client detects that the server has rebooted and begins the process of helping the server rebuild its state. This process is known as client recovery, because the client directs the process.

When the client discovers that the server has rebooted, the client immediately suspends its current activity and begins the process of client recovery. When the recovery process starts, a message, such as the following, is displayed in the system error log `/var/adm/messages`.

```
NOTICE: Starting recovery server basil.example.company.com
```

During the recovery process, the client sends the server information about the client's previous state. Note, however, that during this period the client does not send any new requests to the server. Any new requests to open files or set file locks must wait for the server to complete its recovery period before proceeding.

When the client recovery process is complete, the following message is displayed in the system error log `/var/adm/messages`.

```
NOTICE: Recovery done for server basil.example.company.com
```

Now the client has successfully completed sending its state information to the server. However, even though the client has completed this process, other clients might not have completed their process of sending state information to the server. Therefore, for a period of time, the server does not accept any open or lock requests. This period of time, which is known as the grace period, is designated to permit all the clients to complete their recovery.

During the grace period, if the client attempts to open any new files or establish any new locks, the server denies the request with the `GRACE` error code. On receiving this error, the client must wait for the grace period to end and then resend the request to the server. During the grace period the following message is displayed.

```
NFS server recovering
```

Note that during the grace period the commands that do not open files or set file locks can proceed. For example, the commands `ls` and `cd` do not open a file or set a file lock. Thus, these commands are not suspended. However, a command such as `cat`, which opens a file, would be suspended until the grace period ends.

When the grace period has ended, the following message is displayed.

```
NFS server recovery ok.
```

The client can now send new open and lock requests to the server.

Client recovery can fail for a variety of reasons. For example, if a network partition exists after the server reboots, the client might not be able to reestablish its state with the server before the grace period ends. When the grace period has ended, the server does not permit the client to reestablish its state because new state operations could create conflicts. For example, a new file lock might conflict with an old file lock that the client is trying to recover. When such situations occur, the server returns the `NO_GRACE` error code to the client.

If the recovery of an open operation for a particular file fails, the client marks the file as unusable and the following message is displayed.

```
WARNING: The following NFS file could not be recovered and was marked dead
(can't reopen:  NFS status 70):  file :  filename
```

Note that the number `70` is only an example.

If reestablishing a file lock during recovery fails, the following error message is posted.

```
NOTICE: nfs4_send_siglost:  pid PROCESS-ID lost
lock on server SERVER-NAME
```

In this situation, the `SIGLOST` signal is posted to the process. The default action for the `SIGLOST` signal is to terminate the process.

For you to recover from this state, you must restart any applications that had files open at the time of the failure. Note that the following can occur.

- Some processes that did not reopen the file could receive `I/O` errors.
- Other processes that did reopen the file, or performed the open operation after the recovery failure, are able to access the file without any problems.

Thus, some processes can access a particular file while other processes cannot.

`OPEN` Share Support in NFS Version 4

The NFS version 4 protocol provides several file-sharing modes that the client can use to control file access by other clients. A client can specify the following:

- `DENY_NONE` mode permits other clients read and write access to a file.
- `DENY_READ` mode denies other clients read access to a file.
- `DENY_WRITE` mode denies other clients write access to a file.
- `DENY_BOTH` mode denies other clients read and write access to a file.

The Solaris NFS version 4 server fully implements these file-sharing modes. Therefore, if a client attempts to open a file in a way that conflicts with the current share mode, the server denies the attempt by failing the operation. When such attempts fail with the initiation of the open or create operations, the NFS version 4 client receives a protocol error. This error is mapped to the application error `EACCES`.

Even though the protocol provides several sharing modes, currently the open operation in Solaris does not offer multiple sharing modes. When opening a file, a Solaris NFS version 4 client can only use the `DENY_NONE` mode.

Also, even though the `fcntl` system call has an `F_SHARE` command to control file sharing, the `fcntl` commands cannot be implemented correctly with NFS version 4. If you use these `fcntl` commands on an NFS version 4 client, the client returns the `EAGAIN` error to the application.

Delegation in NFS Version 4

NFS version 4 provides both client support and server support for delegation. Delegation is a technique by which the server delegates the management of a file to a client. For example, the

server could grant either a read delegation or a write delegation to a client. Read delegations can be granted to multiple clients at the same time, because these read delegations do not conflict with each other. A write delegation can be granted to only one client, because a write delegation conflicts with any file access by any other client. While holding a write delegation, the client would not send various operations to the server because the client is guaranteed exclusive access to a file. Similarly, the client would not send various operations to the server while holding a read delegation. The reason is that the server guarantees that no client can open the file in write mode. The effect of delegation is to greatly reduce the interactions between the server and the client for delegated files. Therefore, network traffic is reduced, and performance on the client and the server is improved. Note, however, that the degree of performance improvement depends on the kind of file interaction used by an application and the amount of network and server congestion.

The decision about whether to grant a delegation is made entirely by the server. A client does not request a delegation. The server makes decisions about whether to grant a delegation, based on the access patterns for the file. If a file has been recently accessed in write mode by several different clients, the server might not grant a delegation. The reason is that this access pattern indicates the potential for future conflicts.

A conflict occurs when a client accesses a file in a manner that is inconsistent with the delegations that are currently granted for that file. For example, if a client holds a write delegation on a file and a second client opens that file for read or write access, the server recalls the first client's write delegation. Similarly, if a client holds a read delegation and another client opens the same file for writing, the server recalls the read delegation. Note that in both situations, the second client is not granted a delegation because a conflict now exists. When a conflict occurs, the server uses a callback mechanism to contact the client that currently holds the delegation. On receiving this callback, the client sends the file's updated state to the server and returns the delegation. If the client fails to respond to the recall, the server revokes the delegation. In such instances, the server rejects all operations from the client for this file, and the client reports the requested operations as failures. Generally, these failures are reported to the application as `I/O` errors. To recover from these errors, the file must be closed and then reopened. Failures from revoked delegations can occur when a network partition exists between the client and the server while the client holds a delegation.

Note that one server does not resolve access conflicts for a file that is stored on another server. Thus, an NFS server only resolves conflicts for files that it stores. Furthermore, in response to conflicts that are caused by clients that are running various versions of NFS, an NFS server can only initiate recalls to the client that is running NFS version 4. An NFS server cannot initiate recalls for clients that are running earlier versions of NFS.

The process for detecting conflicts varies. For example, unlike NFS version 4, because version 2 and version 3 do not have an open procedure, the conflict is detected only after the client attempts to read, write, or lock a file. The server's response to these conflicts varies also. For example:

- For NFS version 3, the server returns the `JUKEBOX` error, which causes the client to halt the access request and try again later. The client prints the message `File unavailable`.
- For NFS version 2, because an equivalent of the `JUKEBOX` error does not exist, the server makes no response, which causes the client to wait and then try again. The client prints the message `NFS server not responding`.

These conditions clear when the delegation conflict has been resolved.

By default, server delegation is enabled. You can disable delegation by modifying the `/etc/default/nfs` file. For procedural information, refer to "How to Select Different Versions of NFS on a Server" on page 91.

No keywords are required for client delegation. The NFS version 4 callback daemon, `nfs4cbd`, provides the callback service on the client. This daemon is started automatically whenever a mount for NFS version 4 is enabled. By default, the client provides the necessary callback information to the server for all Internet transports that are listed in the `/etc/netconfig` system file. Note that if the client is enabled for IPv6 and if the IPv6 address for the client's name can be determined, then the callback daemon accepts IPv6 connections.

The callback daemon uses a transient program number and a dynamically assigned port number. This information is provided to the server, and the server tests the callback path before granting any delegations. If the callback path does not test successfully, the server does not grant delegations, which is the only externally visible behavior.

Note that because callback information is embedded within an NFS version 4 request, the server is unable to contact the client through a device that uses Network Address Translation (NAT). Also, the callback daemon uses a dynamic port number. Therefore, the server might not be able to traverse a firewall, even if that firewall enables normal NFS traffic on port 2049. In such situations, the server does not grant delegations.

ACLs and `nfsmapid` in NFS Version 4

An access control list (ACL) provides better file security by enabling the owner of a file to define file permissions for the file owner, the group, and other specific users and groups. ACLs are set on the server and the client by using the `setfacl` command. See the `setfacl(1)` man page for more information. In NFS version 4, the ID mapper, `nfsmapid`, is used to map user or group IDs in ACL entries on a server to user or group IDs in ACL entries on a client. The reverse is also true. The user and group IDs in the ACL entries must exist on both the client and the server.

Reasons for ID Mapping to Fail

The following situations can cause ID mapping to fail:

- If the user or group that exists in an ACL entry on the server cannot be mapped to a valid user or group on the client, the user is not allowed to read the ACL on the client.

  For example, when you issue the `ls -lv` or `ls -lV` command, you receive the error message, `Permission denied`,for the files with user or group ID ACL entities that cannot be mapped from the server to the client. The ID mapper was unable to map a user or group in the ACL. If the ID mapper had been able to map the user or group, a plus (+) sign would have appeared after the permissions in the files list that is produced by `ls -l`. For example:

  ```
  % ls -l
  -rw-r--rw-+   1 luis    staff      11968 Aug 12  2005 foobar
  ```

  Similarly, the `getfacl` command can return the `Permission denied` error message for the same reason. For more information about this command, see the `getfacl(1)` man page.
- If the user or group ID in any ACL entry that is set on the client cannot be mapped to a valid user or group ID on the server, the `setfacl` or the `chmod` command can fail and return the `Permission denied` error message.
- If the client and server have mismatched NFSMAPID_DOMAIN values, ID mapping fails. For more information, see "Keywords for the `/etc/default/nfs` File" on page 131.

Avoiding ID Mapping Problems With ACLs

To avoid ID mapping problems, do the following:

- Make sure that the value for NFSMAPID_DOMAIN is set correctly in the `/etc/default/nfs` file.
- Make sure that all user and group IDs in the ACL entries exist on both the NFS version 4 client and server.

Checking for Unmapped User or Group IDs

To determine if any user or group cannot be mapped on the server or client, use the following script:

```
#! /usr/sbin/dtrace -Fs

sdt:::nfs4-acl-nobody
{
     printf("validate_idmapping: (%s) in the ACL could not be mapped!",
stringof(arg0));
}
```

Note – The probe name that is used in this script is an interface that could change in the future. For more information, see "Stability Levels" in *Solaris Dynamic Tracing Guide*.

Additional Information About ACLs or nfsmapid

See the following:

- "Protecting UFS Files With ACLs (Task Map)" in *System Administration Guide: Security Services*
- Chapter 8, "Using ACLs to Protect Oracle Solaris ZFS Files," in *Oracle Solaris ZFS Administration Guide*
- "`nfsmapid` Daemon" on page 138

UDP and TCP Negotiation

During initiation, the transport protocol is also negotiated. By default, the first connection-oriented transport that is supported on both the client and the server is selected. If this selection does not succeed, the first available connectionless transport protocol is used. The transport protocols that are supported on a system are listed in `/etc/netconfig`. TCP is the connection-oriented transport protocol that is 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 that uses UDP is given higher precedence than the NFS version 2 protocol that is 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. However, 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 bid is not necessary.

The transfer size is not negotiated with systems that use 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.

Note – Starting in the Solaris 10 release, restrictions on wire transfer sizes have been relaxed. The transfer size is based on the capabilities of the underlying transport. For example, the NFS transfer limit for UDP is still 32 Kbytes. However, because TCP is a streaming protocol without the datagram limits of UDP, maximum transfer sizes over TCP have been increased to 1 Mbyte.

How File Systems Are Mounted

The following description applies to NFS version 3 mounts. The NFS version 4 mount process does not include the portmap service nor does it include the `MOUNT` protocol.

When a client needs to mount a file system from a server, the client must obtain a file handle from the server. The file handle must correspond 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 receives the mount port number (`33492`), that number is used to test the availability of 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, the server includes the file handle for the file system (`9000`) being mounted. The client then sends a request for the NFS port number. When the client receives the number from the server, the client tests the availability of the NFS service (`nfsd`). Also, the client requests NFS information about the file system that uses 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 the transactions to obtain information from the portmap service and to determine the NFS port number are skipped.

Note – NFS version 4 provides support for volatile file handles. For more information, refer to "Volatile File Handles in NFS Version 4" on page 172.

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 you use 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 this file handle is enabled on the NFS server. If the mount fails when using the public file handle, then try the mount with the `MOUNT` protocol.

Regular path only – Do not use the public file handle. The `MOUNT` protocol is used.

Client-Side Failover

By using client-side failover, an NFS client can be aware of multiple servers that are making the same data available and can switch to an alternate server when the current server is unavailable. The file system can become unavailable if one of the following occurs.

- If the file system is connected to a server that crashes
- If the server is overloaded
- If a network fault occurs

The failover, under these conditions, is normally transparent to the user. Thus, the failover can occur at any time without disrupting the processes that are 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 182 for a description of what makes a file system identical. A static file system or a file system that is not changed often is the best candidate for failover.

You cannot use CacheFS and client-side failover on the same NFS mount. 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 every file system depends on many factors. Ideally, you should have a minimum of two servers. Each server should support multiple subnets. This setup is better than having a unique server on each subnet. The process requires that each listed server be checked. Therefore, if more servers are listed, each mount is slower.

Failover Terminology

To fully comprehend the process, you need to understand two terms.

- *failover* – The process of selecting a server from a list of servers that support a replicated file system. Normally, the next server in the sorted list is used, unless it fails to respond.
- *remap* – To use 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 file size or file type as the original file system. Permissions, creation dates, and other file attributes are not considered. If the file size or file types are different, the remap fails and the process hangs until the old server becomes available. In NFS version 4, the behavior is different. See "Client-Side Failover in NFS Version 4" on page 183.

You can maintain a replicated file system by using `rdist`, `cpio`, or another file transfer mechanism. Because updating the replicated file systems causes inconsistency, for best results consider these precautions:

- Renaming the old version of the file before installing a new version of the file
- Running the updates at night when client usage is low
- Keeping the updates small
- Minimizing 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 does not "know" about the locks. Because the files should not change, you do not need to lock the file on the server side.

Client-Side Failover in NFS Version 4

In NFS version 4, if a replica cannot be established because the file sizes are different or the file types are not the same, then the following happens.

- The file is marked dead.
- A warning is printed.
- The application receives a system call failure.

Note – If you restart the application and try again to access the file, you should be successful.

In NFS version 4, you no longer receive replication errors for directories of different sizes. In prior versions of NFS, this condition was treated as an error and would impede the remapping process.

Furthermore, in NFS version 4, if a directory read operation is unsuccessful, the operation is performed by the next listed server. In previous versions of NFS, unsuccessful read operations would cause the remap to fail and the process to hang until the original server was available.

Large Files

The OS supports files that are over 2 Gbytes. By default, UFS file systems are mounted with the `-largefiles` option to support the new capability. See "How to Disable Large Files on an NFS Server" on page 86 for instructions if needed.

If the server's file system is mounted with the `-largefiles` option, a Solaris 2.6 NFS client can access large files without the need for changes. However, not all Solaris 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 that run 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 the following:

- A timestamp
- The client IP address
- The UID of the requester
- The file handle of the file or directory object that is being accessed
- 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 the conversion, the daemon tracks the file handles and stores information in a separate file handle-to-path table. That way, the path does not have to be identified again each time a file handle is accessed. Because no changes to the mappings are made in the file handle-to-path table if `nfslogd` is turned off, you must keep the daemon running.

Note – Server logging is not supported in NFS version 4.

How the WebNFS Service Works

The WebNFS service makes files in a directory available to clients by using a public file handle. A file handle is an address that is 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. This ability should also accelerate processes 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 multi-component lookup transaction when the `LOOKUP` is relative to the public file handle. Multi-component lookup

enables the WebNFS server to deliver the file handle to the desired file without exchanging 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. This connection provides quick access without the additional load on the server that is 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
```

Note – The NFS version 4 protocol is preferred over the WebNFS service. NFS version 4 fully integrates all the security negotiation that was added to the MOUNT protocol and the WebNFS service.

How WebNFS Security Negotiation Works

The NFS service includes a protocol that enables a WebNFS client to negotiate a selected security mechanism with a WebNFS server. The new protocol uses security negotiation multi-component lookup, which is an extension to the multi-component lookup that was used in earlier versions of the WebNFS protocol.

The WebNFS client initiates the process by making a regular multi–component lookup request by 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. This reply indicates that the default mechanism is not valid. The client needs to use a stronger default mechanism.

When the client receives the `AUTH_TOOWEAK` error, the client 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 that are required for the specified path. Depending on the size of the array of security mechanisms, the client might have to make more requests to obtain 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 the client supports. The client then issues a regular multi-component lookup request by using the selected security mechanism to acquire the file handle. All subsequent NFS requests are made by using the selected security mechanism and the file handle.

Note – The NFS version 4 protocol is preferred over the WebNFS service. NFS version 4 fully integrates all the security negotiation that was added to the MOUNT protocol and the WebNFS service.

WebNFS Limitations With Web Browser Use

Several functions that a web site that uses 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 if 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. The WebNFS software does not check inside the file to determine the file type. So, 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, NFS browsing 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 way 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 that makes the request, but not the user. Therefore, 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

attacks that involve the introduction of data. An example is the impersonation of a user by generating the right packets or by recording "conversations" and replaying them later. These attacks affect data integrity. Attacks that involve passive eavesdropping, which is merely listening to network traffic without impersonating anybody, are not as dangerous, because data integrity is not compromised. Users can protect the privacy of sensitive information by encrypting data that is sent 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 system includes an authentication system at the level of the remote procedure call (RPC), which is the mechanism on which the 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 that are 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 that is at minimum as secure as a time-sharing system. In a time-sharing system 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 authentication process is completed. Users can log in on any remote computer just as users can log in on a local terminal. The users' login passwords are their assurance of 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, and birthday. The verifier is the photo that is attached to the badge. You can be sure that the badge has not been stolen by checking the photo on the badge against the person who is carrying the badge. 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, such as 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. However, the verifier contains nothing. Because no verifier exists, a superuser could falsify appropriate credentials by using commands such as `su`. Another problem with UNIX authentication is that UNIX authentication 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 keys and secret keys are stored in the namespace. NIS stores the keys in the public-key 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 about 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 as follows:

- 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 by 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 become unsynchronized 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 that was developed at MIT. Kerberos offers a variety of encryption types, including DES. Kerberos support is no longer supplied as part of Secure RPC, but a server-side and client-side implementation is included in the release. See Chapter 21, "Introduction to the Kerberos Service," in *System Administration Guide: Security Services* for more information about the 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`. Therefore, 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. Otherwise, no way exists to authenticate the replies that are given by the boot server. This limitation could be a serious problem, but the limitation 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, the setuid program might not always work. For example, suppose that 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. The reason is that your secret key is passed to that computer's key server, which then stores your secret key. This process 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 the remote computer requires a password. Instead, use the NFS environment to mount file systems that are 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 succeeds. However, the users cannot access their home directories because no authentication has occurred locally. If the user is prompted for a password, the user has access to his or her home directory if 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. The map is a master list that specifies all the maps that autofs should check. The following example shows what an `auto_master` file could contain.

EXAMPLE 6–3 Sample `/etc/auto_master` File

```
# Master map for automounter
#
+auto_master
/net            -hosts           -nosuid,nobrowse
/home           auto_home        -nobrowse
/-              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 the directory if possible. If the directory exists and is not empty, mounting on the directory hides its contents. In this situation, autofs issues a warning.

The notation `/-` as a mount point indicates that this particular map is a direct map. The notation also means that no particular mount point is associated with the map.

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 by using the search that is specified in the name-service switch configuration file (`/etc/nsswitch.conf`). Special maps are also used for `/net`. See "Mount Point `/net`" on page 191 for more information.

mount-options — *mount-options* is an optional, comma-separated list of options that apply to the mounting of the entries that are 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 that begins with **#** is a comment. All the text 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 that are 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`. The map is a built-in map that uses only the hosts database. Suppose that the computer `gumbo` is in the hosts database and it exports any of its file systems. The following command changes the current directory to the root directory of the computer `gumbo`.

```
% cd /net/gumbo
```

Autofs can mount only the *exported* file systems of host `gumbo`, that is, those file systems on a server that are available to network users instead of those file systems 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.

Direct Autofs Maps

A direct map is an automount point. With a direct map, a direct association exists 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. The following 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 that you want to apply to this particular mount. These options are required only if the options 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. One or more file systems are specified 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. The *pathname* 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 that begins 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`. An entry might appear in `/etc/vfstab` as follows:

```
dancer:/usr/local - /usr/local/tmp nfs - yes ro
```

The equivalent entry appears in a direct map as follows:

```
/usr/local/tmp     -ro     dancer:/usr/local
```

Note – No concatenation of options occurs between the automounter maps. Any options that are added to an automounter map override all options that are listed in maps that are searched earlier. For instance, options that are 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 199 for other important features that are associated with this type of map.

Mount Point /–

In Example 6–3, the mount point `/-` tells autofs not to associate the entries in `auto_direct` with any specific mount point. Indirect maps use mount points that are defined in the `auto_master` file. Direct maps use mount points that are 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 namespace. An `auto_master` file that is a local file can have any number of direct map entries if 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, such as 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 without slashes in an indirect map.

mount-options — *mount-options* is the options that you want to apply to this particular mount. These options are required only if the options 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. One or more file systems are specified as *server*:*pathname*.

Note – The *pathname* should not include an automounted mount point. The *pathname* 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 that begins 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 6–3 shows an `auto_master` map that contains the following 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 the following:

```
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`. Suppose that the user `linda` has an entry in the password database that specifies her home directory as `/home/linda`. Whenever `linda` logs in to computer `oak`, autofs mounts the directory `/export/home/linda` that resides 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 that is 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 occurs between the automounter maps. Any options that are added to an automounter map override all options that are listed in maps that are searched earlier. For instance, options that are 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 allow Linda access to her files. With NIS, make the changes on the NIS master server and propagate the relevant databases to the slave servers. On a network that is 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. The components that work together to accomplish automatic mounting are the following:

- The `automount` command
- The `autofs` file system
- The `automountd` daemon

The automount service, `svc:/system/filesystem/autofs`, which is 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. These mounts 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, these mounts 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.

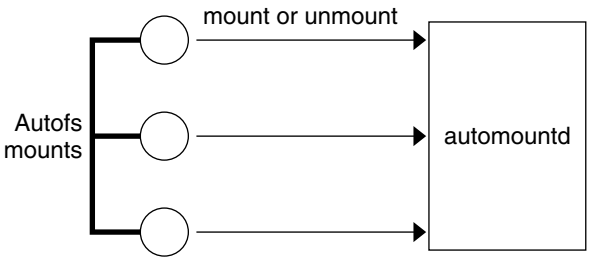
After initially mounting autofs mounts, the `automount` command is used to update autofs mounts as necessary. The command compares the list of mounts in the `auto_master` map with the list of mounted file systems in the mount table file `/etc/mnttab` (formerly `/etc/mtab`). `automount` then makes the appropriate changes. This process allows system administrators to change mount information within `auto_master` and have those changes used by the autofs processes without stopping and restarting 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 namespace or local files.

The following is a simplified overview of how autofs works.

The automount daemon `automountd` is started at boot time by the service `svc:/system/filesystem/autofs`. See Figure 6–3. This service also runs the `automount` command, which reads the master map and installs autofs mount points. See "How Autofs Starts the Navigation Process (Master Map)" on page 197 for more information.

FIGURE 6–3 `svc:/system/filesystem/autofs` Service 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, the following occurs:

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 that are 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 the autofs mount points.

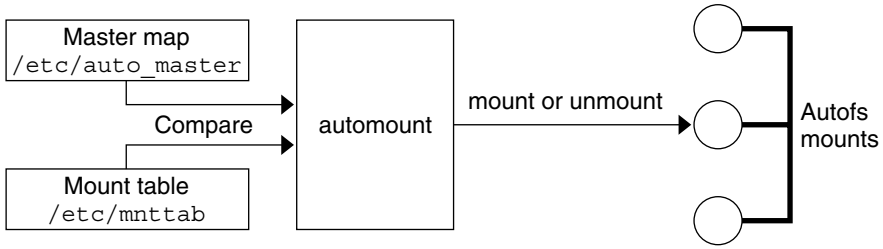
How Autofs Navigates Through the Network (Maps)

Autofs searches a series of maps to navigate 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. Effectively, the maps contain network-wide equivalents of UNIX administration files. Maps are available locally or through a network name service such as NIS or NIS+. See "Modifying How Autofs Navigates the Network (Modifying Maps)" on page 205.

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 map name or an indirect map name, its path, and its mount options, as shown in Figure 6–4. 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 6–4 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. However, the final result changes with the mount point that is specified and the complexity of the maps. The mount process includes 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
/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 is an entry in `/etc/mnttab` 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. Checks the availability of the server's mount service.
2. Mounts the requested file system under `/share`. Now the `/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. Suppose that you expand the `/etc/auto_shared` file from the previous example to contain the following:

```
# 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 198 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 nodes 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. The automounter can complete the unmounting 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)

The example direct map contains the following:

```
/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 locations for the first mount point, two locations for the second mount point. Any of the replicated locations can provide the same service to any user. This procedure is sensible only when you mount a file system that is read-only, as you must have some control over the locations of files that you write or modify. You want to avoid modifying files on one server on one occasion and, minutes later, modifying the "same" file on another server. The benefit is that the best available server is used automatically without any effort that is required by the user.

If the file systems are configured as replicas (see "What Is a Replicated File System?" on page 182), 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.

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 man pages. Which server you mount the man pages from does not matter if 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
```

In this example, 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 following:

- The number of servers that support a particular NFS protocol level
- The proximity of the server
- The weighting

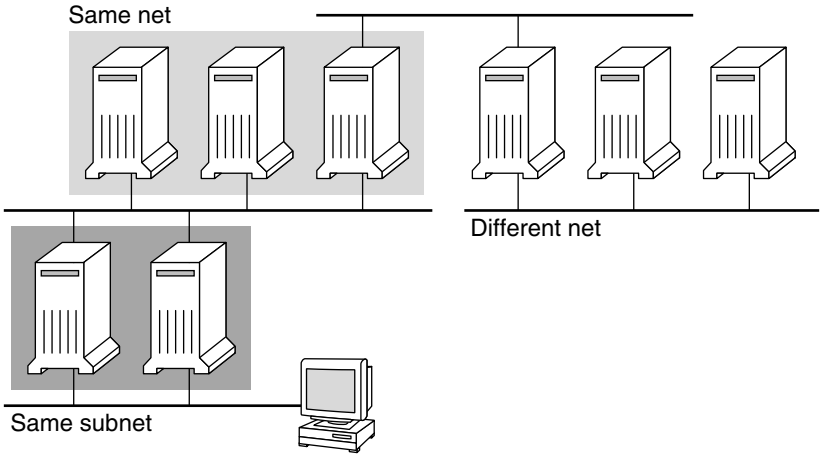
During the sorting process, a count is taken of the number of servers that support each version of the NFS protocol. Whichever version of the protocol is supported on the most servers becomes the protocol that is used by default. This selection provides the client with the maximum number of servers to depend on.

After the largest subset of servers with the same version of the protocol is found, that server list is sorted by proximity. To determine proximity IPv4 addresses are inspected. The IPv4 addresses show which servers are in each subnet. Servers on a local subnet are given preference over servers on a remote subnet. Preference for the closest server reduces latency and network traffic.

Note – Proximity cannot be determined for replicas that are using IPv6 addresses.

Figure 6–5 illustrates server proximity.

FIGURE 6–5 Server Proximity



If several servers that support the same protocol are on the local subnet, the time to connect to each server is determined and the fastest server is used. The sorting can also be influenced by using weighting (see "Autofs and Weighting" on page 202).

For example, if version 4 servers are more abundant, version 4 becomes the protocol that is used by default. However, now the sorting process is more complex. Here are some examples of how the sorting process works:

- Servers on the local subnet are given preference over servers on a remote subnet. So, if a version 3 server is on the local subnet and the closest version 4 server is on a remote subnet, the version 3 server is given preference. Likewise, if the local subnet consists of version 2 servers, they are given preference over remote subnets with version 3 and version 4 servers.
- If the local subnet consists of a varied number of version 2, version 3, and version 4 servers, more sorting is required. The automounter prefers the highest version on the local subnet. In this instance, version 4 is the highest version. However, if the local subnet has more version 3 or version 2 servers than version 4 servers, the automounter "bids down" from the highest version on the local subnet by one version. For example, if the local subnet has three servers with version 4, three servers with version 3, and ten servers with version 2, a version 3 server is selected.
- Similarly, if the local subnet consists of a varied number of version 2 and version 3 servers, the automounter first looks at the which version represents the highest version on the local subnet. Next, the automounter counts the number of servers that run each version. If the highest version on the local subnet also represents the most servers, the highest version is selected. If a lower version has more servers, the automounter bids down from the highest version on the local subnet by one version. For example, if more version 2 servers are on the local subnet than version 3 servers, a version 2 server is selected.

Note – Weighting is also influenced by keyword values in the `/etc/default/nfs` file. Specifically, values for NFS_SERVER_VERSMIN, NFS_CLIENT_VERSMIN, NFS_SERVER_VERSMAX, and NFS_CLIENT_VERSMAX can make some versions be excluded from the sorting process. For more information about these keywords, see "Keywords for the `/etc/default/nfs` File" on page 131.

With failover, the sorting is checked at mount time when a server is selected. Multiple locations are useful in an environment where individual servers might not export their file systems temporarily.

Failover is particularly useful in a large network with many subnets. Autofs chooses the appropriate server and is able to confine NFS network traffic to a segment of the local network. If a server has multiple network interfaces, you can list the host name that is associated with each network interface as if the interface were a separate server. Autofs selects the nearest interface to the client.

Note – No weighting and no proximity checks are performed with manual mounts. The `mount` command prioritizes the servers that are listed from left to right.

For more information, see `automount(1M)` man page.

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 and, therefore, are most likely to be selected. The higher the weighting value, the lower the chance that the server is 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. The variable helps you to accommodate different architecture types that are 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 6–2 shows the predefined map variables.

TABLE 6–2 Predefined Map Variables

Variable	Meaning	Derived From	Example
`ARCH`	Architecture type	`uname -m`	`sun4`
`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`
`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, suppose that you have a file server that exports binaries for SPARC and x86 architectures from `/usr/local/bin/sparc` and `/usr/local/bin/x86` respectively. The clients can mount through a map entry such as 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 that are written for any of the sun4 architectures can run on all sun4 platforms. The `-ARCH` variable is hard-coded to `sun4`.

Maps That Refer to Other Maps

A map entry +*mapname* that is 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 the map name. 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 `hosts`.

This name-service switch file contains an entry for autofs that is 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. Therefore, you can have a few entries in your local `/etc/auto_home` map for the most commonly accessed home directories. You can then 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. Therefore, 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 that is included can be a local file or a built-in map. Remember, only local files can contain + entries.

```
+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 executes 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 needs 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 causes the `/export1` file system from `bee` to be mounted:

```
% ls /execute/src
```

Modifying How Autofs Navigates the Network (Modifying Maps)

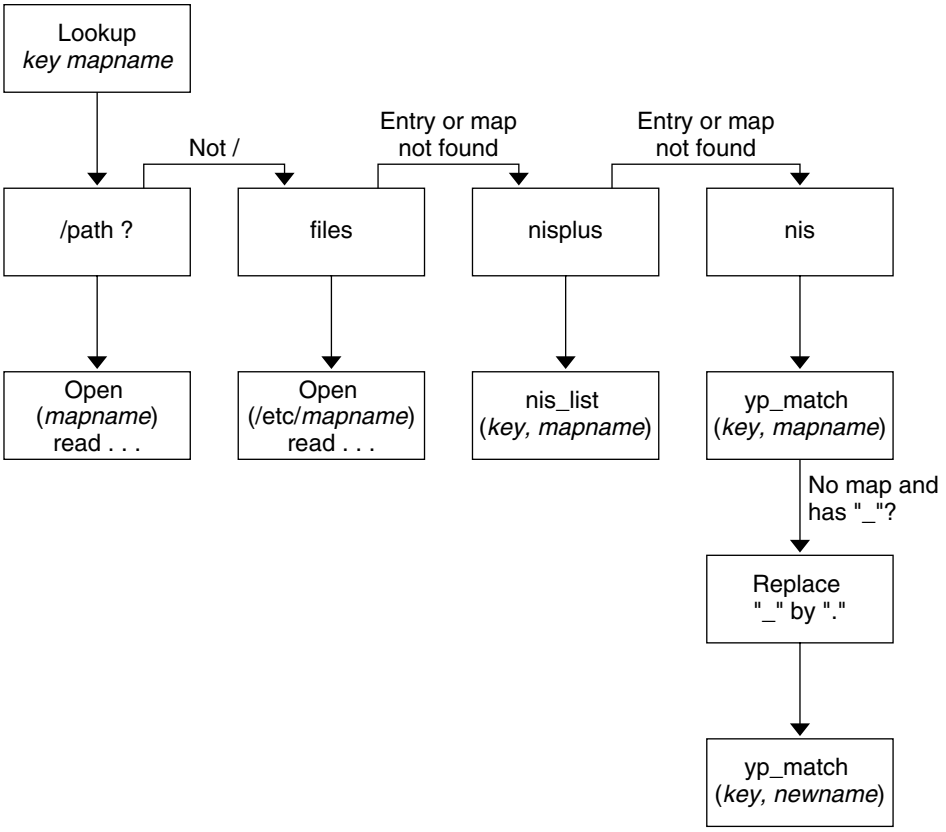
You can modify, delete, or add entries to maps to meet the needs of your environment. As applications and other file systems that users require change their location, the maps must reflect those changes. You can modify autofs maps at any time. Whether your modifications are effective the next time `automountd` mounts a file system depends on which map you modify and what kind of modification you make.

Default Autofs Behavior With Name Services

At boot time autofs is invoked by the service `svc:/system/filesystem/autofs` and autofs checks for the master `auto_master` map. Autofs is subject to the rules that are discussed subsequently.

Autofs uses the name service that is 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 autofs needs, but finds a map name that contains one or more underscores, the underscores are changed to dots. This change allows the old NIS file names to work. Then autofs checks the map again, as shown in Figure 6–6.

FIGURE 6–6 How Autofs Uses the Name Service



The screen activity for this session would resemble 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 autofs uses.

Autofs Reference

The remaining sections of this chapter describe more advanced autofs features and topics.

Autofs and Metacharacters

Autofs recognizes some characters as having a special meaning. Some characters are used for substitutions, and some characters are used 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 the key appears. If you use the ampersand, the previous map changes to the following:

```
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 such as the following:

```
/usr/man                        willow,cedar,poplar:/usr/man
```

You can also simplify the entry further as follows:

```
/usr/man                        willow,cedar,poplar:&
```

Notice that the ampersand substitution uses the whole key string. Therefore, if the key in a direct map starts with a / (as it should), the slash is included in the substitution. Consequently, for example, you could not do the following:

```
/progs                &1,&2,&3:/export/src/progs
```

The reason is that autofs would interpret the example as the following:

```
/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.

Autofs and Special Characters

If you have a map entry that contains special characters, you might have to mount directories that have names that confuse the autofs map parser. The autofs parser is sensitive to names that contain colons, commas, and spaces, for example. These names should be enclosed in double-quotes, as in the following:

```
/vms    -ro    vmsserver: -  -  - "rc0:dk1 - "
/mac    -ro    gator:/ - "Mr Disk - "
```

PART III

SLP Topics

The section provides overview, planning, task and reference information for the Service Location Protocol (SLP) service.

◆ ◆ ◆ CHAPTER 7

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 implementation of SLP for IP intranets.

SLP Architecture

This section outlines the fundamental operation of SLP and describes agents and processes that are used in SLP administration.

SLP provides all of the following services automatically, with little or no configuration.

- Client application requests for information that is required to access a service
- Advertisement of services on network hardware devices or software servers; for example, printers, file servers, video cameras, and HTTP servers
- Managed recovery from primary server failures

In addition, you can do the following to administer and tune SLP operation if necessary.

- Organize services and users into *scopes* that are composed of logical or functional groups
- Enable SLP logging to monitor and troubleshoot the SLP operation on your network
- Adjust SLP timing parameters to enhance performance and scalability
- Configure SLP not to send and not to process multicast messages when SLP is deployed on networks that lack support for multicast routing
- Deploy SLP Directory Agents to improve scalability and performance

Summary of the SLP Design

SLP libraries inform network-aware agents that advertise services in order for those services to be discovered over a network. SLP agents maintain up-to-date information on the type and location of services. These agents can also use proxy registrations to advertise services that are not directly SLP enabled. For more information, see Chapter 10, "Incorporating Legacy Services."

Client applications rely on SLP libraries that make requests directly to the agents that advertise services.

SLP Agents and Processes

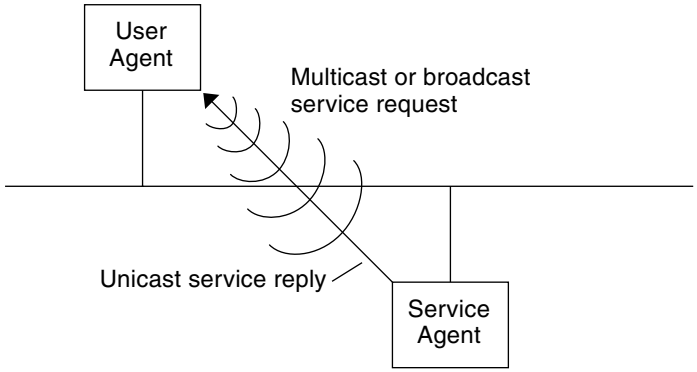
The following table describes the SLP agents. For expanded definitions of these terms and other terms that are used in this volume, refer to the Glossary.

TABLE 7–1 SLP Agents

SLP Agent	Description
Directory Agent (DA)	Process that caches SLP advertisements that are 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.

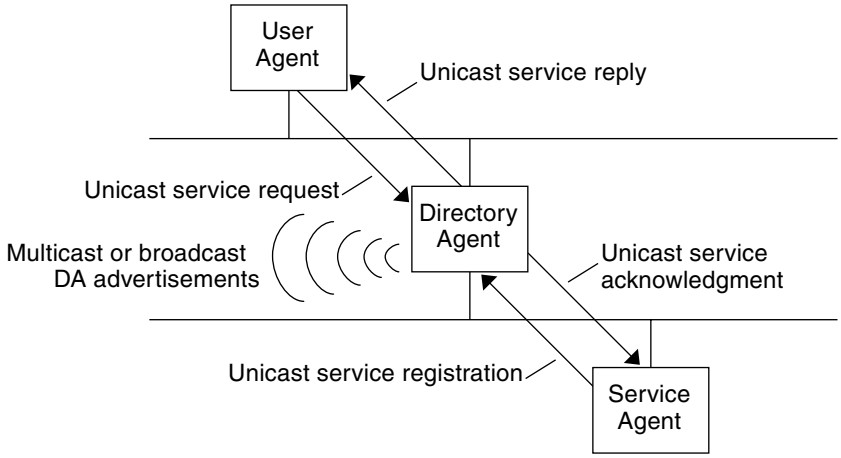
The following figure shows the basic agents and processes that implement the SLP architecture. The figure represents a default deployment of SLP. No special configuration has been done. Only two agents are required: the UA and SA. The SLP framework allows the UA to multicast requests for services to the SA. The SA unicasts a reply to the UA. For example, when the UA sends a service request message, the SA responds with a service reply message. The service reply contains the location of services that match the client's requirements. Other requests and replies are possible for attributes and service types. For more information, see Chapter 11, "SLP (Reference)."

FIGURE 7–1 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 7–2 SLP Architectural Agents and Processes Implemented With a DA



When you deploy DAs, fewer messages are sent in the network and UAs can retrieve information much faster. DAs are essential when the size of a network increases or for situations in which there is no support for multicast routing. 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 that is set for the advertisement. After 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 11, "SLP (Reference)."

SLP Implementation

In the Solaris SLP implementation, the SLP SAs, UAs, DAs, SA servers, scopes, and other architectural components in Table 7–1 are partially mapped into `slpd` and partially into application processes. The SLP daemon, `slpd`, organizes certain off-host SLP interactions to do the following:

- Employ passive and active directory agent discovery in order to discover all DAs on the network
- Maintain an updated table of DAs for the use of the UAs and SAs on the local host
- Act as a proxy SA server for legacy service advertisements (proxy registration)

You can set the `net.slpisDA` property to also configure `slpd` to act as a DA. See Chapter 9, "Administering SLP (Tasks)."

For more information about the SLP daemon, see `slpd(1M)`.

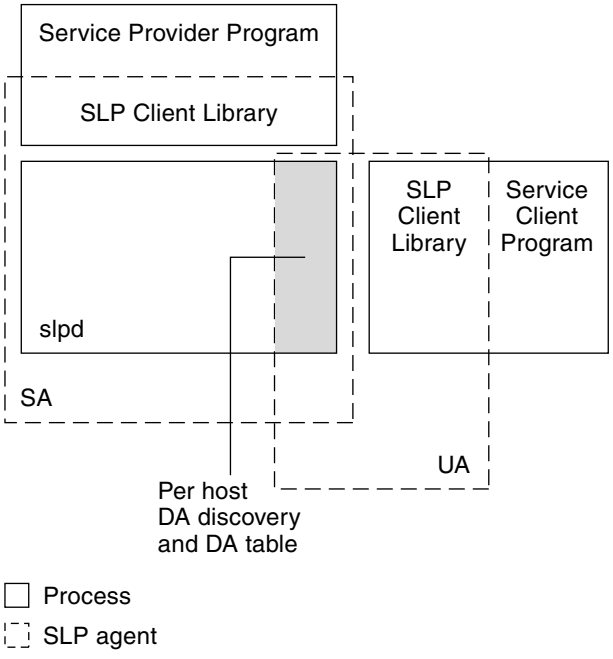
In addition to `slpd`, the C/C++ and Java client libraries (`libslp.so` and `slp.jar`) enable access to the SLP framework for UA and SA clients. The client libraries provide the following features:

- Software that offers network services which can register and deregister service advertisements
- Client software that can request services by issuing queries for service advertisements
- The list of SLP scopes available for registration and requests

No special configuration is necessary to enable the inter-process communication between `slpd` and the client libraries that provide the previous services. You must, however, run the `slpd` process first before you load the client libraries in order for the libraries to function.

In the following figure, the SLP client library in the Service Provider Program employs SA functionality. The Service Provider Program uses the SLP client library to register and deregister services with `slpd`. The SLP client library in the Service Client Program employs UA functionality. The Service Client Program uses the SLP client library to make requests. The SLP client library either multicasts requests to SAs or unicasts them to DAs. This communication is transparent to the application except that the unicast method of issuing requests is faster. The behavior of the client library can be affected by setting various SLP configuration properties. For further information, see Chapter 9, "Administering SLP (Tasks)." The `slpd` process handles all SA functionality, such as answering multicast requests and registering with DAs.

FIGURE 7–3 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*. John Wiley & Sons, Inc. ISBN Number: 0–471–31587–7.
- *Authentication Management Infrastructure Administration Guide*. Part Number: 805–1139–03.
- Guttman, Erik, Charles Perkins, John Veizades, and Michael Day. *Service Location Protocol, Version 2, RFC 2608* from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2608.txt]
- Kempf, James, and Erik Guttman. *An API for Service Location, RFC 2614* from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2614.txt]

◆ ◆ ◆ CHAPTER 8

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 preconfigured with default properties. If your enterprise functions well with default settings, the SLP deployment requires virtually no administration.

In some situations, 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 a 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, which is located in the `/etc/inet` directory. If you decide to change the default property settings, refer to Chapter 9, "Administering SLP (Tasks)," for the appropriate procedures.

Before you modify SLP configuration settings, consider the following questions that are related to key aspects of network administration:

- What network technologies are operating in the enterprise?
- How much network traffic can the technologies handle smoothly?
- How many services, of what type, are available on the network?
- 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 reconfiguration is necessary and what properties you need to modify. 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 which 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 if network or agent reconfiguration is needed:

- The number of hosts that are using a particular DA. Use this information to decide whether to deploy additional DAs for load balancing.
- The number of hosts that are using a particular DA. Use this information to help you determine whether to configure certain hosts with new or different scopes.
- Whether UA requests a timeout or DA acknowledgment is slow. You can determine whether a DA is overloaded by monitoring UA timeouts and retransmissions. You can also check 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Run 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:

   - The 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 the 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 simplify the description of 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`, which is 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 that is 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*.
6. Shows *slphost2* responding to the *slphost1* `SrvReg` with a service acknowledgment (`SrvAck`) that indicates 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 absence of information is because the `snoop` operation is performed over the network loopback.
7. Shows the echo server on *slphost1* deregistering the echo service advertisement. The SLP daemon on *slphost1* forwards the deregistration to the DA on *slphost2*.
8. Shows *slphost2* responding to the *slphost1* with a service acknowledgment (`SrvAck`) that indicates that the deregistration is successful.

   The `/tcp` parameter that is appended to the message number on lines 5, 6, 7, and 8 indicates that the message exchange occurred by TCP.

Where to Go From Here

After monitoring the SLP traffic, you can use the information that was collected from the `snoop` traces to help determine whether any reconfiguration of the SLP defaults is needed. Use the

related information in Chapter 9, "Administering SLP (Tasks)," for configuring SLP property settings. For more information about SLP messaging and service registrations, refer to Chapter 11, "SLP (Reference)."

◆ ◆ ◆ CHAPTER 9

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 situations, 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 achieve the following goals:

- 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 9–1 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`.

TABLE 9–1 SLP Configuration Operations *(Continued)*

Operation	Description
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

If the value defined is not allowed, the default value for that property name is used. In addition, an error message is logged using syslog.

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. For more information, refer to: Guttman, Erik, James Kempf, and Charles Perkins, "Service Templates and service: scheme," RFC 2609 from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2609.txt]

How to Change Your SLP Configuration

Use this procedure to change the property settings in your SLP configuration file. SLP– enabled client or service software also can alter the SLP configuration by using the SLP API. This API is documented in "An API for Service Location," RFC 2614 from the Internet Engineering Task Force (IETF). [http://www.ietf.org/rfc/rfc2614.txt]

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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 224 for general information about the SLP property settings. See the sections that follow this procedure 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.**

   ```
   # svcadm enable network/slp
   ```

Note – The SLP daemon obtains information from the configuration file when you stop or start `slpd`.

Example 9–1 Setting up `slpd` to Operate as a DA Server

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 that are 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 dial-up 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 9–2 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

Sometimes 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` 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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 setting 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.**

   ```
   # svcadm enable network/slp
   ```

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 that is described in "Limiting UAs and SAs to Statically Configured DAs" on page 226.

▼ 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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 that is 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 that is 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.**

   ```
   # svcadm enable network/slp
   ```

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 when it boots. The SLP daemon uses the periodic unsolicited advertisement (the heartbeat) to determine whether a DA is still active. If the heartbeat fails to appear, the daemon removes the DAs the daemon uses and those the daemon offers to UAs.

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.

Note – If DA discovery is completely disabled, the `net.slp.DAAddresses` property must be set in `slp.conf` on the hosts that are executing UAs and SAs so that they access the correct DA.

---

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

3. **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4. **Decrease the `net.slp.DAHeartBeat` value to 1 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.**

```
# svcadm enable network/slp
```

Relieving Network Congestion

If network congestion is high, you can limit the amount of multicast activity. If DAs have not already been deployed in the network, deploying DAs can drastically reduce 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 by using the method that is described in "How to Configure DA Discovery for Dial-up Networks" on page 227. You can completely eliminate multicast for DA discovery by using the method that is described in "Limiting UAs and SAs to Statically Configured DAs" on page 226.

Accommodating Different Network Media, Topologies, or Configurations

This section describes possible scenarios in which you can change the following properties to tune SLP performance.

TABLE 9–3 SLP Performance Properties

Property	Description
`net.slp.DAAttributes`	The minimum refresh interval that a DA accepts for advertisements.
`net.slp.multicastTTL`	The *time-to-live* value that is 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`	The Boolean that is 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 that are registered with a service registration. If the lifetimes are short and reregistrations are occurring often, the timeouts are probably the result of frequent reregistrations.

Note – A service registration is a *reregistration* if the FRESH flag is not set. See Chapter 11, "SLP (Reference)," 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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.**

   ```
   # svcadm enable network/slp
   ```

Configuring the Multicast Time-to-Live Property

The multicast time–to-live property (`net.slp.multicastTTL`) 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 with the standard IP TTL. If the TTL becomes zero, the packet is discarded. When you use TTL scoping for SLP multicasting, your routers must 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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, if the routers are properly configured with TTLs.

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

6. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

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, you should not have to change the `net.slp.MTU` value. 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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.**

```
# svcadm enable network/slp
```

Configuring Broadcast-Only Routing

SLP is designed to use multicast for service discovery in the absence of DAs and for DA discovery. 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 on networks in which 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 248 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Stop `slpd` and all SLP activity on the host.**

```
# svcadm disable network/slp
```

3 **Back up the default `/etc/inet/slp.conf` file before you change the configuration settings.**

4 **Change the `net.slp.isBroadcastOnly` property in the `slpd.conf` file to `True`:**

```
net.slp.isBroadcastOnly=True
```

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

6 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Modifying Timeouts on SLP Discovery Requests

Two situations might require that you 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 be completed. Conversely, if your network is low latency, you can improve performance 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 an 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 whether SLP requests are failing because of timeouts on UA and SA requests and registrations. You can also use the `ping` command to measure the actual latency.

The following table lists configuration properties that control timeouts. You can use the procedures in this section to modify these properties.

TABLE 9–4 Time-out 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 the transmission is abandoned.

`net.slp.multicastMaximumWait`	The property that controls the maximum amount of time a multicast message is transmitted before it is abandoned.
`net.slp.datagramTimeouts`	The upper bound of a DA timeout that is specified by the sum of values that are listed for this property. A UDP datagram is repeatedly sent to a DA until a response is received or the time-out bound is reached.

If frequent timeouts are occurring during multicast service discovery or DA discovery, increase the `net.slp.multicastMaximumWait` property from the default value of 15000 milliseconds (15 seconds). Increasing the maximum wait period allows more time for requests on high latency

networks to be completed. 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 for these properties 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Stop `slpd` and all SLP activity on the host.**

```
# svcadm disable network/slp
```

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 that are 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 that are 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 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 time-out bound for multicast and unicast UDP datagram transmission. When you reduce the time-out bound, you decrease latency that is required to satisfy SLP requests.

6 **Save your changes and close the file.**

7 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

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 that are running `slpd` as an SA server and a host that is running `slpd` as a DA. If multiple service registration messages for the same service appear in the `snoop` trace from the host that is running `slpd` as an SA server, you might have notice collisions.

Collisions can be particularly troubling at boot time. When a DA first starts, it sends 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 that is 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Stop `slpd` and all SLP activity on the host.**

```
# svcadm disable network/slp
```

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 2000 milliseconds (2 seconds).

```
net.slp.randomWaitBound=2000
```

When you lengthen the random-wait bound, a longer delay in registration occurs. 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 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 time-out bound for multicast and unicast UDP datagram transmission. This setting reduces the amount of latency in satisfying SLP requests.

6 **Save your changes and close the file.**

7 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Deploying Scopes

With scopes, you can provision services that depend on the logical, physical, and administrative groupings 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
```

Your organization might, for example, have 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 could be used by everyone on the second floor, or you might restrict the usage to members of a certain department. Scopes provide a way for you 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 that are advertised in other scopes. Configuring scopes controls only which service advertisements a UA detects. The service is responsible for enforcing any access control restrictions.

When to Configure Scopes

SLP can function adequately without any scope configuration. 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.

- The organizations you support want to restrict service advertisement access to their own members.
- The physical layout of the organization you support suggests that services in a certain area be accessed by particular users.
- The service advertisements that are appropriate for specific users to see must be partitioned.

An example of the first circumstance was cited in "Configuring DA Discovery for Dial-up Networks" on page 227. An example of the second 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. You can configure users in Building 1 with the `B1` scope, while you configure users in Building 2 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 can automatically discover available DAs and scopes through `slpd`. The SLP daemon uses active and passive DA discovery to find DAs, or it uses SA discovery if no DAs are running. Alternatively, if the property is configured, UAs use only the configured scopes and do 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 are unable to find them. This situation occurs 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 that are supported by the configured DAs are the same as the scopes that you have configured with the `net.slp.useScopes` property. If the scopes are different, `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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Stop `slpd` and all SLP activity on the host.

```
# svcadm disable network/slp
```

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 indicate which scopes a DA or SA is allowed to use when making requests, or which 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, uppercase or lowercase
- 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 the following.

```
net.slp.useScopes=eng,mktg,bldg6
```

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

6 **Restart `slpd` to activate your changes.**

```
# svcadm enable network/slp
```

Deploying DAs

This section describes the strategic deployment of DAs in a network that is running SLP.

SLP functions adequately with only the base agents (UAs and SAs), and with no deployed DAs or configured scopes. All agents that lack specific configurations use the `default` scope. DAs serve as caches for service advertisements. Deploying DAs decreases the number of messages that are sent on the network and reduces the time that is required to receive responses to messages. 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 and the delays that are associated with gathering unicast replies. In a large network with many UAs and SAs, the amount of multicast traffic that is 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 by 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 accepting 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, the time that is required to obtain replies to queries is greatly reduced (from seconds to milliseconds).

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 by checking the logs from multiple SAs that are 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 routing support, services can be discovered only on the same subnet. Partial support for multicast routing leads to inconsistent ability to discover services on a network. Multicast messages are used to discover DAs. Partial support for multicast routing, therefore, implies that UAs and SAs register services with all known DAs in the SA's scope. For example, if a UA queries a DA that is called `DA1` and the SA has registered services with `DA2`, the UA will fail to discover a service. See "Configuring Broadcast-Only Routing" on page 234 for more information on how to deploy SLP on networks without multicast enabled.

On a network with inconsistent site-wide support for multicast routing, you must configure the SLP UAs and SAs with a consistent list of DA locations by using the `net.slp.DAAdresseses` property.

Finally, the SLPv2 DA supports interoperability with SLPv1. SLPv1 interoperability is enabled by default in the DA. If your network contains SLPv1 devices, such as 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 percent of the bandwidth on your network, as measured by `snoop`.
- UA clients experience long delays or timeouts during multicast service requests.
- You want to centralize the 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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.**

   ```
   # svcadm enable network/slp
   ```

Where to Place DAs

This section provides suggestions for where to place DAs in different situations.

- When multicast routing is not enabled and DAs are required to bridge service discovery between subnets

  In this situation, 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 for SLP" on page 245 for more information on configuring the `net.slp.interfaces` property.
- When DAs are deployed for scalability and 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 infrequent 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.
- Situating DAs so that they are topologically close to UAs on the network to optimize UA access

  Naturally, you must configure the DA with a scope that is 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. Deploy DAs 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 who share services in the scopes spans a number of buildings or physical sites.

You can run 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, UAs in buildings physically and topologically distant from the DA can exhibit slow response or excessive timeouts. In such a scenario, you can deploy a DA in each building to improve response for UA clients within the building.

Links that connect 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 into two a scope that has many registrations. You can then support one of the new scopes by deploying another DA on another host.

SLP and 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 situations, routing between interfaces is disabled. The following sections describe how to configure SLP for such situations.

Multihoming Configuration for SLP

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. This is 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, configuration is required.

When to Configure for Nonrouted, Multiple Network Interfaces

If one of the following conditions exist, you might need to configure multihomed machines.

- 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 situation, broadcast is normally used for service discovery that is not DA-based and for DA discovery on the individual subnets. Broadcast is configured by setting the `net.slp.isBroadcastOnly` property to `True`.

Configuring Nonrouted, Multiple Network Interfaces (Task Map)

TABLE 9–5 Configuring Nonrouted, Multiple Network Interfaces

Task	Description	For 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 246
Arrange proxy service advertisements so that UAs on subnets get service URLs with reachable addresses	Restrict proxy advertisement to a machine that is running `slpd` connected to a single subnet rather than a multihomed host.	"Proxy Advertising on Multihomed Hosts" on page 248
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. Run a DA on a multihomed host, but configure scopes so that SAs and UAs on each subnet use different hosts.	"DA Placement and Scope Name Assignment" on page 248

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 that are 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, a multicast request can arrive at `slpd` from more than one interface. This situation can occur when the routing of packets is handled by another multihomed host or router that connects the subnets that are served by the interfaces.

When such a situation occurs, the SA server or the UA that is sending the request receives two responses from `slpd` on the multihomed host. The responses are then filtered by the client libraries and the client does not see them. The responses are, however, visible in the `snoop` trace.

Note –

If unicast routing is turned off, services that are 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. The service program, which might or might not handle a multihomed host with no routing, is then responsible for assuring that reachable URLs are advertised.

Before you deploy a service on a multihomed host with unicast routing disabled, use `snoop` to determine whether the service handles requests from multiple subnets correctly. Furthermore, if you plan to deploy a DA on the multihomed host, see "DA Placement and Scope Name Assignment" on page 248.

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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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 that is 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.**

```
# svcadm enable network/slp
```

Proxy Advertising on Multihomed Hosts

If a host with multiple interfaces advertises services by using `slpd` and proxy registration, the service URLs that are advertised by `slpd` must contain reachable host names or addresses. If unicast routing is enabled between the interfaces, hosts on all subnets can reach hosts on other subnets. Proxy registrations can also be made for a service on any subnet. If, however, unicast routing is disabled, service clients on one subnet cannot reach services on another subnet through the multihomed host. However, those clients might be able to reach the services 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 and that the URL `service:printing:lpr://oldprinter/queue1` is configured with the `net.slp.interfaces` to listen on all interfaces. The `oldprinter` URL is proxy-advertised on all interfaces. The machines on the `142` and `144` subnets receive the URL in response to service requests, but are unable to access the `oldprinter` service.

The solution to this problem is to perform the proxy advertisement with `slpd` running on a machine that is connected to the `143` subnet only, rather than on the multihomed host. Only hosts on the `143` subnet can obtain the advertisement in response to a service request.

DA Placement and Scope Name Assignment

The placement of DAs and assignment of scope names on a network with a multihomed host must be done carefully to assure that clients obtain accessible services. Be particularly cautious when routing is disabled and the `net.slp.interfaces` property is configured. Again, if unicast routing is enabled between the interfaces on a multihomed machine, no special DA and scope configuration is necessary. The advertisements are 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 problems can result 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 situation, 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 disadvantage 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 so that the SAs and UAs on each subnet have a different scope. For example, in the previous situation, UAs and SAs on the `142` subnet might have a scope that is called `scope142`. UAs and SAs on the `143` subnet might have another scope that is called `scope143` and UAs and SAs on the `144` subnet could have third scope that is called `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

Configuring the `net.slp.interfaces` property enables a DA on the multihomed host to bridge service advertisements between the subnets. Such configuration is useful if multicast routing is turned off in the network, but unicast routing between interfaces on a multihomed host is enabled. 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 that were made on the same subnet, so they cannot locate services off of their subnet.

The most common situation that necessitates configuration of the `net.slp.interfaces` property occurs when multicast is not deployed on the network and broadcast is used instead. Other situations require careful thought and planning to avoid unnecessary duplicate responses or unreachable services.

◆ ◆ ◆ CHAPTER 10

Incorporating Legacy Services

Legacy services are network services that predate the development and implementation of SLP. Services such as the line printer daemon (`lpsched`), the NFS file service, and the 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 networks. You can find 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 be required.

Advertising Legacy Services

You use any of the following methods to advertise legacy services.

- Modify the service to incorporate an SLP SA.
- Write a small program that advertises on behalf of a service that is not SLP enabled.
- Use proxy advertising to have `slpd` advertise the service.

Modifying the Service

If the source code for the software server is available, you can 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.

Advertising a Service That Is Not SLP Enabled

If the source code or an updated PROM that contains SLP is not available, you can write a small application that uses the SLP client library to advertise the service. This application could function as a small daemon that you start or stop from the same shell script you use to start and stop the service.

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. **Determine 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.
   - 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 that is called *BizApp* has a local database that is 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 with the location of the registration file that was created in the previous steps.**

4. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

5. **Stop `slpd` and all SLP activity on the host.**

   ```
   # svcadm disable network/slp
   ```

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 following:

   ```
   net.slp.serializedRegURL=file:/etc/inet/slp.reg
   ```

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

9. **Restart `slpd` to activate your changes.**

   ```
   # svcadm enable network/slp
   ```

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)
(3) service:lpr://bizserver/mainspool,en,65535
(4) scope=eng,corp
(5) make-model=Laserwriter II
(6) location-description=B16-2345
(7) color-supported=monochromatic
(8) fonts-supported=Courier,Times,Helvetica 9 10
(9)
(10) #Advertise FTP server
(11)
(12) ftp://archive/usr/src/public,en,65535,src-server
```

(13) `content=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 10–1 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. All characters through the end of a comment line are ignored.
2, 9, and 14	Blank lines that delimit the advertisements.
3, 12	Service URLs that each have three required fields and one optional field that are separated by commas: ■ Generic or `service:` `URL` advertised. See RFC 2609 for the specification of how to form a service: URL. ■ Language of the advertisement. In the previous example, the field is designated English, *en*. Language is an RFC 1766 language tag. ■ 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 – If used, this field defines the service type. If the service URL is defined, you can 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 that are configured for the host should be included in the 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.

TABLE 10–1 SLP Proxy Registration File Description *(Continued)*

Line Numbers	Description
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 writing a SLP-enabled service that uses the SLP API to advertise on behalf of other services. Modifying the source code is also preferable to using proxy registration. When you modify the source code, you can add service-specific features and closely track service availability. If the source code is unavailable, writing an SLP-enabled helper service that advertises on behalf of other services is preferable to using proxy registration. Ideally, this helper service is integrated into the service start/stop procedure that is used to control activation and deactivation. 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 if `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 shut 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 disadvantage of proxy advertisements.

◆ ◆ ◆ CHAPTER 11

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 with descriptions and function codes that are used to indicate that a 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 11–1 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 that is indicated.
PARSE_ERROR	2	The message fails to follow 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.
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.

TABLE 11–1 SLP Status Codes *(Continued)*

Status Type	Status Code	Description
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 had no remaining file space.
DA_BUSY_NOW	11	The UA or SA should retry, using exponential backoff. 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 11–2 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.

TABLE 11–2 SLP Message Types *(Continued)*

Message Type	Abbreviation	Function Code	Description
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 that is 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.

PART IV

Mail Services Topics

This section provides overview, task, and reference information for the mail service.

◆ ◆ ◆ CHAPTER 12

Mail Services (Overview)

Setting up and maintaining an electronic mail service involves complex tasks that are critical to the daily operation of your network. As a network administrator, you might need to expand an existing mail service. Alternately, you might need to set up a mail service on a new network or a subnet. The chapters on mail services can help you plan and set up a mail service for your network. This chapter provides links to descriptions 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.

See Chapter 13, "Mail Services (Tasks)," for procedural information about how to set up and administer mail services. For details, refer to "Task Map for Mail Services" on page 269.

See Chapter 14, "Mail Services (Reference)," 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, the interactions of `sendmail` with name services, and the features in version 8.13 of `sendmail`. See "Changes in Version 8.13 of `sendmail`" on page 350.

What's New With Mail Services

This section provides information about new features in various Solaris releases.

Changes in this Release

The following changes have been made in the Oracle Solaris 10 Update 10 release:

- The default version of sendmail has been updated to 8.14.
- The sendmail instance was split into two instances to provide better management of the traditional daemon (`svc:/network/smtp:sendmail`) and the client queue runner (`svc:/network/smtp:sendmail-client`).
- The system can be configured to automatically rebuild the `sendmail.cf` and the `submit.mc` configuration files. The required steps are documented in "How to Automatically Rebuild a Configuration File" on page 284.
- By default, the sendmail daemon runs in the new local daemon mode. The local–only mode only accepts incoming mail from the local host or loopback SMTP connections. For instance, mail from a cron job or between local users would be accepted. Outbound mail is routed as expected, only the incoming mail is changed. The `-bl` option is used to select the local-only mode, also known as the Become Local mode. For more information about this mode, see the `sendmail(1M)` man page. For instructions on how to change back to the `-bd` or Become Daemon mode, see "How to Use `sendmail` in the Open Mode" on page 284.

Changes in the Solaris 10 1/06 Release

Starting in the Solaris 10 1/06 release, `sendmail` supports SMTP using Transport Layer Security (TLS). For more information, see the following:

- "Support for Running SMTP With TLS in Version 8.13 of `sendmail`" on page 351
- "How to Set SMTP to Use TLS" on page 285

For a complete list of features in the Solaris 10 1/06 release, see *Oracle Solaris 10 8/11 What's New*.

Changes in the Solaris 10 Release

`sendmail` version 8.13 is the default. For information about version 8.13 and other changes, see the following:

- "Flags Used and Not Used to Compile `sendmail`" on page 316
- "`MILTER`, Mail Filter API for `sendmail`" on page 317
- "Versions of the Configuration File" on page 318
- "Enhancement for `vacation` Utility" on page 329
- "Contents of the `/etc/mail/cf` Directory" on page 331
- "Changes in Version 8.13 of `sendmail`" on page 350
- "Support for TCP Wrappers From Version 8.12 of `sendmail`" on page 359

Additionally, the mail service is managed by the Service Management Facility. Administrative actions on this service, such as enabling, disabling, or restarting, can be performed by using the `svcadm` command. The service's status can be queried by using the `svcs` command. For more information about the Service Management Facility, see the `smf(5)` man page and Chapter 18, "Managing Services (Overview)," in *System Administration Guide: Basic Administration*.

Other `sendmail` Information Sources

The following is a list of additional information sources about `sendmail`.

- Costales, Bryan. *sendmail, Third Edition*. O'Reilly & Associates, Inc., 2002.
- 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 versions of `sendmail` – `http://www.sendmail.org/vendor/sun/`.

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. These sections also provide some of the terms that are used to describe the components.

The first section, "Overview of the Software Components" on page 265, 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 266, 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 319 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

Component	Description
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 to 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

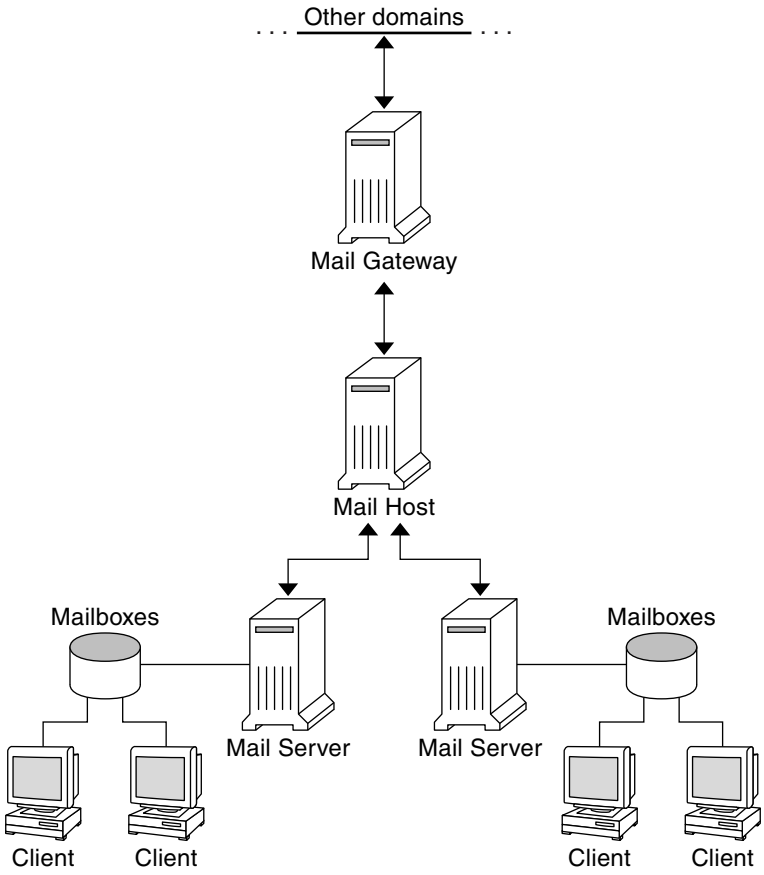
A mail configuration requires three elements, which you can combine on the same system or provide in separate systems.

- A mail host – A system that is configured to handle email addresses that are difficult to resolve
- A minimum of 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 12–1 shows a typical electronic mail configuration, using the three basic mail elements plus a mail gateway.

FIGURE 12–1 Typical Electronic Mail Configuration



Each element is described in detail in "Hardware Components" on page 326.

◆ ◆ ◆ CHAPTER 13

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 12, "Mail Services (Overview)," for an introduction to the components of mail services. This chapter also provides a description of a typical mail service configuration, as shown in Figure 12–1. The following list can help you find groups of related procedures that are covered in this chapter.

See Chapter 14, "Mail Services (Reference)," 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, the interactions of `sendmail` with name services, and the features in version 8.13 of `sendmail` that are not fully described in the `sendmail(1M)` man page.

Task Map for Mail Services

The following table refers you to other task maps 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, and a mail gateway. Learn how to use DNS with `sendmail`.	"Setting Up Mail Services (Task Map)" on page 273

Task	Description	For Instructions
Altering the sendmail configuration	Use these procedures to modify your configuration files or service properties.	"Changing the sendmail Configuration (Task Map)" on page 281
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 291
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, and run a subset of the mail queue. Also, learn how to run the old mail queue.	"Administering the Queue Directories (Task Map)" on page 302
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 305
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. Also, learn where to look for other mail diagnostic information.	"Troubleshooting Procedures and Tips for Mail Services (Task Map)" on page 308
Resolving error messages	Use the information in this section to resolve some mail-related error messages.	"Resolving Error Messages" on page 312

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 271 describes the first configuration type, and "Local Mail and a Remote Connection" on page 272 describes the second type.
- As necessary, choose the systems that are to act as mail servers, mail hosts, and mail gateways.

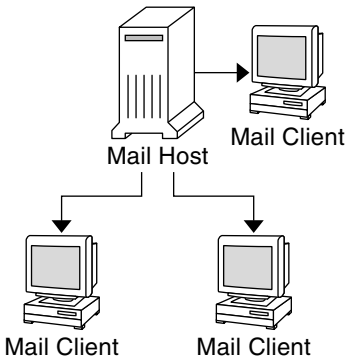
- Make a list of all the mail clients for which you are providing service and include the location of their mailboxes. This list can help you when you are ready to create mail aliases for your users.
- Decide how to update aliases and forward mail messages. You might set up an `aliases` mailbox as a place for users to send requests for mail forwarding. Users could also use this mailbox to send requests 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 mail forwarding. "Administering Mail Alias Files (Task Map)" on page 291 provides a list of tasks that are related to aliasing. "Administering `.forward` Files (Task Map)" on page 305 provides a list of tasks that are related to managing `.forward` files.

After you have completed the planning process, set up the systems on your site to perform the functions that are described in "Setting Up Mail Services (Task Map)" on page 273. For other task information, refer to "Task Map for Mail Services" on page 269.

Local Mail Only

The simplest mail configuration, as shown in Figure 13–1, 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 the clients act as mail servers. Mail addresses are parsed by using the `/etc/mail/aliases` files.

FIGURE 13–1 Local Mail Configuration



To set up this kind of mail configuration, you need the following.

- The default `/etc/mail/sendmail.cf` file, which requires no editing, on each mail client system.

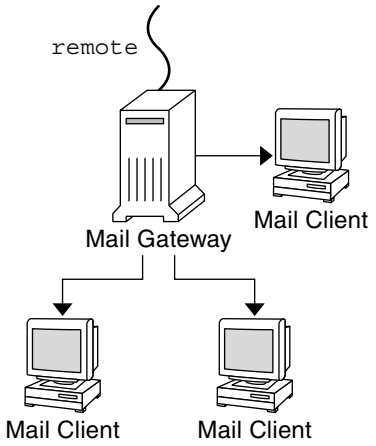
- A server that is designated as the mail host. If you are running NIS or NIS+, you can make this designation by adding `mailhost.`*domain-name* to the `/etc/hosts` file on the mail host. If you are running another name service, such as DNS or LDAP, you must provide additional information in the `/etc/hosts` file. See "How to Set Up a Mail Host" on page 277.
- If you are using a name service other than NIS or NIS+, you need matching `/etc/mail/aliases` files on any system that has a local mailbox.
- Enough space in `/var/mail` on each mail client system to hold the mailboxes.

For task information about setting up your mail service, refer to "Setting Up Mail Services" on page 273. 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 273.

Local Mail and a Remote Connection

The most common mail configuration in a small network is shown in Figure 13–2. One system includes the mail server, the mail host, and the mail gateway that provides the remote connection. Mail is distributed by using the `/etc/mail/aliases` files on the mail gateway. No name service is required.

FIGURE 13–2 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 default `/etc/mail/sendmail.cf` file on each mail client system. This file does not require any editing.

- A server that is designated as the mail host. If you are running NIS or NIS+, you can make this designation by adding `mailhost.`*domain-name* to the `/etc/hosts` file on the mail host. If you are running another name service, such as DNS or LDAP, you must provide additional information in the `/etc/hosts` file. See "How to Set Up a Mail Host" on page 277.
- If you are using a name service other than NIS or NIS+, you need matching `/etc/mail/aliases` files on any system that has a local mailbox.
- Enough space in `/var/mail` on the mail server to hold the client mailboxes.

For task information about setting up your mail service, refer to "Setting Up Mail Services" on page 273. 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 273.

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 274
Setting up a mail client	Steps to enable a user to receive mail	"How to Set Up a Mail Client" on page 276
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 277
Setting up a mail gateway	Steps to manage communication with networks outside your domain	"How to Set Up a Mail Gateway" on page 279
Using DNS with `sendmail`	Steps to enable DNS host lookups	"How to Use DNS With `sendmail`" on page 280

Setting Up Mail Services

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 13–1 in "Local Mail Only" on page 271 for a representation of these configurations. Mail requires two more configurations for communication with networks outside your domain. Refer to Figure 12–1 in "Overview of the Hardware Components" on page 266 or Figure 13–2 in "Local Mail and a Remote Connection" on page 272 for a representation of these configurations. You can combine these configurations on the same system or provide these configurations 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 namespace. Also, for mail to be delivered, the user should have a local home directory for checking the `~/.forward` file. For this reason, home directory servers are often set up as the mail server. "Hardware Components" on page 326 in Chapter 14, "Mail Services (Reference)," provides more information about the mail server.

The mail server can route mail for many mail clients. This type of mail server must have adequate spooling space for client mailboxes.

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.

For clients to access their mailboxes, the `/var/mail` directory should be available for remote mounting. Alternately, a service such as Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) should be available from the server. 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.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `sendmail`.**

   ```
   # svcadm disable -t network/smtp:sendmail
   ```

3. **Check if the `/var/mail` directory is available for remote access.**

   ```
   # share
   ```

   If the `/var/mail` directory is listed, proceed 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 81, to use the `/var/mail` directory to start NFS services.

b. (Optional) If the `/var/mail` directory is not included in the list, add the directory 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.

```
# 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`.

```
# svcadm enable network/smtp:sendmail
```

How to Set Up a Mail Client

A mail client is a user of mail services with a mailbox on a mail server. Additionally, the mail client has a mail alias in the `/etc/mail/aliases` file that points to the location of the mailbox.

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 on the mail client's system or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Stop `sendmail`.

```
# svcadm disable -t network/smtp:sendmail
```

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 that 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 whenever 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 291 for a task map about administering mail alias files. Note that 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`.

```
# svcadm enable network/smtp:sendmail
```

▼ 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 provides your network with a remote connection or connects your network to a parent domain. The following procedure shows you how to set up a mail host.

1 Become superuser on the mail host system or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Stop `sendmail`.

```
# svcadm disable -t network/smtp:sendmail
```

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/sbin/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 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 8.13.1+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.

Create the following entry.

IP-address mailhost `mailhost` *mailhost.domain* `loghost`

5 Restart `sendmail`.

```
# svcadm enable network/smtp:sendmail
```

6 Test your mail configuration.

See "How to Test the Mail Configuration" on page 308 for instructions.

Note – For further information about mail hosts, refer to "Hardware Components" on page 326 in Chapter 14, "Mail Services (Reference)."

▼ 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. Another good candidate is 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 on the mail gateway or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Stop `sendmail`.

```
# svcadm disable -t network/smtp:sendmail
```

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/sbin/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 277.

4 **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.

5 **Restart `sendmail`.**

```
# svcadm enable network/smtp:sendmail
```

6 **Test your mail configuration.**

See "How to Test the Mail Configuration" on page 308 for instructions.

Note – For more information about the mail gateway, refer to "Hardware Components" on page 326 in Chapter 14, "Mail Services (Reference)."

▼ 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 names in the DNS database. For more information

about `sendmail` and DNS, see "Interactions of `sendmail` With Name Services" on page 346 in Chapter 14, "Mail Services (Reference)," or see the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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, in order for the DNS host aliases to be used.

```
# grep hosts /etc/nsswitch.conf
#hosts:     nisplus [NOTFOUND=return] files
hosts:      dns nisplus [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.

Changing the sendmail Configuration (Task Map)

Task	Description	For Instructions
Building a `sendmail` configuration file	Use this procedure to modify your `sendmail.cf` file. An example of how to enable domain masquerading is included.	"How to Build a New `sendmail.cf` File" on page 282
Setting up a virtual host	Steps to configure sendmail so that mail is accepted for more than one domain.	"Setting Up a Virtual Host" on page 283
Setting up automatic rebuilding of the sendmail configuration file	Use this procedure to modify the sendmail service so that the `sendmail.cf` and `submit.mc` configuration files are automatically rebuilt after an upgrade.	"How to Automatically Rebuild a Configuration File" on page 284
Running `sendmail` in the open mode.	Use this procedure to modify the sendmail service properties to enable the open mode.	"How to Use `sendmail` in the Open Mode" on page 284
Setting SMTP to use Transport Layer Security (TLS)	Use this procedure to enable SMTP to have secure connections with TLS.	"How to Set SMTP to Use TLS" on page 285
Managing mail delivery with an alternate configuration	Use this procedure to prevent mail delivery problems that can occur if the master daemon is disabled.	"How to Manage Mail Delivery by Using an Alternate Configuration of `sendmail.cf`" on page 290

Changing the sendmail Configuration

"How to Build a New `sendmail.cf` File" on page 282 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, refer to the following.

- `/etc/mail/cf/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 318 and "`sendmail` Configuration File" on page 338, in Chapter 14, "Mail Services (Reference)," provide some guidance.
- "Additional and Revised `m4` Configuration Macros From Version 8.12 of `sendmail`" on page 365 is also helpful.

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 `/etc/mail/cf/cf/sendmail.mc`.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Stop `sendmail`.

```
# svcadm disable -t network/smtp:sendmail
```

3 Make a copy of the configuration files that you are changing.

```
# cd /etc/mail/cf/cf
# cp sendmail.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 sent mail to be labeled as originating 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
```

While this command displays messages, it sends a message to `testaddr`. 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 308.

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.**

```
# svcadm enable network/smtp:sendmail
```

Setting 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/tips/virtualHosting`. This site provides complete instructions about 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 system, perform the following steps.

```
# cd /etc/mail/cf/cf
# /usr/ccs/bin/make mailserver.cf
# cp mailserver.cf /etc/mail/sendmail.cf
```

mailserver Use the name of the `.cf` file.

"Changing the sendmail Configuration" on page 282 outlines the 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.

How to Automatically Rebuild a Configuration File

If you have built your own copy of `sendmail.cf` or `submit.cf`, the configuration file is not replaced during the upgrade process. The following procedure shows how to configure the sendmail service properties so that the `sendmail.cf` file is automatically rebuilt for you. For instructions on how to automatically build the `submit.cf` configuration file, see Example 13–1. You may combine these procedures if you need to build both files.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Set the sendmail properties.**

```
# svccfg -s sendmail
svc:/network/smtp:sendmail> setprop config/path_to_sendmail_mc=/etc/mail/cf/cf/myhost.mc
svc:/network/smtp:sendmail> quit
```

3. **Refresh and restart the sendmail service.**

   The first command pushes the changes into the running snapshot. The second command restarts the sendmail service using the new options.

```
# svcadm refresh svc:/network/smtp:sendmail
# svcadm restart svc:/network/smtp:sendmail
```

Example 13–1 Establishing Automatic Rebuilding of `submit.cf`

This procedure configures the sendmail service, such that the `submit.mc` configuration file is rebuilt automatically.

```
# svccfg -s sendmail-client:default
svc:/network/smtp:sendmail> setprop config/path_to_submit_mc=/etc/mail/cf/cf/submit-myhost.mc
svc:/network/smtp:sendmail> exit
# svcadm refresh svc:/network/sendmail-client
# svcadm restart svc:/network/sendmail-client
```

How to Use `sendmail` in the Open Mode

In the Solaris 10 release, the sendmail service was changed so that it would run in local–only mode by default. The local-only mode means that only mail from the local host is accepted. Messages from any other systems are rejected. Earlier releases were configured to accept incoming mail from all remote systems, which is known as the open mode. To use the open mode, use the following procedure.

Caution – Running sendmail in the local–only mode is much more secure than running in the open mode. Make sure that you are aware of the potential security risks if you follow this procedure.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Set the sendmail properties.**

```
# svccfg -s sendmail
svc:/network/smtp:sendmail> setprop config/local_only = false
svc:/network/smtp:sendmail> quit
```

3. **Refresh and restart the sendmail service.**

```
# svcadm refresh svc:/network/smtp:sendmail
# svcadm restart svc:/network/smtp:sendmail
```

How to Set SMTP to Use TLS

Starting in the Solaris 10 1/06 release, SMTP can use Transport Layer Security (TLS) in version 8.13 of `sendmail`. This service to SMTP servers and clients provides private, authenticated communications over the Internet, as well as protection from eavesdroppers and attackers. Note that this service is not enabled by default.

The following procedure uses sample data to show you how to set up the certificates that enable `sendmail` to use TLS. For more information, see "Support for Running SMTP With TLS in Version 8.13 of `sendmail`" on page 351.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop `sendmail`.**

```
# svcadm disable -t network/smtp:sendmail
```

3. **Set up the certificates that enable `sendmail` to use TLS.**

   a. **Complete the following:**

```
# cd /etc/mail
# mkdir -p certs/CA
# cd certs/CA
# mkdir certs crl newcerts private
```

```
# echo "01" > serial
# cp /dev/null index.txt
# cp /etc/sfw/openssl/openssl.cnf .
```

b. Use your preferred text editor to change the `dir` value in the `openssl.cnf` file from `/etc/sfw/openssl` to `/etc/mail/certs/CA`.

c. Use the `openssl` command-line tool to implement TLS.

Note that the following command line generates interactive text.

```
# openssl req -new -x509 -keyout private/cakey.pem -out cacert.pem -days 365 \
-config openssl.cnf
Generating a 1024 bit RSA private key
.....................................++++++
.....................................++++++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:California
Locality Name (eg, city) []:Menlo Park
Organization Name (eg, company) [Unconfigured OpenSSL Installation]:Sun Microsystems
Organizational Unit Name (eg, section) []:Solaris
Common Name (eg, YOUR name) []:somehost.somedomain.example.com
Email Address []:someuser@example.com
```

`req`	This command creates and processes certificate requests.
`-new`	This `req` option generates a new certificate request.
`-x509`	This `req` option creates a self-signed certificate.
`-keyout private/cakey.pem`	This `req` option enables you to assign `private/cakey.pem` as the file name for your newly created private key.
`-out cacert.pem`	This `req` option enables you to assign `cacert.pem` as your output file.
`-days 365`	This `req` option enables you to certify the certificate for `365` days. The default value is `30`.
`-config openssl.cnf`	This `req` option enables you to specify `openssl.cnf` as the configuration file.

Note that this command requires that you provide the following:

- Country Name, such as US.
- State or Province Name, such as California.
- Locality Name, such as Menlo Park.
- Organization Name, such as Sun Microsystems.
- Organizational Unit Name, such as Solaris.
- Common Name, which is the machine's fully qualified host name. For more information, see the check-hostname(1M) man page.
- Email Address, such as someuser@example.com.

4 (Optional) If you need a new secure connection, make a new certificate and sign the new certificate with the certificate authority.

a. Make a new certificate.

```
# cd /etc/mail/certs/CA
# openssl req -nodes -new -x509 -keyout newreq.pem -out newreq.pem -days 365 \
-config openssl.cnf
Generating a 1024 bit RSA private key
..............++++++
..............++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:California
Locality Name (eg, city) []:Menlo Park
Organization Name (eg, company) [Unconfigured OpenSSL Installation]:Sun Microsystems
Organizational Unit Name (eg, section) []:Solaris
Common Name (eg, YOUR name) []:somehost.somedomain.example.com
Email Address []:someuser@example.com
```

This command requires that you provide the same information that you provided in step 3c.

Note that in this example, the certificate and private key are in the file newreq.pem.

b. Sign the new certificate with the certificate authority.

```
# cd /etc/mail/certs/CA
# openssl x509 -x509toreq -in newreq.pem -signkey newreq.pem -out tmp.pem
Getting request Private Key
Generating certificate request
# openssl ca -config openssl.cnf -policy policy_anything -out newcert.pem -infiles tmp.pem
Using configuration from openssl.cnf
Enter pass phrase for /etc/mail/certs/CA/private/cakey.pem:
```

```
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jun 23 18:44:38 2005 GMT
            Not After : Jun 23 18:44:38 2006 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = California
            localityName              = Menlo Park
            organizationName          = Sun Microsystems
            organizationalUnitName    = Solaris
            commonName                = somehost.somedomain.example.com
            emailAddress              = someuser@example.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                93:D4:1F:C3:36:50:C5:97:D7:5E:01:E4:E3:4B:5D:0B:1F:96:9C:E2
            X509v3 Authority Key Identifier:
                keyid:99:47:F7:17:CF:52:2A:74:A2:C0:13:38:20:6B:F1:B3:89:84:CC:68
                DirName:/C=US/ST=California/L=Menlo Park/O=Sun Microsystems/OU=Solaris/\
                CN=someuser@example.com/emailAddress=someuser@example.com
                serial:00

Certificate is to be certified until Jun 23 18:44:38 2006 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
# rm -f tmp.pem
```

In this example the file `newreq.pem` contains the unsigned certificate and private key. The file `newcert.pem` contains the signed certificate.

`x509` utility	Displays certificate information, converts certificates to various forms, and signs certificate requests
`ca` application	Used to sign certificate requests in a variety of forms and to generate CRLs (certificate revocation lists)

5 Enable `sendmail` to use the certificates by adding the following lines to your `.mc` file.

```
define('confCACERT_PATH', '/etc/mail/certs')dnl
define('confCACERT', '/etc/mail/certs/CAcert.pem')dnl
define('confSERVER_CERT', '/etc/mail/certs/MYcert.pem')dnl
define('confSERVER_KEY', '/etc/mail/certs/MYkey.pem')dnl
define('confCLIENT_CERT', '/etc/mail/certs/MYcert.pem')dnl
define('confCLIENT_KEY', '/etc/mail/certs/MYkey.pem')dnl
```

For more information, see "Configuration File Options for Running SMTP With TLS" on page 352.

6. **Rebuild and install your `sendmail.cf` file in your `/etc/mail` directory.**

   For detailed instructions, see "Changing the sendmail Configuration" on page 282.

7. **Create symbolic links from the files you created with `openssl` to the files you defined in your `.mc` file.**

   ```
   # cd /etc/mail/certs
   # ln -s CA/cacert.pem CAcert.pem
   # ln -s CA/newcert.pem MYcert.pem
   # ln -s CA/newreq.pem MYkey.pem
   ```

8. **For added security, deny read permission to group and others for `MYkey.pem`.**

   ```
   # chmod go-r MYkey.pem
   ```

9. **Use a symbolic link to install CA certs in the directory assigned to `confCACERT_PATH`.**

   ```
   # C=CAcert.pem
   # ln -s $C 'openssl x509 -noout -hash < $C'.0
   ```

10. **For secure mail with other hosts, install their host certificates.**

    a. **Copy the file defined by the other host's `confCACERT` option to `/etc/mail/certs/`*host.domain*`.cert.pem`.**

       Replace *host.domain* with the other host's fully qualified host name.

    b. **Use a symbolic link to install CA certs in the directory assigned to `confCACERT_PATH`.**

       ```
       # C=host.domain.cert.pem
       # ln -s $C 'openssl x509 -noout -hash < $C'.0
       ```

       Replace *host.domain* with the other host's fully qualified host name.

11. **Restart `sendmail`.**

    ```
    # svcadm enable network/smtp:sendmail
    ```

Example 13–2 `Received:` Mail Header

The following is an example of a `Received:` header for secure mail with TLS.

```
Received: from his.example.com ([IPv6:2001:db8:3c4d:15::1a2f:1a2b])
        by her.example.com (8.13.4+Sun/8.13.4) with ESMTP id j2TNUB8i242496
        (version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=OK)
        for <janepc@her.example.com>; Tue, 29 Mar 2005 15:30:11 -0800 (PST)
Received: from her.example.com (her.city.example.com [192.168.0.0])
        by his.example.com (8.13.4+Sun/8.13.4) with ESMTP id j2TNU7cl571102
        version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=OK)
        for <janepc@her.example.com>; Tue, 29 Mar 2005 15:30:07 -0800 (PST)
```

Note that the value for `verify` is `OK`, which means that the authentication was successful. For more information, see "Macros for Running SMTP With TLS" on page 354.

See Also The following OpenSSL man pages:

- openssl(1) (`http://www.openssl.org/docs/apps/openssl.html`).
- req(1) (`http://www.openssl.org/docs/apps/req.html`).
- x509(1) (`http://www.openssl.org/docs/apps/x509.html`).
- ca(1) (`http://www.openssl.org/docs/apps/ca.html`).

How to Manage Mail Delivery by Using an Alternate Configuration of `sendmail.cf`

To facilitate the transport of inbound mail and outbound mail, the new default configuration of `sendmail` uses a daemon and a client queue runner. 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 "`submit.cf` Configuration File From Version 8.12 of `sendmail`" on page 360.

This procedure ensures that your daemon runs only to accept connections from the local host.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Stop `sendmail` client service.

```
# svcadm disable -t sendmail-client
```

3 Make a copy of the configuration file that you are changing.

```
# cd /etc/mail/cf/cf
# cp submit.mc submit-myhost.mc
```

myhost Select a new name for your `.mc` file.

4 Edit the new configuration file (for example, `submit-`*myhost*`.mc`)

Change the listening host IP address to the `msp` definition.

```
# grep msp submit-myhost.mc
FEATURE('msp', '[#.#.#.#]')dnl
```

5 Build the configuration file by using `m4`.

```
# /usr/ccs/bin/make submit-myhost.cf
```

6 **Install the new configuration file after making a copy of the original.**

```
# cp /etc/mail/submit.cf /etc/mail/submit.cf.save
# cp submit-myhost.cf /etc/mail/submit.cf
```

7 **Restart the `sendmail` client service.**

```
# svcadm enable sendmail-client
```

Administering Mail Alias Files (Task Map)

The following table describes the procedures for administering mail alias files. For more information about this topic, refer to "Mail Alias Files" on page 339 in Chapter 14, "Mail Services (Reference)."

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. Initiate an NIS+ `mail_aliases` table.	"How to Initiate an NIS+ `mail_aliases` Table" on page 292
	List the contents of the NIS+ `mail_aliases` table. This procedure includes examples of how to list individual entries and how to list partial matches.	"How to List the Contents of the NIS+ `mail_aliases` Table" on page 293
	Add aliases to the NIS+ `mail_aliases` table from the command line.	"How to Add Aliases to the NIS+ `mail_aliases` Table From the Command Line" on page 294
	Add entries by editing an NIS+ `mail_aliases` table.	"How to Add Entries by Editing an NIS+ `mail_aliases` Table" on page 295
	Edit entries in an NIS+ `mail_aliases` table. This procedure includes an example of how to delete an entry.	"How to Edit Entries in an NIS+ `mail_aliases` Table" on page 295
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 296
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 297

Task	Description	For Instructions
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 299
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 300

Administering Mail Alias Files

Mail aliases must be unique within the domain. This section provides the procedures for administering mail alias files. Alternately, you can use the Mailing List feature in the Solaris Management Console 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 such as 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 346 and "Mail Alias Files" on page 339 in Chapter 14, "Mail Services (Reference)."

Choose from the following procedures:

- "How to Initiate an NIS+ `mail_aliases` Table" on page 292
- "How to List the Contents of the NIS+ `mail_aliases` Table" on page 293
- "How to Add Aliases to the NIS+ `mail_aliases` Table From the Command Line" on page 294
- "How to Add Entries by Editing an NIS+ `mail_aliases` Table" on page 295
- "How to Edit Entries in an NIS+ `mail_aliases` Table" on page 295
- "How to Set Up an NIS `mail.aliases` Map" on page 296
- "How to Set Up a Local Mail Alias File" on page 297
- "How to Create a Keyed Map File" on page 299

▼ How to Initiate an NIS+ `mail_aliases` Table

You can use the `aliasadm` command to manage entries in an NIS+ table. To create a table, follow these instructions. For more information, refer to the `aliasadm(1M)` man page.

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.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Initiate an NIS+ table.**

   ```
   # aliasadm -I
   ```

3. **Add entries to the table.**

   - To add two or three aliases, refer to "How to Add Aliases to the NIS+ `mail_aliases` Table From the Command Line" on page 294.
   - To add more than two or three aliases, refer to "How to Add Entries by Editing an NIS+ `mail_aliases` Table" on page 295.

How to List the 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.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **List all of the entries in alphabetical order by alias.**

   ```
   # aliasadm -l
   ```

   For more information, refer to the `aliasadm`(1M) man page.

Example 13–3 Listing an Individual Entry From the NIS+ `mail_aliases` Table

Alternately, you can use the `aliasadm` command to list individual entries. After you complete the first step in this procedure, type the following:

```
# aliasadm -m ignatz
ignatz: ignatz@saturn # Alias for Iggy Ignatz
```

The command matches only the complete alias name, not partial strings. You cannot use metacharacters, such as `*` and `?`, with `aliasadm -m`.

Example 13–4 Listing Partial Matches From the NIS+ `mail_aliases` Table

Also, you can use the `aliasadm` command to list partial matches. After you complete the first step in this procedure, type the following:

```
# aliasadm -l | grep partial-string
```

Replace *partial-string* with the desired string for your search.

How to Add Aliases to the NIS+ `mail_aliases` Table From the Command Line

To add two or three aliases to the table, follow the following instructions. If you are adding more than two or three aliases, see "How to Add Entries by Editing an NIS+ `mail_aliases` Table" on page 295.

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.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 "How to Initiate an NIS+ `mail_aliases` Table" on page 292.

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 that you created and ensure that the entry is correct.**

   ```
   # aliasadm -m alias
   ```

   alias The entry that you created

   For more information, refer to the `aliasadm`(1M) man page.

How to Add Entries by Editing an NIS+ `mail_aliases` Table

You can use the `aliasadm` command to manage entries in an NIS+ 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.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

3. **Display and edit the aliases table.**

   ```
   # aliasadm -e
   ```

   This command displays the table and enables you to edit the table. The editor that 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 339 and the `aliasadm`(1M) man page.

How to Edit 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.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Display the alias entry.**

   ```
   # aliasadm -m alias
   ```

   Replace *alias* with 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 339.

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 13–5 Deleting Entries From an NIS+ `mail_aliases` Table

To delete entries from the table, use the following syntax after you complete the first step in this procedure:

```
# aliasadm -d alias
```

Replace *alias* with the alias name for the entry that you are deleting.

▼ 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.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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. The changes 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 an `root` on the mail server or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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
```

The configuration of the `AliasFile` option in `/etc/mail/sendmail.cf` determines whether 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, 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` and the `/etc/mail/aliases.db` files to each of the other systems.

You can copy these 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, 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.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 300
- "How to Create a Separate Mailbox for `postmaster`" on page 301
- "How to Add the `postmaster` Mailbox to the Aliases in the `/etc/mail/aliases` File" on page 301

▼ 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.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 301.

How to Create a Separate Mailbox for `postmaster`

If you are creating a separate mailbox for `postmaster`, follow these instructions.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Create a user account for the person who is designated as `postmaster`. Put an asterisk (`*`) in the password field.**

   For details about adding a user account, refer to "Setting Up User Accounts (Task Map)" in *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 superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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` is an example. Also, 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 303
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 303
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. Also, use this procedure to force a particular message out of the queue.	"How to Run a Subset of the Mail Queue, `/var/spool/mqueue`" on page 303
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 304
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 305

Administering the Queue Directories

This section describes some helpful tasks for queue administration. For information about the client-only queue, refer to "`submit.cf` Configuration File From Version 8.12 of `sendmail`" on page 360. For other related information, you can refer to "Additional Queue Features From Version 8.12 of `sendmail`" on page 371.

Refer to the following:

- "How to Display the Contents of the Mail Queue, `/var/spool/mqueue`" on page 303
- "How to Force Mail Queue Processing in the Mail Queue, `/var/spool/mqueue`" on page 303
- "How to Run a Subset of the Mail Queue, `/var/spool/mqueue`" on page 303
- "How to Move the Mail Queue, `/var/spool/mqueue`" on page 304
- "How to Run the Old Mail Queue, `/var/spool/omqueue`" on page 305

How to Display the Contents of the Mail Queue, /var/spool/mqueue

- **Show how many messages are in the queue and how fast they are being cleared from the queue.**

  Type the following:

  ```
  # /usr/bin/mailq | more
  ```

  This command provides the following information.

  - The queue IDs
  - The size of the message
  - The date that the message entered the queue
  - The message status
  - The sender and the recipients

  Additionally, 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 superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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. Also, use this procedure to force a particular message from the queue.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 of *user@host.domain*, such as a host name.

   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.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Kill the `sendmail` daemon.**

   ```
   # svcadm disable network/smtp:sendmail
   ```

   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 that is 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`.**

   ```
   # svcadm enable network/smtp:sendmail
   ```

▼ How to Run the Old Mail Queue, /var/spool/omqueue

To run an old mail queue, follow these instructions.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Run the old mail queue.**

   ```
   # /usr/lib/sendmail -oQ/var/spool/omqueue -q
   ```

   The -oQ flag specifies an alternate queue directory. 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 342 in Chapter 14, "Mail Services (Reference)."

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 306
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 306
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 307

Administering .forward Files

This section contains several procedures that are related to .forward file administration. Because these files can be edited by users, the files can cause problems. For more information, refer to ". .forward Files" on page 342 in Chapter 14, "Mail Services (Reference)."

Refer to the following:

- "How to Disable .forward Files" on page 306
- "How to Change the .forward–File Search Path" on page 306
- "How to Create and Populate /etc/shells" on page 307

How to Disable .forward Files

This procedure, which prevents automated forwarding, disables the .forward file for a particular host.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Make a copy of /etc/mail/cf/domain/solaris-generic.m4 or your site-specific domain m4 file.**

```
# cd /etc/mail/cf/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','')dnl
```

If a value for confFORWARD_PATH already exists in the m4 file, replace the value 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 282.

Note – When you edit the .mc file, remember to change DOMAIN('solaris-generic') to DOMAIN('*mydomain*').

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 superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Make a copy of /etc/mail/cf/domain/solaris-generic.m4 or your site-specific domain m4 file.**

```
# cd /etc/mail/cf/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 the value 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 282.

Note – When you edit the .mc file, remember to change DOMAIN('solaris-generic') to DOMAIN('*mydomain*').

▼ How to Create and Populate /etc/shells

This file is not included in the standard release. You must add the file 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, the following procedure, which employs a script that can be downloaded, is easier to use.

1 **Download the script.**

http://www.sendmail.org/vendor/sun/gen-etc-shells.html

2 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 308
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 309
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 310
Verifying connections to other systems	Tips for verifying connections to other systems	"How to Verify Connections to Other Systems" on page 311
Logging messages by using the `syslogd` program	Tips for gathering error message information	"Logging Error Messages" on page 311
Checking other sources for diagnostic information	Tips for getting diagnostic information from other sources	"Other Sources for Mail Diagnostic Information" on page 312

Troubleshooting Procedures and Tips for Mail Services

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 that you make to your configuration file, follow these instructions.

1 **Restart `sendmail` on any system that has a revised configuration file.**

```
# svcadm refresh network/smtp: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 connected to a network, 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 from the mail host to another domain 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. 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 the program passes the message to UUCP for delivery.

8. **From different systems, send a message to `postmaster` and ensure that the message is delivered to your postmaster's mailbox.**

How to Check Mail Aliases

The following example shows you how to verify an alias.

```
% mconnect
connecting to host localhost (127.0.0.1), port 25
connection open
220 your.domain.com ESMTP Sendmail 8.13.6+Sun/8.13.6; Tue, 12 Sep 2004 13:34:13 -0800 (PST)
expn sandy
250 2.1.5 <sandy@phoenix.example.com>
quit
221 2.0.0 your.domain.com closing connection
%
```

In this example, the `mconnect` program opened a connection to a mail server on a local host and enabled you to test that connection. The program runs interactively, so you can issue various diagnostic commands. For a complete description, see the `mconnect(1)` man page. The entry,

`expn sandy`, provided the expanded address, `sandy@phoenix.example.com`. Thus, you have verified that mail can be delivered when using the alias `sandy`.

Remember to avoid loops and inconsistent databases when both local and domain-wide aliases are used. Be especially careful to avoid the creation of alias loops when you move a user from one system to another system.

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-sraddress*

   mail-address Use the mail address that you are testing.

3. **End the session.**

   Press Control-d.

Example 13–6 Address Test Mode Output

The following is an example of the output from the address test mode.

```
% /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 8.13.1+Sun/8.13.1; Sat, 4 Sep 2004 3:52:56 -0700
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?

Logging Error Messages

Your mail service logs most error messages by using the `syslogd` program. By default, the `syslogd` program sends these messages to a system that is 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 that are higher than 10 are usually used for debugging. See "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 that the message took as the message 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. The script 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 messages and outgoing messages.

Resolving Error Messages

This section describes how you can resolve some `sendmail`–related error messages. 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
- **Solution:** What you can do to fix the problem or to continue with your work

`451 timeout waiting for input during` *source*

Cause: When `sendmail` reads from any source that might time out, such as an SMTP connection, the program 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.

Solution: 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.

`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.

Solution: 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.

Solution: 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.

Solution: 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.

Description: 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.

For additional information, refer to `http://www.sendmail.org/faq/section4#4.5`.

`config error: mail loops back to myself.`

Cause: This error message occurs if you set up an MX record and make host *bar* the mail exchanger for domain *foo*. However, you fail to configure host *bar* to know that it is the mail exchanger for domain *foo*.

Also, another possibility is that both the sending system and the receiving system are identifying as the same domain.

Solution: For instructions, refer to `http://www.sendmail.org/faq/section4#4.5`.

`host name configuration error`

Description: This is an old `sendmail` message, which replaced `I refuse to talk to myself` and is now replaced by the `Local configuration error` message.

Solution: Follow the instructions that were provided for resolving this error message, `554` *hostname*`... Local configuration error`.

`user unknown`

Cause: When you try to send mail to a user, the error `Username... user unknown` is displayed. The user is on the same system.

Solution: 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.

For additional information, refer to `http://www.sendmail.org/faq/section4#4.17`.

◆ ◆ ◆ CHAPTER 14

Mail Services (Reference)

The `sendmail` program is a mail transport agent. The program uses a configuration file to provide aliasing and forwarding, automatic routing to network gateways, and flexible configuration. The Solaris OS supplies standard configuration files that most sites can use. Chapter 12, "Mail Services (Overview)," provides an introduction to the components of mail services and a description of a typical mail service configuration. Chapter 13, "Mail Services (Tasks)," explains how to set up and administer an electronic mail system. This chapter provides information about the following topics.

For details that are not covered in these chapters, see the following man pages:

- `sendmail`(1M)
- `mail.local`(1M)
- `mailstats`(1)
- `makemap`(1M)
- `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.

- "Alternative sendmail Commands" on page 317
- "Versions of the Configuration File" on page 318

Flags Used and Not Used to Compile sendmail

Starting in the Solaris 10 release, the following flags are used to compile sendmail. If your configuration requires other flags, you need to download the source and recompile the binary. You can find information about this process at http://www.sendmail.org.

TABLE 14–1 General sendmail Flags

Flag	Description
SOLARIS=21000	Support for the Solaris 10 release.
MILTER	Support for the Mail Filter API. In version 8.13 of sendmail, this flag is enabled by default. See "MILTER, Mail Filter API for sendmail" on page 317.
NETINET6	Support for IPv6. This flag has been moved from conf.h to Makefile.

TABLE 14–2 Maps and Database Types

Flag	Description
NDBM	Support for ndbm databases
NEWDB	Support for Berkeley 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 14–3 OS Flags

Flag	Description
SUN_EXTENSIONS	Support for extensions that are included in sun_compat.o.
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.

TABLE 14–3 OS Flags *(Continued)*

Flag	Description
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 10 release.

TABLE 14–4 Generic Flags Not Used in This 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 that are 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.

MILTER, Mail Filter API for sendmail

MILTER, sendmail's Mail Filter API, enables third-party programs to access mail messages as they are being processed to filter meta-information and content. You do not need to build the filter and configure sendmail to use it. This API is enabled by default in version 8.13 of sendmail.

For more details, see the following:

- http://www.sendmail.org
- https://www.milter.org/

Alternative sendmail Commands

The Solaris release does not include all of the command synonyms that are provided in the generic release from sendmail.org. This table includes a complete list of the command aliases. The table also lists whether the commands are included in the Solaris release and how to generate the same behavior by using sendmail.

TABLE 14–5 Alternate sendmail Commands

Alternate Name	In This 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

Starting in the Solaris 10 release, 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 10. You can also define the vendor. Either Berkeley or Sun is a valid vendor option. 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 14–6 Version Values for the Configuration File

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 version 8.12 and version 8.13 of sendmail. Version 8.12 is the default for the Solaris 9 release. Starting in the Solaris 10 release, version 8.13 is the default.

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 "Changing the sendmail Configuration" on page 282 in Chapter 13, "Mail Services (Tasks)."

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 system supplies the following mail user agents.

- `/usr/bin/mail`
- `/usr/bin/mailx`
- `/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 system 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 system.

- The UUCP local delivery agent, which uses `uux` to deliver mail
- The local delivery agent, which is `mail.local` in the standard Solaris release

"Changes From Version 8.12 of `sendmail`" on page 359 provides information on these related topics.

- "Additional Delivery Agent Flags From Version 8.12 of `sendmail`" on page 369
- "Additional Equates for Delivery Agents From Version 8.12 of `sendmail`" on page 370

Mailers and `sendmail`

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 "Changing the sendmail Configuration" on page 282 in Chapter 13, "Mail Services (Tasks)." This section provides a brief description of two types of mailers.

- "Simple Mail Transfer Protocol (SMTP) Mailers" on page 320
- "UNIX-to-UNIX Copy Program (UUCP) Mailers" on page 320

For additional information about mailers, see `http://www.sendmail.org/m4/readme.html` or `/etc/mail/cf/README`.

Simple Mail Transfer Protocol (SMTP) Mailers

SMTP is the standard mail protocol that is used on the Internet. This protocol defines these mailers.

- `smtp` provides regular 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 From Version 8.12 of `sendmail`" on page 369 and "Additional Delivery Agent Flags From Version 8.12 of `sendmail`" on page 369.

UNIX-to-UNIX Copy Program (UUCP) Mailers

If possible, avoid using UUCP. For an explanation, refer to `http://www.sendmail.org/m4/uucp_mailers.html` or do a search in `/etc/mail/cf/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. Complexity is introduced if you are administering a mail system that has more than one system with mailboxes or that has one or more domains. Additional complexity can be generated if you have a UUCP (or other) mail connection to servers outside your network. The information in the following sections can help you understand the parts and complexities of a mail address.

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 14–7 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 errors.

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 346.

Typical Format for Mail Addresses

Typically, a mail address has the following format. For further details, refer to "Route–Independent Mail Addresses" on page 323.

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 and `sendmail`" on page 320.

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, so the more local the subdomain, the closer the subdomain 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. Actually, 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 first read by searching for the `@` sign. The domain hierarchy is then read 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, such as 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, 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 14–8 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. Alternately, user names can be identified 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.
`owner-owner`	This alias is used when no `owner-`*aliasname* alias exists for errors to be returned to. This address should redirect mail to the person who handles mail errors. This address also 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, this convention 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 the mail 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 291 in Chapter 13, "Mail Services (Tasks)." Also, you can refer to "Mail Alias Files" on page 339 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. So, 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 who is 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. Otherwise, you could update all the local `/etc/mail/aliases` files to keep the aliases synchronized.
- Users can also create and use aliases. Users can create aliases either in their local `~/.mailrc` file, which only the user 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 326
- "Mail Server" on page 327
- "Mail Client" on page 327

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 327. 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 cannot be delivered. 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. Alternately, you can add the word `mailhost` similarly to the hosts file in the name service. For detailed task information, refer to "How to Set Up a Mail Host" on page 277 in Chapter 13, "Mail Services (Tasks)."

A good candidate for a mail host is a system that is configured as a router from your network to the Internet global network. For more information, refer to Chapter 15, "Solaris PPP 4.0 (Overview)," Chapter 24, "UUCP (Overview)," and "Configuring an IPv4 Router" in *System Administration Guide: IP Services*. If no systems on your local network have a modem, designate a system as the mail host.

Some sites use standalone machines that are not networked in a time-sharing configuration. Specifically, 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 single-system network. "Overview of the Hardware Components" on page 266 in Chapter 12, "Mail Services (Overview)," 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 274 in Chapter 13, "Mail Services (Tasks)."

The mail server routes all mail from a client. When a client sends mail, the mail server puts the mail 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. Otherwise, users can use 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.

By establishing a mail server for all mailboxes, you can simplify your process of doing 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 a user of mail services with a mailbox on a mail server. Additionally, the mail client has a mail alias in the `/etc/mail/aliases` file that points to the location of the mailbox. For task information, refer to "How to Set Up a Mail Client" on page 276 in Chapter 13, "Mail Services (Tasks)."

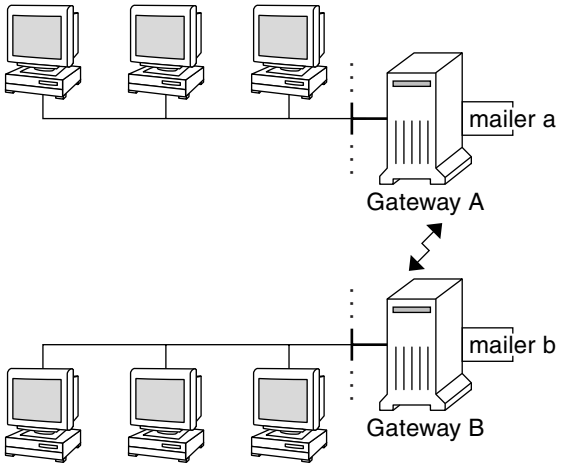
Mail Gateway

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 the mail gateway 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 the gateway to send and receive 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 14–1 Gateway Between Different Communications Protocols



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 other networks, and set up that gateway as the mail gateway. For task information, refer to "How to Set Up a Mail Gateway" on page 279 in Chapter 13, "Mail Services (Tasks)."

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.

- "/etc/default/sendmail File" on page 344

Enhancement for vacation Utility

Starting in the Solaris 10 release, the vacation utility has been enhanced to enable a user to specify which incoming messages receive autogenerated replies. With this enhancement the user can avoid sharing confidential or contact information with unknown people. Messages from spammers or unknown people would not receive a reply.

This enhancement works by matching an incoming sender's email address to a list of domains or email addresses in a .vacation.filter file. This file is created by the user and is in the user's home directory. If a domain or email address match is found, a reply is sent. If no match is found, no reply is sent.

The .vacation.filter might contain entries such as these:

```
company.com
mydomain.com
onefriend@hisisp.com
anotherfriend@herisp.com
```

Note that each line contains one domain or one email address. Each entry must be on a separate line. For a sender's email address to match with an email address entry, the match must be exact, except for case. Whether the letters in the sender's address are lowercase or uppercase is ignored. For a sender's email address to match with a domain entry, the sender's address must contain the listed domain. For example, both somebody@dept.company.com and someone@company.com would be a match for a domain entry of company.com.

For more information, see the vacation(1) man page.

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	File	A program that lists the content of the mail queue.
mailstats	File	A program that is used to read mail statistics that are stored in the /etc/mail/statistics file (if present).

Name	Type	Description
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	Symbolic Link	A symbolic 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 mailx user agent.
aliases	File	Mail-forwarding information.
aliases.db	File	Default binary form of mail-forwarding information that is created by running newaliases.
aliases.dir	File	Binary form of mail-forwarding information that is created by running newaliases. Can still be used, but is no longer used by default starting with the Solaris 9 release.
aliases.pag	File	Binary form of mail-forwarding information that is created by running newaliases. Can still be used, but is no longer used by default starting with the Solaris 9 release.
mailx.rc	File	Default settings for the mailx user agent.
main.cf	Symbolic link	A symbolic link from this sample configuration file for main systems to sendmail.cf is provided for backwards compatibility. This file is not needed in version 8.13 of sendmail.
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 "submit.cf Configuration File From Version 8.12 of sendmail" on page 360.

Name	Type	Description
`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`.
`statistics`	File	`sendmail` statistics file. If this file is present, `sendmail` logs the amount of traffic through each mailer. Previously, this file was called `sendmail.st`.
`subsidiary.cf`	Symbolic link	A symbolic link from this sample configuration file for subsidiary systems to `sendmail.cf` is provided for backwards compatibility. This file is not needed in version 8.13 of `sendmail`.
`trusted-users`	File	File that lists the users (one user 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 `/etc/mail/cf` Directory

Within the `/etc/mail` directory is a subdirectory, `cf`, that contains all of the necessary files to build a `sendmail.cf` file. The content of `cf` is shown in Table 14–9.

Starting in the Solaris 10 release, to support a read-only `/usr` file system, the content of the `/usr/lib/mail` directory has been moved to the `/etc/mail/cf` directory. Note, however, these exceptions. The shell scripts `/usr/lib/mail/sh/check-hostname` and `/usr/lib/mail/sh/check-permissions` are now in the `/usr/sbin` directory. See "Other Files Used for Mail Services" on page 334. For backward compatibility, symbolic links point to each file's new location.

TABLE 14–9 Contents of the `/etc/mail/cf` Directory Used for Mail Services

Name	Type	Description
`README`	File	Describes the configuration files.
`cf/main.cf`	Symbolic Link	As of the Solaris 10 release, this file name is linked to `cf/sendmail.cf`. This file used to be the main configuration file.
`cf/main.mc`	Symbolic Link	As of the Solaris 10 release, this file name is linked to `cf/sendmail.mc`. This file was the file used to create the main configuration file.

TABLE 14–9 Contents of the /etc/mail/cf Directory Used for Mail Services *(Continued)*

Name	Type	Description
cf/Makefile	File	Provides rules for building new configuration files.
cf/submit.cf	File	Is the configuration file for the mail submission program (MSP), which is used to submit messages.
cf/submit.mc	File	Is the file used to build the submit.cf file. The file defines m4 macros for the mail submission program (MSP).
cf/sendmail.cf	File	Is the main configuration file for sendmail.
cf/sendmail.mc	File	Contains the m4 macros that are used to generate the sendmail.cf file.
cf/subsidiary.cf	Symbolic Link	As of the Solaris 10 release, this file name is linked to cf/sendmail.cf. This file used to be the configuration file for hosts that NFS-mount /var/mail from another host.
cf/subsidiary.mc	Symbolic Link	As of the Solaris 10 release, this file name is linked to cf/sendmail.mc. This file used to contain the m4 macros that were used to generate the subsidiary.cf file.
domain	Directory	Provides site-dependent subdomain descriptions.
domain/generic.m4	File	Is the generic domain file from Berkeley Software Distribution.
domain/solaris-antispam.m4	File	Is the domain file with changes that make sendmail function like the previous versions of sendmail. However, 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 the previous versions of sendmail.
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.

TABLE 14–9 Contents of the /etc/mail/cf Directory Used for Mail Services *(Continued)*

Name	Type	Description
mailer	Directory	Contains definitions of mailers, which include local, smtp, and uucp.
main-v7sun.mc	File	Obsolete: as of the Solaris 10 release, this file name is renamed to cf/sendmail.mc.
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.
subsidiary-v7sun.mc	File	Obsolete: as of the Solaris 10 release, this file name is renamed to cf/sendmail.mc.

Contents of the /usr/lib Directory

The following table shows the contents of the /usr/lib directory, which is used for mail services.

TABLE 14–10 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 (sendmail restricted shell) that uses the "|program" syntax of sendmail to restrict programs that sendmail can run to those programs listed in the /var/adm/sm.bin directory. Refer to the smrsh(1M) man page for recommendations about what to include in /var/adm/sm.bin. To enable, include this m4 command, FEATURE('smrsh'), in your mc file.
mail	symbolic link	A symbolic link points to the/etc/mail/cf directory. For more information, refer to "Contents of the /etc/mail/cf Directory" on page 331.

Other Files Used for Mail Services

Several other files and directories are used for mail services, as shown in Table 14–11.

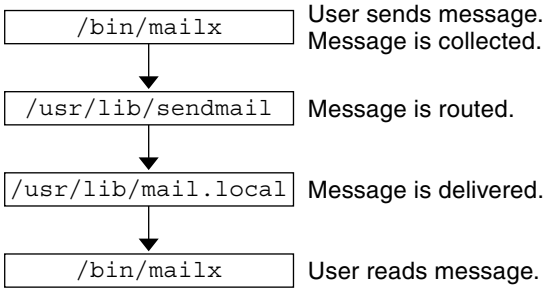
TABLE 14–11 Other Files Used for Mail Services

Name	Type	Description
`/etc/default/sendmail`	File	Lists the environment variables for the startup script for `sendmail`.
`/etc/shells`	File	Lists the valid login shells.
`/etc/mail/cf/sh`	Directory	Contains shell scripts that are used by the `m4` build process and migration aids.
`/usr/sbin/check-permissions`	File	Checks permissions of `:include:` aliases and `.forward` files and their parent directory path for correct permissions.
`/usr/sbin/check-hostname`	File	Verifies that `sendmail` is able to determine the fully qualified host name.
`/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`	Symbolic Link	A symbolic 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.
`/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 14–2.

FIGURE 14–2 Interactions of Mail Programs



The following is a description of the interactions of mail programs.

1. Users send messages by using programs such as `mailx`. See the man page for `mailx(1)` for more information.
2. The message is collected by the program that generated the message, and the message 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 which 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 the mail by using `mail`, `mailx`, 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 queuing, 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 from logical naming. For more information about domains, refer to "Mail Addresses" on page 321.
  - 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 system 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 to a local delivery agent, such as `mail.local` or `procmail`.
- `sendmail` is a mail transfer agent that accepts messages from user agents, such as `mailx` and Mozilla Mail, and routes the messages through the Internet to their destination.
- `sendmail` controls email messages that users send in the following ways.
  - 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 336
- "`sendmail` Features" on page 338
- "`sendmail` Configuration File" on page 338

`sendmail` and Its Rerouting Mechanisms

The `sendmail` program supports three mechanisms for mail rerouting. The mechanism that 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 that 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 of name service aliasing.

- The use of a name service 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 of 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 339 in this chapter. For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 291 in Chapter 13, "Mail Services (Tasks)."

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 342 in this chapter. For a task map, refer to "Administering `.forward` Files (Task Map)" on page 305 in Chapter 13, "Mail Services (Tasks)."

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 about inclusion, refer to "`/etc/mail/aliases` File" on page 340 in this chapter. For a task map, refer to "Administering Mail Alias Files (Task Map)" on page 291 in Chapter 13, "Mail Services (Tasks)."

Note – 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 originate from a number of name service sources, such as 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.

`sendmail` Features

The `sendmail` program provides the following features.

- `sendmail` is reliable. The program is designed to correctly deliver every message. No message should ever become completely lost.
- `sendmail` uses existing software for delivery whenever possible. For example, the user interacts with a mail-generating and a mail-sending program. When 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, `sendmail` can be used as an Internet mail gateway. See "Interactions of Mail Programs" on page 335 for a more detailed description of the process.
- `sendmail` can be configured to handle complex environments, including multiple networks. `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. Additionally, 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. The custom mailer can provide functions such as 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.

`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.

The Solaris operating system provides two default configuration files in the `/etc/mail` directory.

1. `sendmail.cf`, a configuration file that is used to run `sendmail` in daemon mode.

2. `submit.cf`, a configuration file that is used to run `sendmail` in mail-submission program mode, instead of daemon mode. For more information, refer to "`submit.cf` Configuration File From Version 8.12 of `sendmail`" on page 360.

When setting up mail clients, mail servers, mail hosts, or mail gateways, consider the following:

- 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, you need 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 273 or "Changing the sendmail Configuration" on page 282 in Chapter 13, "Mail Services (Tasks)." Note that with `sendmail` version 8.13, you no longer need the `main.cf` file.

The following list describes some configuration parameters that you can change, depending on the requirements of your site.

- Time values, which specify the following information.
  - Read timeouts.
  - Length of time a message remains undelivered in the queue before the message is returned to the sender. Refer to "Additional Queue Features From Version 8.12 of `sendmail`" on page 371. For a task map, refer to "Administering the Queue Directories (Task Map)" on page 302.
- Delivery modes, which specify how quickly mail is delivered.
- Load limits, which increase efficiency during busy periods. These parameters prevent `sendmail` from 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.

- "`.mailrc` Aliases" on page 340
- "`/etc/mail/aliases` File" on page 340
- "NIS `aliases` Map" on page 341
- "NIS+ `mail_aliases` Table" on page 342

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 291 in Chapter 13, "Mail Services (Tasks)."

`.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 that 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, an error occurs. 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 the file 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. This program recompiles the database and makes 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 297 in Chapter 13, "Mail Services (Tasks)." Otherwise, you can use the Mailing List feature in the Solaris Management Console to administer the mail aliases that are stored in the local `/etc` files.

You can create aliases for only local names, such as 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 Solaris Management Console, 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 291 in Chapter 13, "Mail Services (Tasks)."

NIS `aliases` Map

All users in a local domain can use the entries that are in the NIS `aliases` map. The reason is that 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. However, 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 296 in Chapter 13, "Mail Services (Tasks)."

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 14–12 describes the four columns that are in an NIS+ `mail_aliases` table.

TABLE 14–12 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 "Administering Mail Alias Files (Task Map)" on page 291 in Chapter 13, "Mail Services (Tasks)." Alternately, you can use the Solaris Management Console 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.

- "Situations to Avoid" on page 343
- "Controls for `.forward` files" on page 343

- "`.forward.`*hostname* File" on page 343
- "`.forward+`*detail* File" on page 344

For a task map, refer to "Administering `.forward` Files (Task Map)" on page 305 in Chapter 13, "Mail Services (Tasks)."

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` checks for `user` in the NIS or NIS+ aliases and sends the message back to `user@host1`. This routing 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 this file 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 the options 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, rather than the functions 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 checks 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 queue run intervals, which are periodic. 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 that 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 checking for the mail alias in the name service and substituting the new value, if a new value is found. This new alias is then checked again.
2. If the mail is local, deliver the mail 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. However, 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. Furthermore, this section describes the rules for effective use of name services, and the specific interactions of sendmail with name services. For details, refer to the following topics.

If you are looking for related task information, refer to "How to Use DNS With sendmail" on page 280 or "Administering Mail Alias Files (Task Map)" on page 291 in Chapter 13, "Mail Services (Tasks)."

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. Actually, the mail host can be an authorized host.

With 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 the section describes the various requirements. Refer to the following.

- Man pages for 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.

Requirements for Name Services

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), if 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 originates 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 host names and short host names** – Follow the previous instructions about 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 291 in Chapter 13, "Mail Services (Tasks)."

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 understand. This requirement is satisfied when you follow the regular procedure for setting up NIS and DNS.

- **Matching full host names 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 280 and "Administering Mail Alias Files (Task Map)" on page 291 in Chapter 13, "Mail Services (Tasks)."

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. This NIS+ table has 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. An example is `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 host names and short host names** – To satisfy this requirement, you can duplicate the entries in the host table. Otherwise, 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. Otherwise, you can type all host entries in the user name service domains into a master host table at mail domain level. Effectively, you are merging multiple host tables that are logical or physical into one host table. Therefore, 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 291 in Chapter 13, "Mail Services (Tasks)."

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. This NIS+ table has 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. An example is 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 both 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 host names 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. Alternately, 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 291 and "How to Use DNS With sendmail" on page 280 in Chapter 13, "Mail Services (Tasks)."

Changes in Version 8.13 of sendmail

Although this new version of sendmail provides many new features, the FallBackSmartHost option is the most significant addition. Because of this option you no longer need to use main.cf and subsidiary.cf. The main.cf file was used in environments that supported MX records. The subsidiary.cf file was used in environments without a fully operative DNS. In such environments a smart host was used instead of MX records. The FallBackSmartHost option provides unified configuration. It operates like an MX record of last possible preference for all environments. To ensure that mail gets delivered to clients, this option, if enabled, provides a well-connected (or smart) host that serves as a backup (or failover) for MX records that fail.

For more information about version 8.13, see the following sections:

- "Additional Command-Line Options in Version 8.13 of sendmail" on page 356
- "Additional and Revised Configuration File Options in Version 8.13 of sendmail" on page 356
- "Additional and Revised FEATURE() Declarations in Version 8.13 of sendmail" on page 358

Additionally, starting in the Solaris 10 1/06 release, SMTP can run with Transport Layer Security (TLS). See the following description.

Support for Running SMTP With TLS in Version 8.13 of sendmail

Communications between SMTP servers and clients are not usually controlled or trusted on either end. This lack of security might allow a third party to monitor and even alter a communication between a server and a client. Starting in the Solaris 10 1/06 release, SMTP can use Transport Layer Security (TLS) in version 8.13 of sendmail to resolve this problem. This extended service to SMTP servers and clients provides the following:

- Private, authenticated communications over the Internet
- Protection from eavesdroppers and attackers

Note – The implementation of TLS is based on the Secure Sockets Layer (SSL) protocol.

STARTTLS is the SMTP keyword that initiates a secure SMTP connection by using TLS. This secure connection might be between two servers or between a server and a client. A secure connection is defined as follows:

- The source email address and the destination address are encrypted.
- The content of the email message is encrypted.

When the client issues the STARTTLS command, the server responds with one of the following:

- `220 Ready to start TLS`
- `501 Syntax error (no parameters allowed)`
- `454 TLS not available due to temporary reason`

The 220 response requires the client to start the TLS negotiation. The 501 response notes that the client incorrectly issued the STARTTLS command. STARTTLS is issued with no parameters. The 454 response necessitates that the client apply rule set values to determine whether to accept or maintain the connection.

Note that to maintain the Internet's SMTP infrastructure, publicly used servers must not require a TLS negotiation. However, a server that is used privately might require the client to perform a TLS negotiation. In such instances, the server returns this response:

```
530 Must issue a STARTTLS command first
```

The 530 response instructs the client to issue the STARTTLS command to establish a connection.

The server or client can refuse a connection if the level of authentication and privacy is not satisfactory. Alternately, because most SMTP connections are not secure, the server and client might maintain an unsecure connection. Whether to maintain or refuse a connection is determined by the configuration of the server and the client.

Support for running SMTP with TLS is not enabled by default. TLS is enabled when the SMTP client issues the STARTTLS command. Before the SMTP client can issue this command, you must set up the certificates that enable sendmail to use TLS. See "How to Set SMTP to Use TLS" on page 285. Note that this procedure includes defining new configuration file options and rebuilding your sendmail.cf file.

Configuration File Options for Running SMTP With TLS

The following table describes the configuration file options that are used to run SMTP with TLS. If you declare any of these options, use one of the following syntaxes:

- O *OptionName=argument* # for the configuration file
- -O *OptionName=argument* # for the command line
- define(‘*m4Name*',*argument*) # for m4 configuration

TABLE 14–13 Configuration File Options for Running SMTP With TLS

Option	Description
CACertFile	m4 name: confCACERT
	Argument: *filename*
	Default value: undefined
	Identifies the file that contains one CA certificate.
CACertPath	m4 name: confCACERT_PATH
	Argument: *path*
	Default value: undefined
	Identifies the path to the directory that contains certificates of CAs.
ClientCertFile	m4 name: confCLIENT_CERT
	Argument: *filename*
	Default value: undefined
	Identifies the file that contains the certificate of the client. Note that this certificate is used when sendmail acts as a client.

TABLE 14–13 Configuration File Options for Running SMTP With TLS *(Continued)*

Option	Description
ClientKeyFile	m4 name: confCLIENT_KEY Argument: *filename* Default value: undefined Identifies the file that contains the private key that belongs to the client certificate.
CRLFile	m4 name: confCRL Argument: *filename* Default value: undefined Identifies the file that contains the certificate revocation status, which is used for X.509v3 authentication.
DHParameters	m4 name: confDH_PARAMETERS Argument: *filename* Default value: undefined Identifies the file that contains the Diffie-Hellman (DH) parameters.
RandFile	m4 name: confRAND_FILE Argument: file:*filename* or egd:*UNIX socket* Default value: undefined Uses the file: prefix to identify the file that contains random data or uses the egd: prefix to identify the UNIX socket. Note that because the Solaris OS supports the random number generator device, this option does not need to be specified. See the random(7D) man page.
ServerCertFile	m4 name: confSERVER_CERT Argument: *filename* Default value: undefined Identifies the file that contains the server's certificate. This certificate is used when sendmail acts as a server.
Timeout.starttls	m4 name: confTO_STARTTLS Argument: *amount of time* Default value: 1h Sets the amount of time the SMTP client waits for a response to the STARTTLS command.

TABLE 14–13 Configuration File Options for Running SMTP With TLS *(Continued)*

Option	Description
TLSSrvOptions	m4 name: confTLS_SRV_OPTIONS Argument: V Default value: undefined Determines whether the server asks for a certificate from the client. If this option is set to V, no client verification is performed.

For sendmail to support SMTP's use of TLS, the following options must be defined:

- CACertPath
- CACertFile
- ServerCertFile
- ClientKeyFile

Other options are not required.

Macros for Running SMTP With TLS

The following table describes the macros that are used by the STARTTLS command.

TABLE 14–14 Macros for Running SMTP With TLS

Macro	Description
${cert_issuer}	Holds the distinguished name (DN) of the certification authority (CA), which is the certificate issuer.
${cert_subject}	Holds the DN of the certificate that is called the **cert subject**.
${cn_issuer}	Holds the common name (CN) of the CA, which is the **cert issuer**.
${cn_subject}	Holds the CN of the certificate that is called the **cert subject**.
${tls_version}	Holds the version of TLS that is used for the connection.
${cipher}	Holds a set of cryptographic algorithms (known as a **cipher suite**) that is used for the connection.
${cipher_bits}	Holds in bits the key length of the symmetric encryption algorithm that is used for the connection.

TABLE 14–14 Macros for Running SMTP With TLS *(Continued)*

Macro	Description
${verify}	Holds the result of the verification of the certificate that was presented. Possible values are as follows: ■ OK – The verification succeeded. ■ NO – No certificate was presented. ■ NOT – No certificate was requested. ■ FAIL – The certificate that was presented could not be verified. ■ NONE – STARTTLS has not been performed. ■ TEMP – Temporary error occurred. ■ PROTOCOL – SMTP error occurred. ■ SOFTWARE – STARTTLS handshake failed.
${server_name}	Holds the name of the server with the current outgoing SMTP connection.
${server_addr}	Holds the address of the server with the current outgoing SMTP connection.

Rule Sets for Running SMTP With TLS

The following table describes rule sets that determine whether an SMTP connection that uses TLS should be accepted, continued, or refused.

TABLE 14–15 Rule Sets for Running SMTP With TLS

Rule Set	Description
tls_server	Acting as a client, sendmail uses this rule set to determine whether the server is currently supported by TLS.
tls_client	Acting as a server, sendmail uses this rule set to determine whether the client is currently supported by TLS.
tls_rcpt	This rule set requires verification of the recipient's MTA. This recipient restriction makes attacks such as DNS spoofing impossible.
TLS_connection	This rule set checks the requirement that is specified by the RHS of the access map against the actual parameters of the current TLS connection.
try_tls	sendmail uses this rule set to determine the feasibility of using STARTTLS when connecting to another MTA. If the MTA cannot properly implement STARTTLS, then STARTTLS is not used.

For more information, see http://www.sendmail.org/m4/starttls.html.

Security Considerations Related to Running SMTP With TLS

As a standard mail protocol that defines mailers that run over the Internet, SMTP is not an end-to-end mechanism. Because of this protocol limitation, TLS security through SMTP does not include mail user agents. Mail user agents act as an interface between users and a mail transfer agent such as `sendmail`.

Also, mail might be routed through multiple servers. For complete SMTP security the entire chain of SMTP connections must have TLS support.

Finally, the level of negotiated authentication and privacy between each pair of servers or a client and server pair must be considered. For more information, see "Authentication Services" in *System Administration Guide: Security Services*.

Additional Command-Line Options in Version 8.13 of `sendmail`

The following table describes additional command-line options that are available in version 8.13 of `sendmail`. Other command-line options are described in the `sendmail(1M)` man page.

TABLE 14–16 Command-Line Options Available in Version 8.13 of `sendmail`

Option	Description
`-D` *logfile*	Sends debugging output to the indicated *logfile*, instead of including this information with the standard output.
`-q[!]Q`*substr*	Specifies the processing of quarantined jobs that have this *substr*, which is a substring of the quarantine *reason*. See the description of the `-Q`*reason* option. If ! is added, this option processes quarantined jobs that do not have this *substr*.
`-Q`*reason*	Quarantines a normal queue item with this *reason*. If no *reason* is given, the quarantined queue item is unquarantined. This option works with the `-q[!]Q`*substr* option. The *substr* is a portion (or substring) of the *reason*.

Additional and Revised Configuration File Options in Version 8.13 of `sendmail`

The following table describes the added and revised configuration file options. If you declare any of these options, use one of the following syntaxes.

```
O OptionName=argument          # for the configuration file
-O OptionName=argument         # for the command line
define('m4Name',argument)     # for m4 configuration
```

TABLE 14–17 Configuration File Options Available in Version 8.13 of sendmail

Option	Description
ConnectionRateWindowSize	m4 name: confCONNECTION_RATE_WINDOW_SIZE Argument: *number* Default value: 60 Sets the number of seconds for incoming connections to be maintained.
FallBackSmartHost	m4 name: confFALLBACK_SMARTHOST Argument: *hostname* To ensure that mail gets delivered to the clients, this option provides a well-connected host that serves as a backup (or failover) for MX records that fail.
InputMailFilters	m4 name: confINPUT_MAIL_FILTERS Argument: *filename* Lists the input mail filters for the sendmail daemon.
PidFile	m4 name: confPID_FILE Argument: *filename* Default value: /var/run/sendmail.pid As in previous releases, the file name is macro-expanded before it is opened. Additionally, in version 8.13, the file is unlinked when sendmail exits.
QueueSortOrder	m4 name: confQUEUE_SORT_ORDER Added argument: none In version 8.13 none is used to specify no sorting order.
RejectLogInterval	m4 name: confREJECT_LOG_INTERVAL Argument: *period-of-time* Default value: 3h, which represents three hours. When a daemon connection is refused for the *period-of-time* specified, the information is logged.

TABLE 14–17 Configuration File Options Available in Version 8.13 of sendmail *(Continued)*

Option	Description
SuperSafe	m4 name: confSAFE_QUEUE Short name: s Added argument: postmilter Default value: true If postmilter is set, sendmail defers synchronizing the queue file until all milters have signaled acceptance of the message. For this argument to be useful, sendmail must be running as an SMTP server. Otherwise, postmilter operates as if you are using the true argument.

Additional and Revised FEATURE() Declarations in Version 8.13 of sendmail

The following table describes the added and revised FEATURE() declarations. This m4 macro uses the following syntax.

FEATURE(‘*name*’, ‘*argument*’)

TABLE 14–18 FEATURE() Declarations Available in Version 8.13 of sendmail

Name of FEATURE()	Description
conncontrol	Works with the access_db rule set to check the number of incoming SMTP connections. For details, see /etc/mail/cf/README.
greet_pause	Adds the greet_pause rule set, which enables open proxy and SMTP slamming protection. For details, see /etc/mail/cf/README.
local_lmtp	The default argument continues to be mail.local, which is the LMTP-capable mailer in this Solaris release. However, in version 8.13, if a different LMTP-capable mailer is used, its path name can be specified as a second parameter and the arguments that are passed to the second parameter can be specified in the third parameter. For example: FEATURE(‘local_lmtp’, ‘/usr/local/bin/lmtp’, ‘lmtp’)
mtamark	Provides experimental support for “Marking Mail Transfer Agents in Reverse DNS with TXT RRs” (MTAMark). For details, see /etc/mail/cf/README.
ratecontrol	Works with the access_db rule set to control connection rates for hosts. For details, see /etc/mail/cf/README.
use_client_ptr	If this FEATURE() is enabled, the rule set check_relay overrides its first argument with this argument, $&{client_ptr}.

Changes From Version 8.12 of sendmail

This section contains information about the following topics.

Support for TCP Wrappers From Version 8.12 of sendmail

TCP wrappers provide a way of implementing access controls by checking the address of a host requesting a particular network service against an access control list (ACL). Requests are granted or denied, accordingly. Besides providing this access control mechanism, TCP wrappers also log host requests for network services, which is a useful monitoring function. Examples of network services that might be placed under access control include `rlogind`, `telnetd`, and `ftpd`.

Starting with version 8.12, `sendmail` enables the use of TCP wrappers. This check does not bypass other security measures. By enabling TCP wrappers in `sendmail`, a check has been added to validate the source of a network request before the request is granted. See the `hosts_access`(4) man page.

Note – Support for TCP wrappers in inetd(1M) and sshd(1M) started with the Solaris 9 release.

For information about ACLs, see "Using Access Control Lists to Protect UFS Files" in *System Administration Guide: Security Services*.

submit.cf Configuration File From Version 8.12 of sendmail

Starting with version 8.12, sendmail includes an additional configuration file, /etc/mail/submit.cf. This 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 started by programs (such as mailx), as well as by users. Refer to the descriptions of the -Ac option and the -Am option in the sendmail(1M) man page.
- submit.cf is used in the following operating modes:
  - -bm, which is the default operating mode
  - -bs, which uses standard input to run SMTP
  - -bt, which 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:

- Starting with the Solaris 9 release, submit.cf is provided automatically.
- submit.cf does not require any planning or preliminary procedures prior to the installation of the Solaris 9 release or a more recent release.
- 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.

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 From Version 8.12 of sendmail

With the addition of submit.cf, the following functional changes have occurred:

- Starting with 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 302.
- The mail-submission program mode runs without root privilege, which might prevent sendmail from having access to certain files (such as the .forward files). Therefore, 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 sendmail in daemon mode, you would only prevent the delivery of inbound mail. Starting with 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 "How to Manage Mail Delivery by Using an Alternate Configuration of sendmail.cf" on page 290 for a way to resolve this problem.

Additional or Deprecated Command-Line Options From Version 8.12 of sendmail

The following table describes additional or deprecated command-line options for sendmail. Other command-line options are described in the sendmail(1M) man page.

TABLE 14–19 Additional or Deprecated Command-Line Options From Version 8.12 of 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 "submit.cf Configuration File From Version 8.12 of sendmail" on page 360.
-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 "submit.cf Configuration File From Version 8.12 of sendmail" on page 360.
-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 as of version 8.12. Mail user agents should use the -G argument.

Additional Arguments for the PidFile and ProcessTitlePrefix Options From Version 8.12 of sendmail

The following table describes additional macro-processed arguments for the PidFile and ProcessTitlePrefix options. For more information about these options, see the sendmail(1M) man page.

TABLE 14–20 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)

Additional Defined Macros From Version 8.12 of sendmail

The following table describes additional 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 14–21 Additional 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 is especially useful for virtual hosting.

TABLE 14–21 Additional 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 14–25.
${nrcpts}	Holds the number of validated recipients.
${ntries}	Holds the number of delivery attempts.
${rcpt_mailer}, ${rcpt_host}, ${rcpt_addr}, ${mail_mailer}, ${mail_host}, ${mail_addr}	Holds the results of parsing the RCPT and MAIL arguments, which is the resolved right-hand side (RHS) triplet from the mail delivery agent ($#*mailer*), the host ($@*host*), and the user ($:*addr*).

Additional Macros From Version 8.12 of sendmail

In this section, you can find a table that describes the additional macros that are used to build the sendmail configuration file.

TABLE 14–22 Additional 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.

TABLE 14–22 Additional Macros Used to Build the sendmail Configuration File *(Continued)*

Macro	Description
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.

Additional MAX Macros From Version 8.12 of sendmail

Use the following 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 14–23 Additional 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.

Additional and Revised m4 Configuration Macros From Version 8.12 of sendmail

This section contains a table of additional 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 “Changing the sendmail Configuration” on page 282 in Chapter 13, “Mail Services (Tasks).”

TABLE 14–24 Additional and Revised m4 Configuration Macros for sendmail

m4 Macro	Description
FEATURE()	For details, refer to "Changes to the FEATURE() Declaration From Version 8.12 of sendmail" on page 366.
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 From Version 8.12 of sendmail

Refer to the following tables for information about the specific changes to the FEATURE() declarations.

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 "Changing the sendmail Configuration" on page 282 in Chapter 13, "Mail Services (Tasks)."

TABLE 14–25 Additional 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. This FEATURE() is delimited by the following string, <@>. *sender*@*sdomain*<@>*recipient*@*rdomain* is an example.
delay_checks	Argument: friend, which enables a spam-friend test, or hater, which enables a 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 /etc/mail/cf/README file.

TABLE 14–25 Additional and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
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 /etc/mail/cf/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.
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.

TABLE 14–25 Additional and Revised FEATURE() Declarations *(Continued)*

Name of FEATURE()	Description
nouucp	Argument: reject, which does not allow the ! token, or nospecial, which does allow the ! token. A FEATURE() that determines whether 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 antispam 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.
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 14–26 Unsupported FEATURE() Declarations

Name of FEATURE()	Replacement
rbl	FEATURE('dnsbl') and FEATURE('enhdnsbl') replace this FEATURE(), which has been removed.

TABLE 14–26 Unsupported FEATURE() Declarations *(Continued)*

Name of FEATURE()	Replacement
remote_mode	MASQUERADE_AS('$S') replaces FEATURE('remote_mode') in /etc/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 From Version 8.12 of sendmail

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 and sendmail" on page 320. If you need to build a new sendmail.cf file, refer to "Changing the sendmail Configuration" on page 282 in Chapter 13, "Mail Services (Tasks)."

Additional Delivery Agent Flags From Version 8.12 of sendmail

The following table describes additional 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 14–27 Additional 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.

Additional Equates for Delivery Agents From Version 8.12 of sendmail

The following table describes additional 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.

M*agent-name*, *equate*, *equate*, ...

The following example includes the new W= equate. This equate 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 "Changing the sendmail Configuration" on page 282 in Chapter 13, "Mail Services (Tasks)."

Note – Typically, you modify the equate definitions in the mailer directory only when you fine-tune.

TABLE 14–28 Additional 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

Additional Queue Features From Version 8.12 of sendmail

The following list provides details about additional 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 nondaemon 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.
- Starting with version 8.12, 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 302.
- 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 302.

Changes for LDAP From Version 8.12 of 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 /etc/mail/cf/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 dividing 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*.
- In addition to using the %s token to parse an LDAP filter specification, you can 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.

TABLE 14–29 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 additional LDAP map flags.

TABLE 14–30 Additional 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.

Change to the Built-In Mailer From Version 8.12 of `sendmail`

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 14–31 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

Additional Rule Sets From Version 8.12 of `sendmail`

The following table lists the additional rule sets and describes what the rule sets do.

TABLE 14–32 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.

TABLE 14–32 New Rule Sets *(Continued)*

Set	Description
check_etrn	Uses the ETRN command (as check_rcpt uses RCPT).
check_expn	Uses the EXPN command (as check_rcpt uses RCPT).
check_vrfy	Uses the VRFY command (as check_rcpt uses RCPT).

The following list describes additional rule set features.

- Numbered rule sets are also named, but the rule sets 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 From Version 8.12 of sendmail

Note the following changes.

- Starting in the Solaris 10 release, to support a read-only /usr file system, the contents of the /usr/lib/mail directory has been moved to the /etc/mail/cf directory. For details, refer to "Contents of the /etc/mail/cf Directory" on page 331. Note, however, that the shell scripts /usr/lib/mail/sh/check-hostname and /usr/lib/mail/sh/check-permissions are now in the /usr/sbin directory. See "Other Files Used for Mail Services" on page 334. For backward compatibility, symbolic links point to each file's new location.
- The new name for /usr/lib/mail/cf/main-v7sun.mc is /etc/mail/cf/cf/main.mc.
- The new name for /usr/lib/mail/cf/subsidiary-v7sun.mc is /etc/mail/cf/cf/subsidiary.mc.
- 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, no change is made. 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, no change is made. The default content is zero length.

sendmail Version 8.12 and IPv6 Addresses in Configuration

Starting with 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.

PART V

Serial Networking Topics

This section about serial networking provides overview, task, and reference information for PPP and UUCP.

◆ ◆ ◆ CHAPTER 15

Solaris PPP 4.0 (Overview)

This section covers serial networking topics. Serial networking refers to the use of a serial interface, such as an RS-232 or V.35 port, to connect two or more computers for data transfer. Unlike LAN interfaces, such as Ethernet, these serial interfaces are used to connect systems that are separated by large distances. PPP (Point-to-Point Protocol) and UUCP (UNIX-to-UNIX CoPy) are distinct technologies that can be used to implement serial networking. When a serial interface is configured for networking, it is made available for multiple users, in much the same way as any other network interface, such as Ethernet.

This chapter introduces Solaris PPP 4.0. This version of PPP enables two computers in different physical locations to communicate with each other by using PPP over a variety of media. Starting with the Solaris 9 release, Solaris PPP 4.0 is included as part of the base installation.

The following topics are discussed:

Solaris PPP 4.0 Basics

Solaris PPP 4.0 implements the Point-to-Point Protocol (PPP), a data link protocol, which is 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 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 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, Tru64 UNIX,and all three major BSD variants:

- FreeBSD
- OpenBSD
- NetBSD

Solaris PPP 4.0 brings the highly configurable features of ANU PPP-2.4 to machines that run the Solaris operating system. Machines that run Solaris PPP 4.0 can easily set up PPP links 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 releases
- Solstice PPP 3.0.1
- Microsoft Windows 98 DUN
- Cisco IOS 12.0 (synchronous)

Which Version of Solaris PPP to Use

Solaris PPP 4.0 is the PPP implementation that is supported. The Solaris 9 release and later releases do not include the earlier Asynchronous Solaris PPP (`asppp`) software. For more information, refer to the following:

- Chapter 23, “Migrating From Asynchronous Solaris PPP to Solaris PPP 4.0 (Tasks)”
- Solaris System Administrator Collection at http://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 communications 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. You can also use a combination of both the configuration file and command-line methods. 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
  - Microsoft client-side callback support

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 512.

Where to Go for More Information About PPP

Many resources with information about PPP can be found in print and online. The following subsections give some suggestions.

Professional Reference Books About PPP

For more information about widely used PPP implementations, including ANU PPP, refer to the following books:

- Carlson, James. *PPP Design, Implementation, and Debugging*. 2nd ed. Addison-Wesley, 2000.
- Sun, Andrew. *Using and Managing PPP*. O'Reilly & Associates, 1999.

Web Sites About PPP

Go to the following web sites for general information about PPP:

- For technical information, FAQs, discussions about Solaris system administration, and earlier versions of PPP, go to the system administrators' resource, http://www.sun.com/bigadmin/home/index.html.
- For modem configuration and advice about many different implementations of PPP, refer to Stokely Consulting's Web Project Management & Software Development web site: http://www.stokely.com/unix.serial.port.resources/ppp.slip.html.

Requests for Comments (RFCs) About PPP

Some useful Internet RFCs about PPP include the following:

- 1661 and 1662, which describe the major features of PPP
- 1334, which describes authentication protocols, such as Password Authentication Protocol (PAP) and Challenge-Handshake Authentication Protocol (CHAP)
- 1332, an informational RFC that 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 About PPP

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)

Also, see the man page for `pppdump`(1M). You can find the PPP-related man pages by using the `man` command.

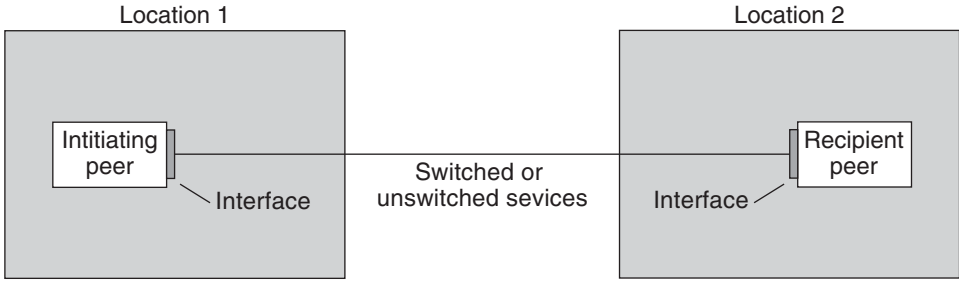
PPP Configurations and Terminology

This section introduces PPP configurations. The section also defines 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 15–1 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`, `hihp`, 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 home computer that calls a peer at an ISP, configured to receive incoming calls. Another scenario is a corporate site where a local machine transmits data over a PPP link to a peer in another building.

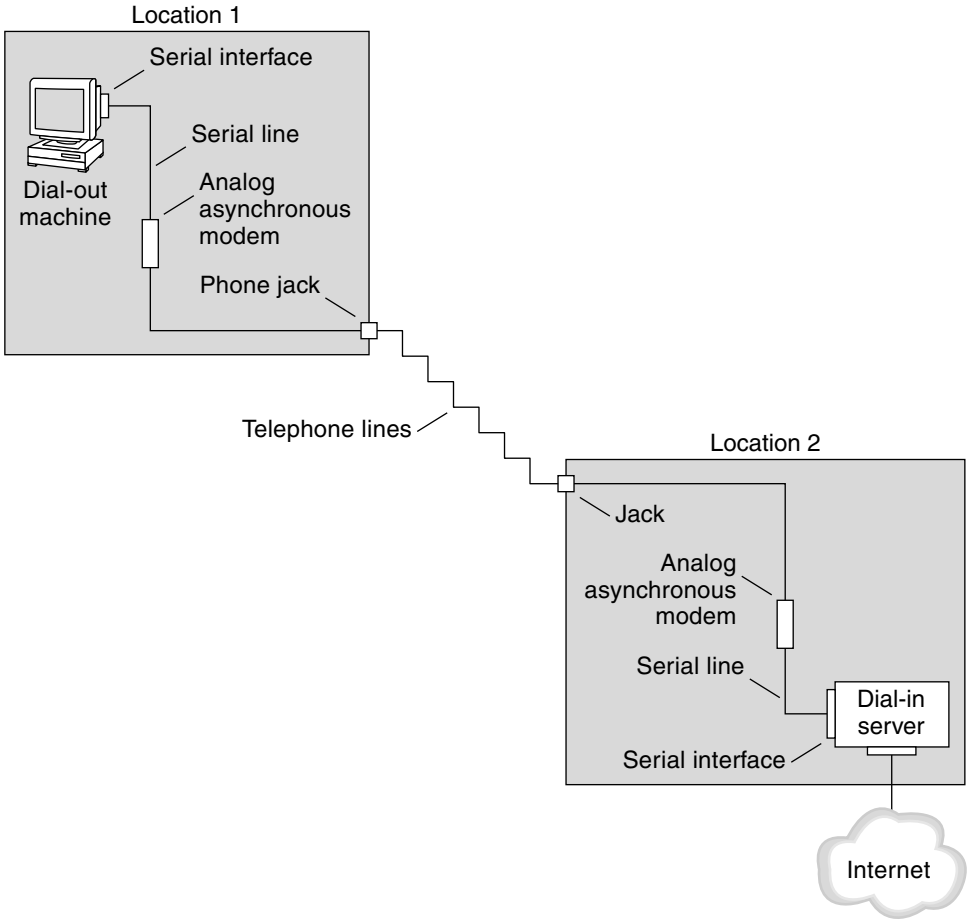
In this guide, the local peer that initiates the dial-up connection is referred to as the *dial-out machine*. The peer that receives the incoming call is referred to as the *dial-in server*. This machine is actually the target peer of the dial-out machine and might or might not be a true server.

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 the target peer of the dial-out machine.

Parts of the Dial-up PPP Link

See the following figure.

FIGURE 15–2 Basic Analog Dial-up PPP Link



The configuration for Location 1, the dial-out 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 that run Solaris software.
- Asynchronous modem or ISDN terminal adapter (TA) that is connected to a telephone jack.
- Telephone lines and services of a telephone company.

The configuration for Location 2, the dial-in side of the link, is composed of the following elements:

- Telephone jack or similar connector, which is 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 TAs 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 488 for information about 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. The dial-out machine then sends instructions over the serial line to its modem, including the phone number of the dial-in server.
3. Modem dials the phone number to establish a telephone connection with the modem on the dial-in server.

   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*. 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.
5. When modem-to-modem negotiation is completed, 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 dial-out machine can then run `telnet` or a similar command to a host that is reachable through the dial-in server.

Leased-Line PPP Overview

A hardwired, *leased-line* PPP configuration involves two peers that are connected by a link. This link consists of a switched or an 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

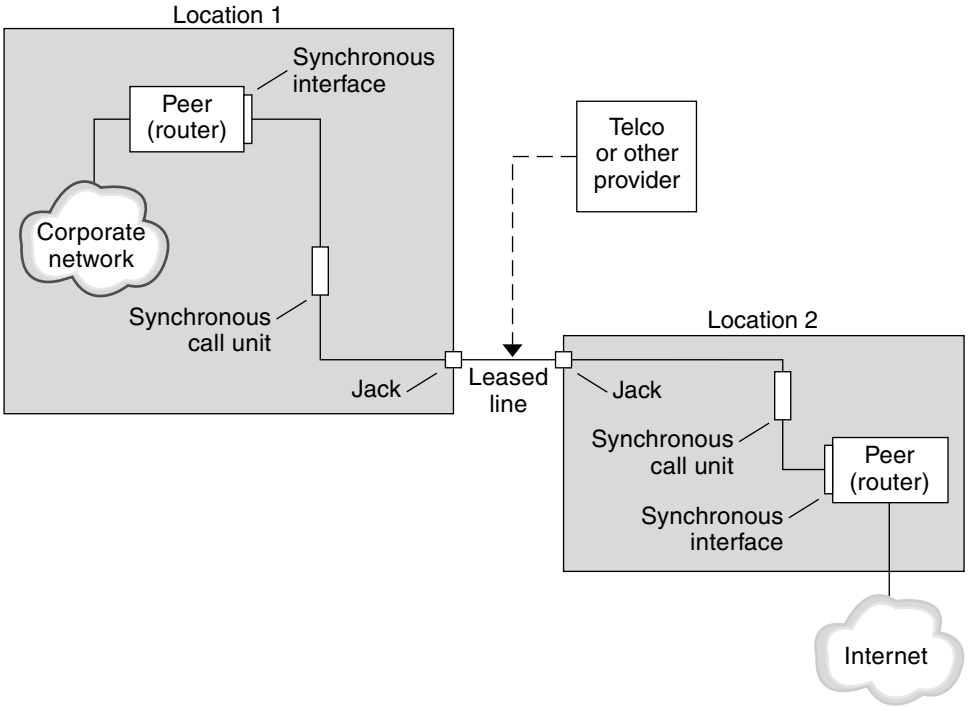
Both dial-up and leased-line links involve two peers that are connected by a communications medium. The next table summarizes the differences between the link types.

Leased Line	Dial-up Line
Always connected, unless a system administrator or power failure takes the leased-line down.	Initiated on demand, when a user tries to call a remote peer.
Uses synchronous and asynchronous communications. For asynchronous communications, a long-haul modem is often used.	Uses asynchronous communications.
Rented from a provider.	Uses existing telephone lines.
Requires synchronous units.	Uses less costly modems.
Requires synchronous ports, which are common on most SPARC systems. However, synchronous ports are not common on x86 systems and newer SPARC systems.	Uses standard serial interfaces that are included on most computers.

Parts of a Leased-Line PPP Link

See the following figure.

FIGURE 15–3 Basic Leased-Line Configuration



The leased-line link contains the following parts:

- **Two peers**, each peer at one 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 that run Solaris software require you to purchase a synchronous interface card, such as HSI/P, to connect to a leased line. Other machines, such as UltraSPARC workstations, 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.

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 connect explicitly 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, the link stays active. A leased-line link remains active unless the 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 another 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 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, the machine always authenticates remote callers.

You might 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 if your local machine must call peers that authenticate callers.

Authenticators and Authenticatees

The calling machine on a PPP link is considered the *authenticatee* because the caller must prove its identity to the remote peer. The peer is considered the *authenticator*. The authenticator 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. This file 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 authentication is not often used in leased-line links. The nature of leased-line contracts usually means that both participants on the ends of the line are known to each other. Both participants 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, see "Password Authentication Protocol (PAP)" on page 491. For a CHAP explanation, see "Challenge-Handshake Authentication Protocol (CHAP)" on page 494.

Why Use PPP Authentication?

Providing authentication on a PPP link is optional. Moreover, though authentication does verify that a peer is to be trusted, PPP authentication does not provide confidentiality of data. For confidentiality, use encryption software, such as IPsec, PGP, SSL, Kerberos, and the Solaris Secure Shell.

Note – Solaris PPP 4.0 does not implement the PPP Encryption Control Protocol (ECP), which is described in RFC 1968.

Consider implementing PPP authentication in the following situations:

- 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, such as `/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 applies 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

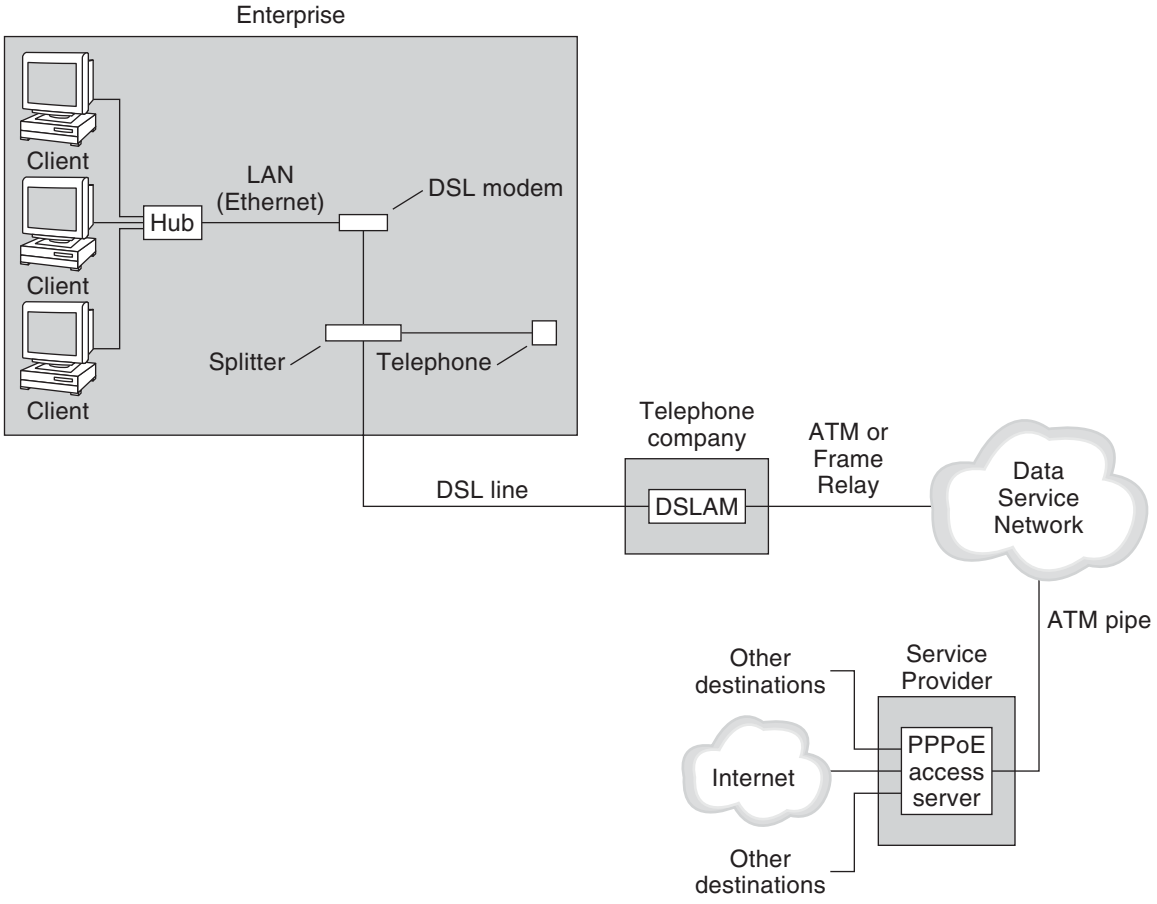
PPPoE is a proprietary protocol from RedBack Networks. PPPoE is a discovery protocol, rather than another version of standard PPP. In a PPPoE scenario, a machine that initiates PPP communications first must locate, or *discover*, a peer that runs PPPoE. The PPPoE protocol uses Ethernet broadcast packets 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. 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 499.

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 15–4 Participants in a PPPoE Tunnel



PPPoE Consumers

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 on 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."

PPPoE 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 15–2, but the access server does not use modems. The access server 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 16

Planning for the PPP Link (Tasks)

Setting up a PPP link involves a set of discrete tasks, which includes 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 16, "Planning for the PPP Link (Tasks)," 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:

- "Planning a Dial-up PPP Link" on page 396
- "Planning a Leased-Line Link" on page 399
- "Planning for Authentication on a Link" on page 401
- "Planning for DSL Support Over a PPPoE Tunnel" on page 407

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 that are 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 16–1 Task Map for PPP Planning

Task	Description	For Instructions
Plan for a dial-up PPP link	Gather information that is required to set up a dial-out machine or a dial-in server	"Planning a Dial-up PPP Link" on page 396
Plan for a leased-line link	Gather information that is required to set up a client on a leased line	"Planning a Leased-Line Link" on page 399

TABLE 16–1 Task Map for PPP Planning *(Continued)*

Task	Description	For Instructions
Plan for authentication on the PPP link	Gather information that is required to configure PAP or CHAP authentication on the PPP link	"Planning for Authentication on a Link" on page 401
Plan for a PPPoE tunnel	Gather information that is required to set up a PPPoE tunnel over which a PPP link can run	"Planning for DSL Support Over a PPPoE Tunnel" on page 407

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 17, "Setting Up a Dial-up PPP Link (Tasks)"

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 to "Dial-up PPP Overview" on page 383.

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 about authentication planning, refer to "Planning for Authentication on a Link" on page 401. For PPPoE planning, refer to "Planning for DSL Support Over a PPPoE Tunnel" on page 407.

TABLE 16–2 Information for a Dial-out Machine

Information	Action
Maximum modem speed	Refer to documentation that was provided by the modem manufacturer.
Modem connection commands (AT commands)	Refer to documentation that was 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 that was required by dial-in server	Contact the dial-in server's administrator or ISP documentation if dial-in server is at the ISP.

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 about authentication planning, refer to "Planning for Authentication on a Link" on page 401. For PPPoE planning, refer to "Planning for DSL Support Over a PPPoE Tunnel" on page 407.

TABLE 16–3 Information for a Dial-in Server

Information	Action
Maximum modem speed	Refer to documentation that was 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 421.
Dedicated IP address for PPP communications	Obtain an address from the individual at your company who is responsible for delegating IP addresses.

Example of a Configuration for Dial-up PPP

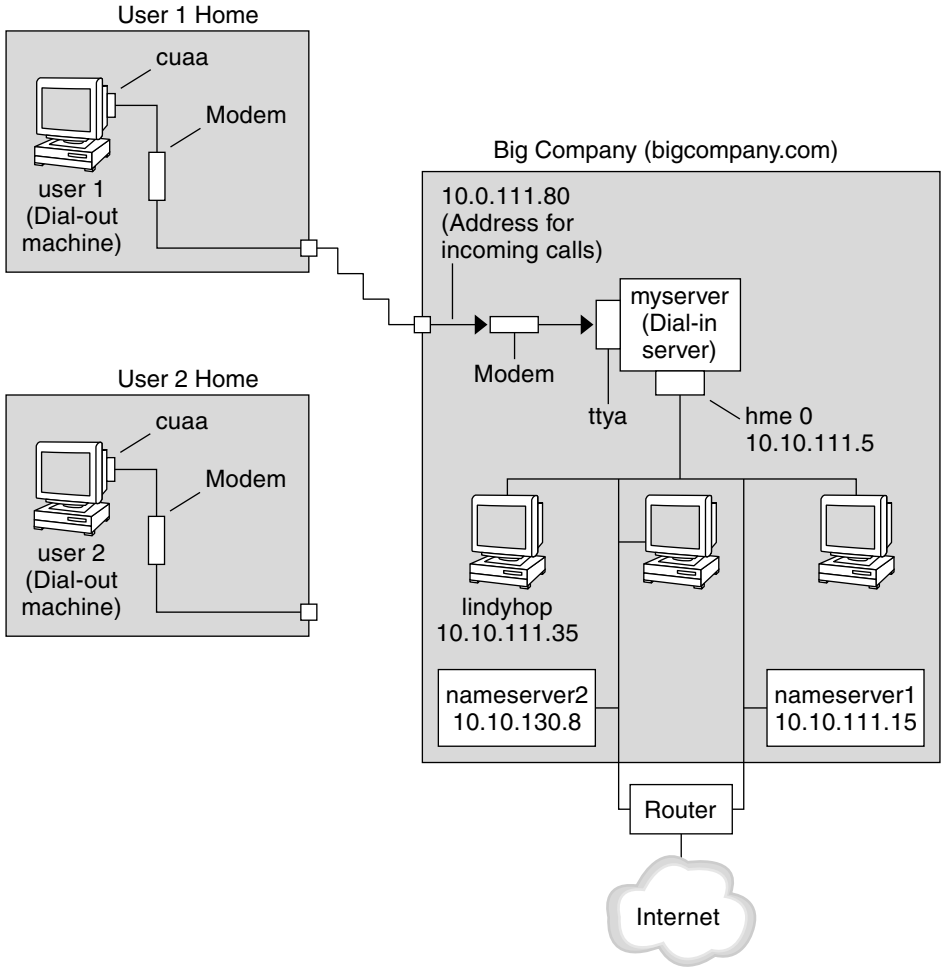
The tasks to be introduced in Chapter 17, "Setting Up a Dial-up PPP Link (Tasks)," execute a small company's requirement to let employees work at home a few days a week. Some employees require the Solaris OS on their home machines. These workers also need to log in remotely to their work machines on 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 17, "Setting Up a Dial-up PPP Link (Tasks)."

FIGURE 16–1 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. The calls from remote users are answered in the order received by the modem that is attached to the dial-in server at Big Company. A PPP connection is established between the peers. The dial-out machine can then remotely log in to a host machine on the Intranet.

Where to Go for More Information About Dial-up PPP

Refer to the following:

- To set up a dial-out machine, see Table 17–2.
- To set up a dial-in machine, see Table 17–3.
- To get an overview of dial-up links, see "Dial-up PPP Overview" on page 383.
- To get detailed information about PPP files and commands, see "Using PPP Options in Files and on the Command Line" on page 471.

Planning a Leased-Line Link

Setting up a leased-line link involves configuring the peer at one end of a switched or unswitched service that is leased from a provider.

This section includes the following information:

- Planning information for a leased-line link
- Explanation of the sample link that is shown in Figure 16–2

For an introduction to leased-line links, refer to "Leased-Line PPP Overview" on page 387. For tasks about setting up the leased line, see Chapter 18, "Setting Up a Leased-Line PPP Link (Tasks)."

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, 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, based on the provider and any governmental restrictions in your locale. The network provider can give you information about the unit that is needed, if this equipment is not provided with the leased line.

Information to Be Gathered for the Leased-Line Link

Before you configure the local peer, you might need to gather the items that are listed in the next table.

TABLE 16–4 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/P 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. Alternatively, 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. Alternatively, you can contact the system administrator of the remote peer.
(Optional) Speed of the link	Refer to the service provider documentation. Alternatively, you can contact the system administrator of the remote peer.
(Optional) Compression that is used by the remote peer	Refer to the service provider documentation. Alternatively, you can contact the system administrator of the remote peer.

Example of a Configuration for a Leased-Line Link

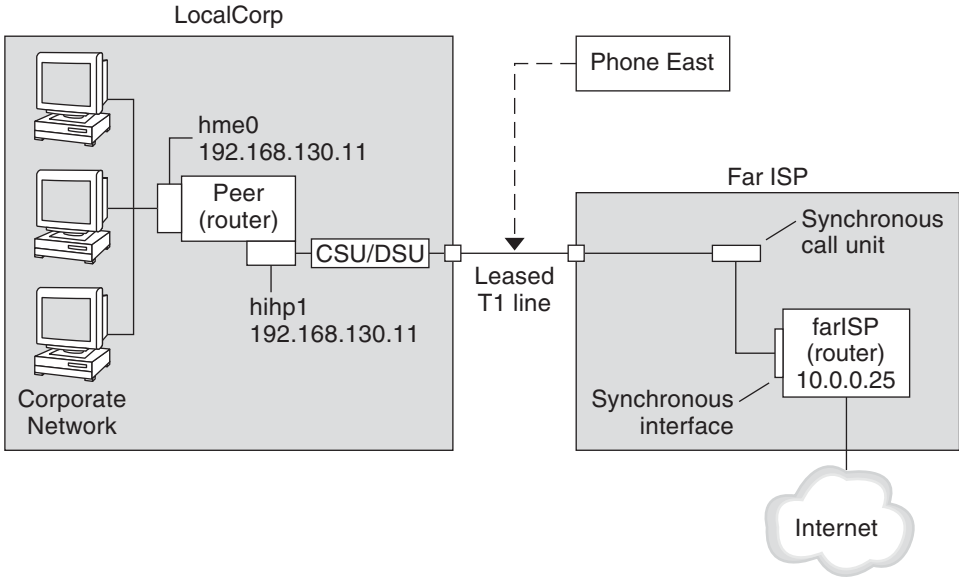
The tasks in Chapter 18, "Setting Up a Leased-Line PPP Link (Tasks)," show how to implement the goal of a medium-sized organization (LocalCorp) to provide Internet access for its employees. Currently, the employees' computers are connected on a private corporate intranet.

LocalCorp requires speedy transactions and access to the many resources on the Internet. The organization signs a contract with Far ISP, a service provider, which 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. Then, Phone East provides a CSU/DSU that is already configured to LocalCorp.

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 16–2 Example of a Leased-Line Configuration



In the figure, a router is set up for PPP at LocalCorp. The router connects to the corporate Intranet through its `hme0` interface. The second connection is through the machine's HSI/P interface (hihp1) to the CSU/DSU digital unit. The CSU/DSU then connects to the installed leased line. The administrator at LocalCorp configures the HSI/P interface and PPP files. The administrator then types `/etc/init.d/pppd` to initiate the link between LocalCorp and Far ISP.

Where to Go for More Information About Leased Lines

Refer to the following:

- Chapter 18, "Setting Up a Leased-Line PPP Link (Tasks)"
- "Leased-Line PPP Overview" on page 387

Planning for Authentication on a Link

This section contains planning information for providing authentication on the PPP link. Chapter 19, "Setting Up PPP Authentication (Tasks)," 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 491 and CHAP, which is described in "Challenge-Handshake Authentication Protocol (CHAP)" on page 494.

Before you set up authentication on a link, you must choose which authentication protocol best meets 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 types of machines. For information about choosing the appropriate authentication protocol for your site, see "Why Use PPP Authentication?" on page 390.

This section includes the following information:

- Planning information for both PAP and CHAP authentication
- Explanations of the sample authentication scenarios that are shown in Figure 16–3 and Figure 16–4

For tasks about setting up authentication, see Chapter 19, "Setting Up PPP Authentication (Tasks)."

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.

TABLE 16–5 Prerequisites Before Configuring Authentication

Information	For Instructions
Tasks for configuring a dial-up link	Chapter 17, "Setting Up a Dial-up PPP Link (Tasks)."
Tasks for testing the link	Chapter 21, "Fixing Common PPP Problems (Tasks)."
Security requirements for your site	Your corporate security policy. If you do not have a policy, 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 390. For more detailed information about these protocols, refer to "Authenticating Callers on a Link" on page 491.

Examples of PPP Authentication Configurations

This section contains examples of authentication scenarios to be used in the procedures in Chapter 19, "Setting Up PPP Authentication (Tasks)."

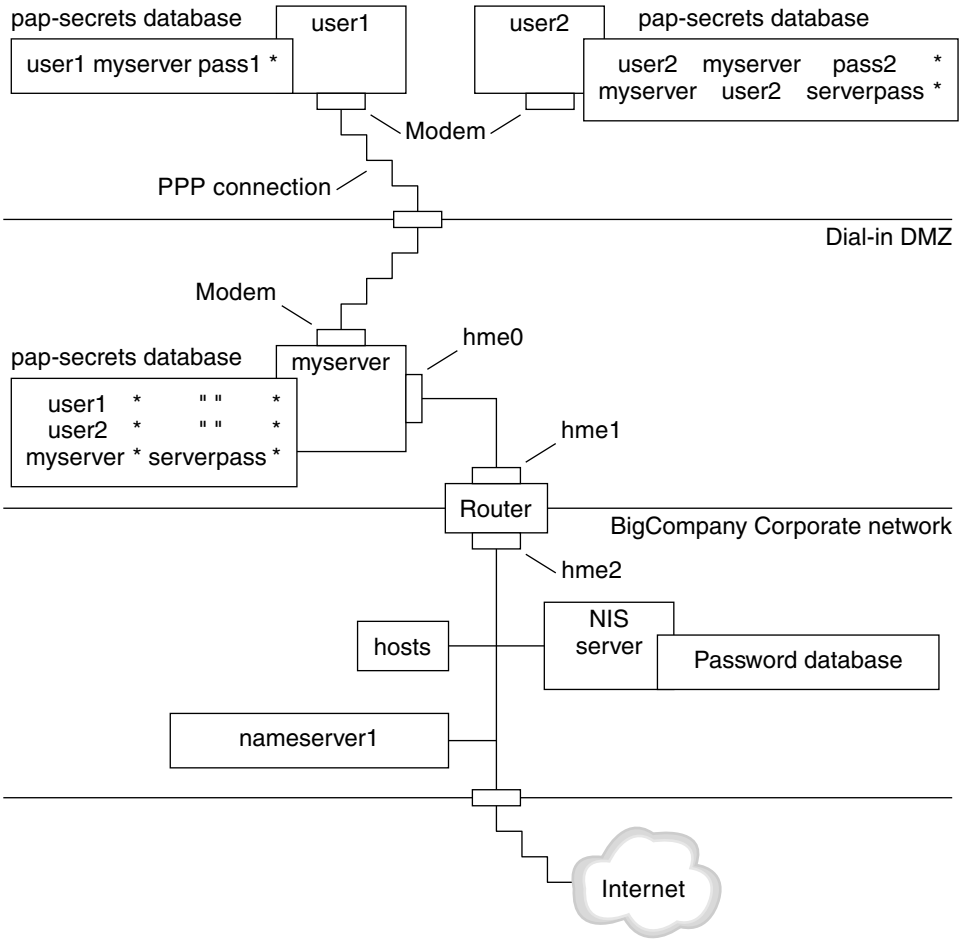
- "Example of a Configuration Using PAP Authentication" on page 403
- "Example of a Configuration Using CHAP Authentication" on page 405

Example of a Configuration Using PAP Authentication

The tasks in "Configuring PAP Authentication" on page 432 show how to set up PAP authentication over the PPP link. The procedures use as an example a PAP scenario that was created for the fictitious "Big Company" in "Example of a Configuration for Dial-up PPP" on page 397.

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 Big Company'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, the administrators implement the following scenario that uses PAP authentication.

FIGURE 16–3 Example of a 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 “demilitarized zone.” The DMZ is an isolated network that is set up for security purposes. The DMZ 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 that is pictured in Figure 16–3 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 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 on the Internet.

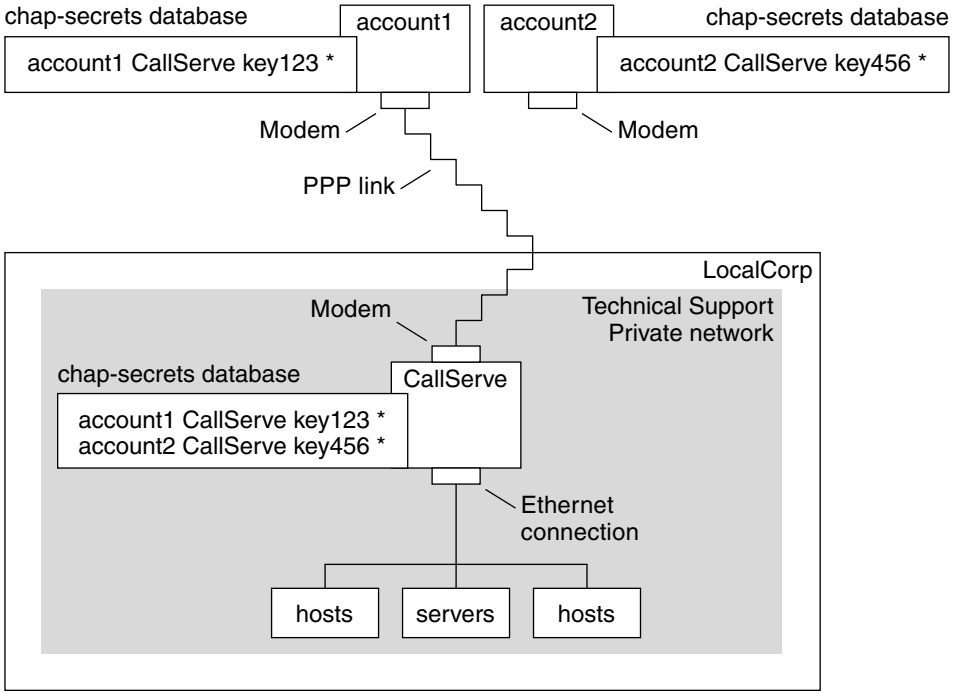
Example of a Configuration Using CHAP Authentication

The tasks in "Configuring CHAP Authentication" on page 439 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 of a Configuration for a Leased-Line Link" on page 400.

LocalCorp provides connectivity to the Internet over a leased line to an ISP. The Technical Support department within LocalCorp generates heavy network traffic. Therefore, Technical Support 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 in the private network's database, remote callers must be authenticated in order to be granted permission to log in.

Therefore, the system administrators implement the following CHAP authentication scenario for a dial-up PPP configuration.

FIGURE 16–4 Example of a CHAP Authentication Scenario (Calling a Private Network)



The only link from the Technical Support network to the outside world is the serial line to the dial-in server's end of the 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 Go for More Information About Authentication

Choose from the following:

- See "Configuring PAP Authentication" on page 432.
- See "Configuring CHAP Authentication" on page 439.
- See "Authenticating Callers on a Link" on page 491 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 391.

A PPPoE tunnel involves three participants: a consumer, a telephone company, and an ISP. You either configure PPPoE for consumers, such as PPPoE clients at your company or consumers in their homes, or you configure PPPoE 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 that is introduced in "Example of a Configuration for a PPPoE Tunnel" on page 408

For tasks about setting up a PPPoE tunnel, see Chapter 20, "Setting Up a PPPoE Tunnel (Tasks)."

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. The telephone company 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 that is 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 true for corporate DSL consumers
- (Optional) A splitter, usually installed by the provider

Many different DSL configurations are possible, which depend on the user or corporation's needs and the services that are offered by the provider.

TABLE 16–6 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 for any required setup procedures.
If setting up PPPoE clients at a corporate site, gather the names of users who are being assigned PPPoE client systems. If you configure remote PPPoE clients, you might be responsible for giving users information about adding home DSL equipment.	Ask management at your company for a list of authorized users.
Find out which interfaces are available on the PPPoE client.	Run the `ifconfig -a` command on each machine for interface names.
(Optional) Obtain the password for the PPPoE client.	Ask users for their preferred passwords. Or, assign passwords to the users. 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 that your company provides. For example, you need to configure interfaces for Internet access, as well as the Ethernet interfaces from the telephone company's bridge.

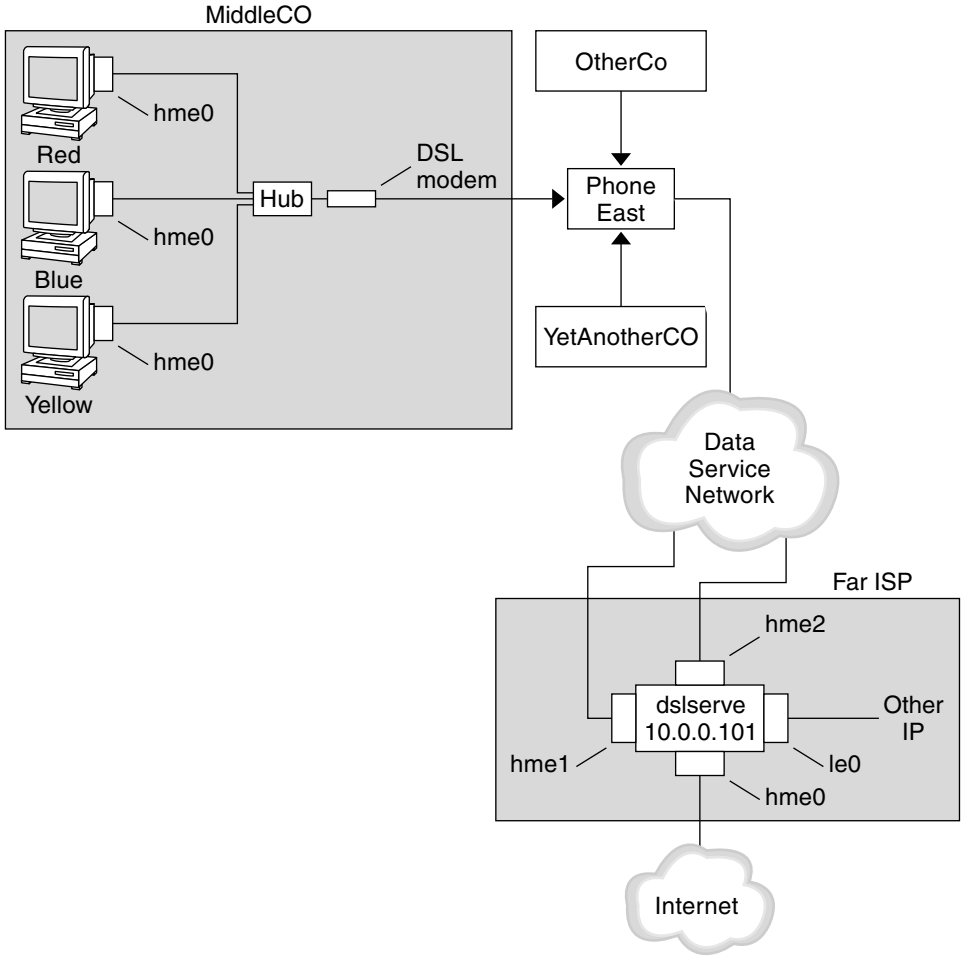
TABLE 16–7 Planning for a PPPoE Access Server

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 and network planners for their requirements and suggestions.
(Optional) Types of services to provide to the consumers	Ask management and network planners for their requirements and suggestions.
(Optional) Host names and passwords for remote clients	Ask network planners and other individuals at your site who are responsible for contract negotiations. The host names and passwords are used for PAP or CHAP authentication, not for UNIX login.

Example of a Configuration for a PPPoE Tunnel

This section contains an example of a PPPoE tunnel, which is used as an illustration for the tasks in Chapter 20, “Setting Up a PPPoE Tunnel (Tasks).” Though the illustration shows all participants in the tunnel, you only administer one end, either the client side or server side.

FIGURE 16–5 Example of a 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 Far ISP. Far ISP offers Internet and other IP services to customers who buy DSL from Phone East.

Example of a 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 of a 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:

- `eri0` – Primary network interface that connects 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

Choose from the following:

- See "Setting Up the PPPoE Client" on page 446.
- See "Setting Up a PPPoE Access Server" on page 448.
- See "Creating PPPoE Tunnels for DSL Support" on page 499, and the `pppoed`(1M), `pppoec`(1M), and `sppptun`(1M) man pages.

◆ ◆ ◆ CHAPTER 17

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 the following:

- "Configuring the Dial-out Machine" on page 412
- "Configuring the Dial-in Server" on page 418
- "Calling the Dial-in Server" on page 423

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 22–1.

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 17–1 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 396
2. Configure the dial-out machine	Set up PPP on the machine that makes the call over the link.	Table 17–2
3. Configure the dial-in server	Set up PPP on the machine that receives incoming calls.	Table 17–3
4. Call the dial-in server	Type the `pppd` command to initiate communications.	"How to Call the Dial-in Server" on page 424

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 dial- in-from-home scenario that was introduced in Figure 16–1. You can perform the tasks at your company before passing on the machine to a prospective user. Alternatively, you can instruct experienced users in the setup of their 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 17–2 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 396
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 413
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 415
4. Define the conversation between the dial-out machine and the peer	Gather communications data for use when you create the chat script.	"How to Create the Instructions for Calling a Peer" on page 416
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 417
6. Call the peer	Type the `pppd` command to initiate communications.	"How to Call the Dial-in Server" on page 424

Dial-up PPP Template Files

Solaris PPP 4.0 provides template files. Each template contains common options for a particular PPP configuration file. The next table lists the sample templates that can be used for setting up a dial-up link, and their equivalent Solaris PPP 4.0 files.

Template File	PPP Configuration File	For Instructions
`/etc/ppp/options.tmpl`	`/etc/ppp/options`	"`/etc/ppp/options.tmpl` Template" on page 476

Template File	PPP Configuration File	For Instructions
`/etc/ppp/options.ttya.tmpl`	`/etc/ppp/options.`*ttyname*	"`options.ttya.tmpl` Template File" on page 478
`/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 484
`/etc/ppp/peers/myisp.tmpl`	`/etc/ppp/peers/`*peer-name*	"`/etc/ppp/peers/myisp.tmpl` Template File" on page 481

If you decide to use one of the template files, be sure to rename the template to its equivalent PPP configuration file. The one exception is the chat file template `/etc/ppp/myisp-chat.tmpl`. You can choose any name for your chat script.

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 release 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 16–2.

▼ 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 list shows the basic parameter settings for modems that use Solaris PPP 4.0.

- **DCD –** Follow carrier instructions
- **DTR –** Set low so that the modem hangs up and puts the modem on-hook
- **Flow Control –** Set to RTS/CTS for full-duplex hardware flow control

- **Attention Sequences** – Disable

If you have problems setting up the link and suspect that the modem is at fault, first consult the modem manufacturer's documentation. Also, a number of web sites offer help with modem programming. Finally, you can find some suggestions for clearing modem problems in "How to Diagnose Modem Problems" on page 462.

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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

4. **Run the `/usr/sadm/bin/smc` command, as explained in "Setting Up Terminals and Modems With Serial Ports Tool (Overview)" in *System Administration Guide: Advanced Administration*. This command opens the Solaris Management Console.**

   Use the Solaris Management Console to do the following.

   a. **Select the port where you have attached the modem.**

   b. **Specify modem direction as dial-out only.**

      You can set up the modem as bidirectional. However, the dial-out-only choice is more secure against possible intruders.

Note – You can set the baud rate and timeout from `/usr/sadm/bin/smc`. 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 413.

The next tasks show how to enable the dial-out machine to successfully initiate communications with the dial-in server. Communications are initiated as defined in the options 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 customize to accommodate your needs. Refer to "Dial-up PPP Template Files" on page 412 for detailed information about these files.

How to Define Communications Over the Serial Line

1 **Become superuser on the dial-out machine or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Create a file that is 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 475.

3 **(Optional) Create a file that is 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`.

```
# cat /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 476.

4 **Set the modem speed, as described in "How to Set the Modem Speed" on page 420.**

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 that is 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 Modem Speed for a Dial-up Link" on page 482.

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 **In a chat script, provide instructions for initiating calls to the particular peer.**

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 486. For complete details about creating a chat script, read the section "Defining the Conversation on the Dial-up Link" on page 482.

Note – You do not invoke the chat script directly. Rather, you use the file name of the chat script as an argument to the `chat` command, which invokes the script.

If a peer runs Solaris or a similar 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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`.

3. **Edit the `/etc/nsswitch.conf` file to have the DNS database searched first for host information.**

   ```
   hosts:      dns [NOTFOUND=return] files
   ```

4. **Create a file for the peer.**

   For example, you would create the following file to define the dial-in server `myserver`:

   ```
   # cat /etc/ppp/peers/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 must time out after an idle period of 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.

See Also The following list provides references to related information.

- To configure another dial-out machine, see "How to Configure the Modem and Serial Port (Dial-out Machine)" on page 413.
- To test modem connectivity by dialing out to another computer, see `cu(1C)` and `tip(1)` man pages. These utilities can help you test if your modem is properly configured. Also, use these utilities to test if you can establish a connection with another machine.
- To learn more about the configuration files and options, see "Using PPP Options in Files and on the Command Line" on page 471.
- To configure a dial-in server, see "Configuring Devices on the Dial-in Server" on page 419.

Configuring the Dial-in Server

The tasks in this section are for configuring the dial-in server. The dial-in server is a 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` that was introduced in Figure 16–1.

Tasks for Configuring the Dial-in Server (Task Map)

TABLE 17–3 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 396

TABLE 17–3 Task Map for Setting Up the Dial-in Server *(Continued)*

Task	Description	For Instructions
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 419
3. Configure calling peer information	Set up the user environments and PPP options for every dial-out machine that is permitted to call the dial-in server.	"How to Configure Users of the Dial-in Server" on page 421
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 422

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 release
- 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 413.

2. **Attach the modem to the serial port on the dial-in server.**

3. **Become superuser on the dial-in server or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

4. **Configure the serial port by using the `/usr/sadm/bin/smc` command for the Solaris Management Console, as described in "Setting Up Terminals and Modems With Serial Ports Tool (Overview)" in *System Administration Guide: Advanced Administration*.**

   Use the Solaris Management Console to do the following:

   a. **Select the serial port where you have attached the modem.**

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 about speeds to use with Sun Microsystems' computers, see "Configuring Modem Speed for a Dial-up Link" on page 482.

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 `/usr/sadm/bin/smc`, as discussed in "Setting Up Terminals and Modems With Serial Ports Tool (Overview)" in *System Administration Guide: Advanced Administration*.**

See Also The following list provides references to related information.

- "How to Configure the Modem and Serial Port (Dial-in Server)" on page 419
- "How to Configure Users of the Dial-in Server" on page 421

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 419.

- Dedicated an IP address to be assigned to incoming calls from remote users. Consider creating a dedicated incoming IP address if 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 497.

How to Configure Users of the Dial-in Server

1. **Become superuser on the dial-in server or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Create a new account on the dial-in server for each remote PPP user.**

   You can use the Solaris Management Console to create a new user. The `/usr/sadm/bin/smc` command opens the Solaris Management Console. For instructions about creating a new user through Solaris Management Console, see "Setting Up User Accounts (Task Map)" in *System Administration Guide: Basic Administration*.

3. **Use Solaris Management Console to assign parameters for the new user.**

   For example, the following table shows the parameters for an account that is called `pppuser` for `user1` on the dial-out machine `myhome`.

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 that is found in the chat script in "How to Create the Instructions for Calling a Peer" on page 416.
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/.pppprc` file that contains various options that are specific to the user's PPP session.**

   For example, you might create the following `.pppprc` file for `pppuser`.

   ```
   # cat /export/home/pppuser/.pppprc
   noccp
   ```

`noccp` turns off compression control on the link.

See Also The following list provides references to related information.

- "How to Configure Users of the Dial-in Server" on page 421.
- "How to Define Communications Over the Serial Line (Dial-in Server)" on page 422.

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. The options that are defined in the following PPP configuration files determine how communications are established.

- `/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 471.

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 419.
- 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 421.

▼ How to Define Communications Over the Serial Line (Dial-in Server)

1 Become superuser on the dial-in server or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Create the `/etc/ppp/options` file with the following entry.

```
nodefaultroute
```

`nodefaultroute` indicates that no `pppd` session on the local system can establish a default route without `root` privileges.

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 that are calling in over serial port `ttya`
`xonxoff`	Allows the serial line to handle communications from modems with software flow control enabled

See Also If you have followed all the procedures in this chapter, you have completed the configuration of the dial-up link. The following list provides references to related information.

- To test modem connectivity by dialing out to another computer, see `cu(1C)` and `tip(1)` man pages. These utilities can help you test if your modem is properly configured. Also, use these utilities to test if you can establish a connection with another machine.
- To configure more options for the dial-in server, see "Configuring the Dial-in Server" on page 418.
- To configure more dial-out machines, see "Configuring the Dial-out Machine" on page 412.
- To have the remote machine call the dial-in server, see "Calling the Dial-in Server" on page 423.

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 local PPP configuration files. However, 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 412
- Set up the dial-in server, as described in "Configuring the Dial-in Server" on page 418

▼ 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 57600 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 417

3 **Contact a host on the server's network, for example, the host `lindyhop` that is shown in Figure 16–1:**

```
ping lindyhop
```

If the link is not working correctly, refer to Chapter 21, "Fixing Common PPP Problems (Tasks)."

4 **Terminate the PPP session:**

```
% pkill -x pppd
```

See Also If you have followed all the procedures in this chapter, you have completed the configuration of the dial-up link. The following list provides references to related information.

- To have users start working on their dial-out machines, see "How to Call the Dial-in Server" on page 424.
- To fix problems on the link, see Chapter 21, "Fixing Common PPP Problems (Tasks)."
- To learn more about the files and options that are used in this chapter, see "Using PPP Options in Files and on the Command Line" on page 471.

◆ ◆ ◆ CHAPTER 18

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 the following:

Setting Up a Leased Line (Task Map)

Leased-line links are relatively easy to set up, in comparison with 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 that are involved in setting up the basic leased-line link.

Note – Some types of leased lines do require the CSU/DSU to "dial" the address of the opposite peer. For example, Frame Relay uses Switched Virtual Circuits (SVCs) or Switched 56 service.

TABLE 18–1 Task Map for Setting Up the Leased-Line Link

Task	Description	For Instructions
1. Gather preconfiguration information	Gather data that is needed prior to setting up the link.	Table 16–4
2. Set up the leased-line hardware	Assemble the CSU/DSU and synchronous interface card.	"How to Configure Synchronous Devices" on page 426
3. Configure the interface card, if required	Configure the interface script to be used when the leased line is initiated.	"How to Configure Synchronous Devices" on page 426

TABLE 18–1 Task Map for Setting Up the Leased-Line Link *(Continued)*

Task	Description	For Instructions
4. 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 427
5. Start up the leased line	Configure your machine to start up PPP over the leased line as part of the booting process.	"How to Configure a Machine on a Leased Line" on page 427

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 of a Configuration for a Leased-Line Link" on page 400. 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 that is installed at your site by the provider
- Synchronous unit (CSU/DSU)
- Solaris 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 necessary.**

Follow the instructions in the manufacturer's documentation.

2 **Connect the cables from the CSU/DSU to the interface.**

If necessary, connect cables 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 that is shown in Figure 16–2 uses an HSI/P interface card.

The following script, `hsi-conf`, starts the HSI/P interface.

```
#!/bin/ksh
/opt/SUNWconn/bin/hsip_init hihp1 speed=1536000 mode=fdx loopback=no \
nrzi=no txc=txc rxc=rxc txd=txd rxd=rxd signal=no 2>&1 > /dev/null
```

`hihp1`	Indicates that HSI/P is the synchronous port used
`speed=1536000`	Set to indicate the speed of the CSU/DSU

See Also To configure the local machine on the leased line, refer to "How to Configure a Machine on a Leased Line" on page 427.

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 of a Configuration for a Leased-Line Link" on page 400 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 configure the synchronous devices for the link, as described in "Configuring Synchronous Devices on the Leased Line" on page 426
- 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 or networks 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) or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add an entry for the remote peer in the router's `/etc/hosts` file.

```
# cat /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 example `/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 this example leased-line link, you create the file `/etc/ppp/peers/farISP`.

```
# cat /etc/ppp/peers/farISP
init '/etc/ppp/conf_hsi'
local
/dev/hihp1
sync
noauth
192.168.130.10:10.0.0.25
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 the link. `init` then configures the HSI interface by 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. Also tells `pppd` to ignore the Data Carrier Detect (DCD) input signal.
`/dev/hihp1`	Gives the device name of synchronous interface.
`sync`	Establishes synchronous encoding for the link.
`noauth`	Establishes that the local system does not need to demand authentication from the peer. However, the peer could still demand authentication.
`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 to go quiet after issuing maximum number of LCP Configure-Requests and to wait for the peer to start.

Option	Definition
`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. These forms of compression accelerate transmissions on a dial-up link but could 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.

```
# cat /etc/ppp/demand
#!/bin/sh
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 content of `$PPPDIR/demand`.

Code Sample	Explanation
`if [ -f /var/run/ppp-demand.pid ] && /usr/bin/kill -s 0 ‘/bin/cat /var/run/ppp-demand.pid‘`	These lines check to see if `pppd` is running. If `pppd` is running, it does not need to be started.
`/usr/bin/pppd call farISP`	This line launches `pppd`. `pppd` reads the options from `/etc/ppp/options`. The `call farISP` option on the command line causes it to read `/etc/ppp/peers/farISP`, also.

The Solaris PPP 4.0 startup script `/etc/rc2.d/S47pppd` invokes the `demand` script as part of the 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 found, `$PPPDIR/demand` is executed. During the course of executing `$PPPDIR/demand`, the link is established.

Note – To reach machines outside the local network, have users run `telnet`, `ftp`, `rsh`, or similar commands.

See Also If you have followed all the procedures in this chapter, you have completed the configuration of the leased-line link. The following list provides references to related information.

- To find troubleshooting information, see "Fixing Leased-Line Problems" on page 468.
- To learn more about the files and options that are used in this chapter, see "Using PPP Options in Files and on the Command Line" on page 471.

◆ ◆ ◆ CHAPTER 19

Setting Up PPP Authentication (Tasks)

This chapter contains tasks for setting up PPP authentication. Subjects that are covered include the following:

- "Configuring PAP Authentication" on page 432
- "Configuring CHAP Authentication" on page 439

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. You can configure authentication over leased lines if authentication is required by your corporate security policy. 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 390. More detailed information about PPP authentication is in the `pppd`(1M) man page and in "Authenticating Callers on a Link" on page 491.

Configuring PPP Authentication (Task Map)

This section contains task maps to help you quickly access procedures for PPP authentication.

TABLE 19–1 Task Map for General PPP Authentication

Task	Description	For Instructions
Configure PAP authentication	Use these procedures to enable PAP authentication on a dial-in server and a dial-out machine.	"Setting Up PAP Authentication (Task Maps)" on page 432
Configure CHAP authentication	Use these procedures to enable CHAP authentication on a dial-in server and a dial-out machine.	"Setting Up CHAP Authentication (Task Maps)" on page 439

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 "Examples of PPP Authentication Configurations" on page 402 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 that belong to trusted callers
- Ideally, for dial-in server authentication, obtained superuser permission for the machine where the network password database is administered, for example, in LDAP, NIS, or local files
- 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 19–2 Task Map for PAP Authentication (Dial-in Server)

Task	Description	For Instructions
1. Gather preconfiguration information	Collect user names and other data that is needed for authentication.	"Planning for Authentication on a Link" on page 401
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 433
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 433
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 435

TABLE 19–3 Task Map for PAP Authentication (Dial-out Machine)

Task	Description	For Instructions
1. Gather preconfiguration information	Collect user names and other data that is needed for authentication.	"Planning for Authentication on a Link" on page 401

TABLE 19–3 Task Map for PAP Authentication (Dial-out Machine) *(Continued)*

Task	Description	For Instructions
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 436
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 438

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 16–3 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 about the `login` option, refer to "Using the `login` Option With `/etc/ppp/pap-secrets`" on page 494.

1 **Assemble a list of all potential trusted callers. Trusted callers are 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 – Verification is particularly important for the sample PAP configuration, which uses the `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 with their UNIX user names. For information about standard `/etc/ppp/pap-secrets`, refer to "`/etc/ppp/pap-secrets` File" on page 492.

Do the following if a potential trusted caller does not have a UNIX user name and password:

a. **Confirm with their managers that callers whom you do not know personally have permission 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

4 **Edit the `/etc/ppp/pap-secrets` file.**

This release provides a `pap-secrets` file in `/etc/ppp` that contains comments about 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 16–3, the trusted caller `user2` requires authentication from remote peers. Therefore, `myserver`'s `/etc/ppp/pap-secrets` file contains PAP credentials for use when a link is established with `user2`.

See Also The following list provides references to related information.

- "Modifying the PPP Configuration Files for PAP (Dial-in Server)" on page 434
- "Configuring PAP Authentication for Trusted Callers (Dial-out Machines)" on page 436

Modifying the PPP Configuration Files for PAP (Dial-in Server)

The tasks in this section explain how to update any 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 as examples the PPP configuration files that were introduced in "How to Define Communications Over the Serial Line (Dial-in Server)" on page 422.

1 Log in as superuser on the dial-in server or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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
auth
login
nodefaultroute
proxyarp
ms-dns 10.0.0.1
idle 120
```

`auth`	Specifies that the server must authenticate callers before establishing the link.
`login`	Specifies that the remote caller be authenticated by using the standard UNIX user authentication services.
`nodefaultroute`	Indicates that no `pppd` session on the local system can establish a default route without `root` privileges.
`proxyarp`	Adds an entry to the system's Address Resolution Protocol (ARP) table that specifies the IP address of the peer and the Ethernet address of the system. With this option the peer appears to be on the local Ethernet to other systems.
`ms-dns 10.0.0.1`	Enables `pppd` to supply a Domain Name Server (DNS) address, `10.0.0.1`, for the client
`idle 120`	Specifies that idle users are disconnected after two minutes.

3 In the `/etc/ppp/options.cua.a` file, add the following address for the `cua/a` user.

```
:10.0.0.2
```

4 In the `/etc/ppp/options.cua.b` file, add the following address for the `cua/b` user.

```
:10.0.0.3
```

5 **In the `/etc/ppp/pap-secrets` file, add the following entry.**

```
*       *           ""      *
```

Note – The `login` option, as previously described, supplies the necessary user authentication. This entry in the `/etc/ppp/pap-secrets` file is the standard way of enabling PAP with the `login` option.

See Also To configure PAP authentication credentials for trusted callers of the dial-in server, refer to "Configuring PAP Authentication for Trusted Callers (Dial-out Machines)" on page 436.

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 systems before distribution to prospective callers. Or, if the remote callers already have their machines, you can give these callers 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

Using the sample PAP configuration that was introduced in Figure 16–3, assume that the dial-out machine belongs to `user1`.

2 **Modify the pap-secrets database for the caller.**

This release 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 – Because most ISPs do not supply authentication credentials, the previous scenario might be unrealistic for communications with an ISP.

See Also The following list provides references to related information.

- "How to Create a PAP Credentials Database (Dial-in Server)" on page 433
- "How to Configure PAP Authentication Credentials for the Trusted Callers" on page 436

Modifying PPP Configuration Files for PAP (Dial-out Machine)

The following tasks 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 16–3. `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 as examples the PPP configuration files that were introduced in "How to Define Communications Over the Serial Line" on page 415. The procedure configures the dial-out machine that belongs to `user2`, as shown in Figure 16–3.

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.

```
# cat /etc/ppp/options
lock
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.

> **Note –** This dial-out machine demands authentication from its peers, even though most dial-out machines do not make this demand. Either way is acceptable.

require-pap — Demands PAP credentials from the peer.

3 Create an `/etc/ppp/peers/`*peer-name* file for the remote machine `myserver`.

The next example shows how to add PAP support to the existing `/etc/ppp/peers/myserver` file that was created in "How to Define the Connection With an Individual Peer" on page 417.

```
# cat /etc/ppp/peers/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

See Also The following list provides references to related information.

- To test the PAP authentication setup by calling the dial-in server, see "How to Call the Dial-in Server" on page 424.
- To learn more about PAP authentication, see "Password Authentication Protocol (PAP)" on page 491.

Configuring CHAP Authentication

The tasks in this section explain how to implement authentication on a PPP link by using the Challenge-Handshake Authentication Protocol (CHAP). The tasks use the example that is shown in Figure 16–4 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 19–4 Task Map for CHAP Authentication (Dial-in Server)

Task	Description	For Instructions
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 441
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 441

TABLE 19–4 Task Map for CHAP Authentication (Dial-in Server) *(Continued)*

Task	Description	For Instructions
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 442

TABLE 19–5 Task Map for CHAP Authentication (Dial-out Machine)

Task	Description	For Instructions
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 441
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 444

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 433. 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 can access the network 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.**

Trusted callers include all 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 or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

4 **Modify the `/etc/ppp/chap-secrets` file.**

This release 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`.

See Also The following list provides references to related information.

- "How to Create a CHAP Credentials Database (Dial-in Server)" on page 441
- "How to Add CHAP Support to the PPP Configuration Files (Dial-in Server)" on page 442
- "Configuring CHAP Authentication for Trusted Callers (Dial-out Machines)" on page 442

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.

```
# cat /etc/ppp/options
lock
nodefaultroute
name CallServe
auth
```

`name` *CallServe*	Defines *CallServe* as the CHAP name of the user on the local machine, in this instance the dial-in server
`auth`	Makes the local machine authenticate callers before establishing the link

3 **Create the remaining PPP configuration files to support the trusted callers.**

See "How to Configure Users of the Dial-in Server" on page 421 and "How to Define Communications Over the Serial Line (Dial-in Server)" on page 422.

See Also To configure CHAP authentication credentials for trusted callers, refer to "How to Create a CHAP Credentials Database (Dial-in Server)" on page 441.

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.

For remote callers to configure CHAP, ensure that the callers' local CHAP secrets match the callers' equivalent CHAP secrets in the dial-in server's `/etc/ppp/chap-secrets` file. Then give the callers 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

   Using the sample CHAP configuration in "Example of a Configuration Using CHAP Authentication" on page 405, assume that the dial-out machine belongs to trusted caller `account1`.

2. **Modify the chap-secrets database for caller `account1`.**

   This release includes an `/etc/ppp/chap-secrets` file that has helpful comments but no options. You can add the following options to the 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

   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`.

See Also The following list provides references to related information.

- "How to Create a CHAP Credentials Database (Dial-in Server)" on page 441
- "How to Configure CHAP Authentication Credentials for the Trusted Callers" on page 443

Adding CHAP to the Configuration Files (Dial-out Machine)

To learn more about CHAP authentication, refer to "Challenge-Handshake Authentication Protocol (CHAP)" on page 494. The next task configures the dial-out machine that belongs to caller `account1`, which is introduced in "Example of a Configuration Using CHAP Authentication" on page 405.

▼ 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.**

```
# cat /etc/ppp/options
lock
nodefaultroute
```

3. **Create an `/etc/ppp/peers/`*peer-name* file for the remote machine `CallServe`.**

```
# cat /etc/ppp/peers/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 417.

See Also To test CHAP authentication by calling the dial-in server, refer to "How to Call the Dial-in Server" on page 424.

◆ ◆ ◆ CHAPTER 20

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 the following:

- "Major Tasks for Setting Up a PPPoE Tunnel (Task Maps)" on page 445
- "Setting Up the PPPoE Client" on page 446
- "Setting Up a PPPoE Access Server" on page 448

The tasks use the scenario that was introduced in "Planning for DSL Support Over a PPPoE Tunnel" on page 407 as an example. For an overview of PPPoE, refer to "Support for DSL Users Through PPPoE" on page 391.

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 20–1 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 446
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 447
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 415
4. Create the tunnel	Call the access server.	"How to Define a PPPoE Access Server Peer" on page 447

TABLE 20–2 Task Map for Setting Up a PPPoE Access Server

Task	Description	For Instructions
1. Set up a PPPoE access server	Define the Ethernet interface to be used for the PPPoE tunnel and define the services that the access server offers.	"How to Set Up a PPPoE Access Server" on page 449
2. 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 422
3. (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 450

Setting Up the PPPoE Client

To provide PPP to client systems over DSL, you must first configure PPPoE on the interface that is connected to the modem or hub. Then you need to change 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 release 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. These devices include, for example, the DSL modem and the splitter, which the telephone company rather than you might assemble.

▼ How to Configure an Interface for a PPPoE Client

Use this procedure to define the Ethernet interface to be used for the PPPoE tunnel.

1 Become superuser on the PPPoE client or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 500.

3 **Configure the interface for PPPoE use.**

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

4 **(Optional) Verify that the interface is now plumbed for PPPoE.**

```
# /usr/sbin/sppptun query
hme0:pppoe
hme0:pppoed
```

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 500.

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 480.

1 **Become superuser on the PPPoE client or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 Far ISP that is introduced in "Example of a Configuration for a PPPoE Tunnel" on page 408.

```
# 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 507.

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 412.**

b. Create an `/etc/ppp/options.sppptun` file. `/etc/ppp/options.sppptun` defines PPP options for the serial port to which the interface that is plumbed for PPPoE is attached.

You can use any options that are available for the `/etc/ppp/options.`*ttyname* file that is described in "`/etc/ppp/options.`*ttyname* Configuration File" on page 476. 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.

```
# touch /etc/ppp/options
```

5 Test if PPP can run over the DSL line.

```
% pppd debug updetach call dslserve
```

`dslserve` is the name that is given to the access server at the ISP that is shown in "Example of a Configuration for a PPPoE Tunnel" on page 408. The `debug updetach` option causes debugging information to be displayed in a terminal window.

If PPP is running correctly, the terminal output shows the link becoming active. If PPP still does not run, try the following command to see if the servers are running correctly:

```
# /usr/lib/inet/pppoec -i hme0
```

Note – Users of configured PPPoE clients can begin running PPP over a DSL line by typing the following:

```
% pppd call ISP-server-name
```

Then the users can run an application or a service.

See Also The following list provides references to related information.

- See "Setting Up the PPPoE Client" on page 446.
- See "Creating PPPoE Tunnels for DSL Support" on page 499.
- See Chapter 21, "Fixing Common PPP Problems (Tasks)."
- See "Setting Up a PPPoE Access Server" on page 448.

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. The procedure involves determining which interfaces on the server to involve in the PPPoE tunnel and defining which services are made available to the users.

How to Set Up a PPPoE Access Server

Use this procedure to define the Ethernet interface to be used for the PPPoE tunnel and to configure the services that the access server offers.

1. **Become superuser on the access server or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add the name of the Ethernet interfaces that are 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` that is shown in "Example of a Configuration for a PPPoE Tunnel" on page 408.

   ```
   # cat /etc/ppp/pppoe.if
   hme1
   hme2
   ```

3. **Define global services that are provided by the access server in the `/etc/ppp/pppoe` file.**

   The following `/etc/ppp/pppoe` file lists the services that are provided by access server `dslserve`, which was shown in Figure 16–5.

   ```
   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.

4. **Set up the PPP configuration files in the same way that you would for a dial-in server.**

   For more information, refer to "Creating an IP Addressing Scheme for Callers" on page 497.

5. **Start the `pppoed` daemon.**

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

   `pppd` also plumbs the interfaces that are listed in `/etc/ppp/pppoe.if`.

6. **(Optional) Verify that the interfaces on the server are plumbed for PPPoE.**

   ```
   # /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 500.

How to Modify an Existing /etc/ppp/pppoe File

1. **Become superuser on the access server or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Modify /etc/ppp/pppoe, as needed.**

3. **Cause the pppoed daemon to recognize the new services.**

   ```
   # 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 performing 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 ability as a convenience factor, not a security measure.

Using the example that is shown in "Example of a Configuration for a PPPoE Tunnel" on page 408, these steps show how to reserve one of dslserve's interfaces, hme1, for clients at MiddleCo.

1. **Configure the access server's interfaces and define the services, as shown in "How to Set Up a PPPoE Access Server" on page 449.**

2. **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.

3 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.

```
# cat /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 for the `clients` option. Wildcards are allowed.

For example, you can specify the numeric address `clients 8:0:20:*:*:*`. By using wildcards, all matching addresses in `/etc/ethers` are accepted.

4 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 that are allowed to run PPP over `dslserve`'s `hme1` interface.

For more information about PAP authentication, see "Configuring PAP Authentication" on page 432.

See Also The following list provides references to related information.

- To learn more about PPPoE, see "Creating PPPoE Tunnels for DSL Support" on page 499.
- To troubleshoot PPPoE and PPP problems, see "Solving PPP-Related and PPPoE-Related Problems" on page 457.
- To configure a PPPoE client, see "Setting Up the PPPoE Client" on page 446.
- To configure PAP authentication for a client, see "Configuring PAP Authentication for Trusted Callers (Dial-out Machines)" on page 436.
- To configure PAP authentication on a server, see "Configuring PAP Authentication on the Dial-in Server" on page 433.

◆ ◆ ◆ CHAPTER 21

Fixing Common PPP Problems (Tasks)

This chapter contains information for troubleshooting common problems that occur with Solaris PPP 4.0. The following topics are covered:

- "Tools for Troubleshooting PPP" on page 454
- "Solving PPP-Related and PPPoE-Related Problems" on page 457
- "Fixing Leased-Line Problems" on page 468
- "Diagnosing and Fixing Authentication Problems" on page 469

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 "Professional Reference Books About PPP" on page 381 and "Web Sites About PPP" on page 382.

Solving PPP Problems (Task Map)

Use the following task map to quickly access advice and solutions for common PPP problems.

TABLE 21–1 Task Map for Troubleshooting PPP

Task	Definition	For Instructions
Obtain diagnostic information about the PPP link	Use PPP diagnostic tools to obtain output for troubleshooting.	"How to Obtain Diagnostic Information From `pppd`" on page 455
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 456
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 457

TABLE 21–1 Task Map for Troubleshooting PPP *(Continued)*

Task	Definition	For Instructions
Troubleshoot general communications problems	Identify and fix communications problems that affect the PPP link.	"How to Diagnose and Fix Communications Problems" on page 460
Troubleshoot configuration problems	Identify and fix problems in the PPP configuration files.	"How to Diagnose Problems With the PPP Configuration" on page 461
Troubleshoot modem-related problems	Identify and fix modem problems.	"How to Diagnose Modem Problems" on page 462
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 463
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 465
Troubleshoot common problems for leased lines	Identify and fix performance problems on a leased line.	"Fixing Leased-Line Problems" on page 468
Troubleshoot problems related to authentication	Identify and fix problems related to the authentication databases.	"Diagnosing and Fixing Authentication Problems" on page 469
Troubleshoot problem areas for PPPoE	Use PPP diagnostic tools to obtain output for identifying and fixing PPPoE problems.	"How to Obtain Diagnostic Information for PPPoE" on page 466

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 during regular usage
- 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 over the link. Run a command such as `ping` or `traceroute` to a host on the peer's network. Then observe the results. However, you should use PPP and UNIX debugging tools to monitor performance of an established link or to troubleshoot a problematic link.

This section explains how to obtain 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Run pppd with the serial device configured for PPP as the argument:**

   ```
   # pppd cua/b debug updetach
   ```

   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 that is produced is sent to the `/etc/ppp/connect-errors` file.

Example 21–1 Output From a Properly Operating Dial-up Link

```
# pppd /dev/cua/b debug updetach
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., Oct  6
    2004 09:36:22)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Oct  6 2004 09:36:22)
    rcvd [LCP ConfRej id=0x7b <asyncmap 0x0>]
sent [LCP Ident id=0x7c magic=0x0 "ppp-2.4.0b1 (Sun Microsystems, Inc., Sep 15
    2004 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.,
    Sep 15 2004 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.,
    Oct  6 2004 09:36:22)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Oct  6 2004 09:36:22)
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 21–2 Output From a Properly Operating Leased-Line Link

```
# pppd /dev/se_hdlc1 default-asyncmap debug updetach
pppd 2.4.0b1 (Sun Microsystems, Inc., Oct 24 2004 07:13:18) started by root, uid 0
synchronous speed appears to be 0 bps
```

```
init option: '/etc/ppp/peers/syncinit.sh' started (pid 105122)
Serial port initialized.
synchronous speed appears to be 64000 bps
Using interface sppp0
Connect: sppp0 <--> /dev/se_hdlc1
sent [LCP ConfReq id=0xe9 <magic 0x474283c6><pcomp> <accomp>]
rcvd [LCP ConfAck id=0xe9 <magic 0x474283c6><pcomp> <accomp>]
rcvd [LCP ConfReq id=0x22 <magic 0x8e3a53ff><pcomp> <accomp>]
sent [LCP ConfReq id=0x22 <magic 0x8e3a53ff><pcomp> <accomp>]
sent [LCP Ident id=0xea magic=0x474283c6 "ppp-2.4.0b1 (Sun Microsystems, Inc., Oct
    22 2004 14:31:44)"]
sent [IPCP ConfReq id=0xf7 <addr 0.0.0.0> <compress VJ Of o1>]]
sent [CCP ConfReq id=0x3f <deflate 15> <deflate(old#) 15> <bsd v1 15>]
rcvd [LCP Ident id=0x23 magic=0x8e3a53ff "ppp-2.4.0b1 (Sun Microsystems, Inc., Oct
    22 2004 14:31:44)"]
Peer Identification: ppp-2.4.0b1 (Sun Microsystems, Inc., Oct 22 2004 14:31:44)
rcvd [IPCP ConfReq id=0x25 <addr 10.0.0.1> <compress VJ Of 01>]
sent [IPCP ConfAck id=0x25 <addr 10.0.0.1> <compress VJ Of 01>]
rcvd [CCP ConfReq id=0x3 <deflate 15> <deflate(old#) 15 <bsd v1 15>]
sent [CCP ConfAck id=0x3 <deflate 15> <deflate(old#) 15 <bsd v1 15>]
rcvd [IPCP ConfNak id=0xf8 <addr 10.0.0.2>]
rcvd [IPCP ConfReq id=0xf7 <addr 10.0.0.2> <compress VJ Of 01>]
rcvd [CCP ConfAck id=0x3f <deflate 15> <deflate(old#) 15 <bsd v1 15>]
Deflate (15) compression enabled
rcvd [IPCP ConfAck id=0xf8 <addr 10.0.0.2> <compress VJ Of 01>]
local  IP address 10.0.0.2
remote IP address 10.0.0.1
```

How to Turn on PPP Debugging

The next task shows how to use the `pppd` command to obtain debugging information.

Note – You only need to perform step 1 through step 3 once for each host. Thereafter, you can proceed to step 4 to turn on debugging for the host.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 **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.

6 **View the contents of the log file.**

```
# tail -f /var/log/pppdebug
```

For an example of a log file, see Step 3.

Solving PPP-Related and PPPoE-Related Problems

Refer to the following sections for information about how to resolve PPP-related and PPPoE-related problems.

▼ How to Diagnose Network Problems

If the PPP link becomes active but few hosts on the remote network are reachable, a network problem could be indicated. The following procedure shows you how to isolate and fix network problems that affect a PPP link.

1 **Become superuser on the local machine or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Shut down the problematic link.**

3 **Disable any optional protocols in the configuration files by adding the following options to your PPP configuration:**

```
noccp novj nopcomp noaccomp default-asyncmap
```

These options provide the simplest uncompressed PPP that is available. Try to invoke these options as arguments to `pppd` on the command line. If you can reach the previously unreachable hosts, add the options in either of the following places.

- `/etc/ppp/peers/`*peer-name*, after the `call` option
- `/etc/ppp/options`, ensuring that the options apply globally

4 Call the remote peer. Then enable debugging features.

```
% pppd debug call peer-name
```

5 Obtain verbose logs from the chat program by using the `-v` option of `chat`.

For example, use the following format in any PPP configuration file:

```
connect 'chat -v -f /etc/ppp/chatfile'
```

/etc/ppp/chatfile represents the name of your chat file.

6 Try to re-create the problem by using Telnet or other applications to reach the remote hosts.

Observe the debugging logs. If you still cannot reach remote hosts, the PPP problem might be network-related.

7 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 must set up a name-to-address translation (NAT) server 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.

8 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. Also check the routing tables for any routers on the path back to the peer.

Ensure that the intermediate routers have not been misconfigured. Often the problem can be found in the path back to the peer.

9 (Optional) If the machine is a router, check the optional features.

```
# ndd -set /dev/ip ip_forwarding 1
```

For more information about `ndd`, refer to the `ndd`(1M) man page.

In the Solaris 10 release, you can use `routeadm`(1M), instead of `ndd`(1M).

```
# routeadm -e ipv4-forwarding -u
```

Note – The `ndd` command is not persistent. The values set with this command are lost when the system is rebooted. The `routeadm` command is persistent. The values set with this command are maintained after the system is rebooted.

10 Check the statistics that are obtained from `netstat -s` and similar tools.

For complete details about `netstat`, refer to the `netstat(1M)` man page.

a. Run statistics on the local machine.

b. Call the peer.

c. Observe the new statistics that are generated by `netstat -s`. For more information, refer to "Common Network Problems That Affect PPP" on page 459.

11 Check the DNS configuration.

A faulty name service configuration causes applications to fail because IP addresses cannot be resolved.

Common Network Problems That Affect PPP

You can use the messages that are generated by `netstat -s` to fix the network problems that are shown in the following table. For related procedural information, refer to "How to Diagnose Network Problems" on page 457.

TABLE 21–2 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 table.
`ICMP input destination unreachable`	The local host is missing a route.	Add the missing route to the local host's routing tables.

▼ How to Diagnose and Fix Communications Problems

Communications problems occur when the two peers cannot successfully establish a link. Sometimes these problems are actually negotiation problems that are caused by incorrectly configured chat scripts. The following procedure shows you how to clear communication problems. For clearing negotiation problems that are caused by a faulty chat script, see Table 21–5.

1 **Become superuser on the local machine or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Call the peer.**

3 **Call the remote peer. Then enable debugging features.**

```
% pppd debug call peer-name
```

You might need to obtain debugging information from the peer in order to fix certain communications problems.

4 **Check the resulting logs for communication problems. For more information, refer to "General Communications Problems That Affect PPP" on page 460.**

General Communications Problems That Affect PPP

The following table describes symptoms that are related to log output from the procedure, "How to Diagnose and Fix Communications Problems" on page 460.

TABLE 21–3 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: ■ The machine or modem might have faulty cabling. ■ The modem configuration might have incorrect bit settings. Or, the configuration might have broken flow control. ■ The chat script might have failed. In this situation, see Table 21–5.
The `pppd debug` output shows that LCP starts, but higher-level protocols fail or show CRC errors.	The asynchronous control character map (ACCM) is incorrectly set.	Use the `default-async` option to set the ACCM to the standard default of FFFFFFFF. First, try to use `default-async` as an option to `pppd` on the command line. If the problem clears, then add `default-async` to `/etc/ppp/options` or to `/etc/ppp/peers/`*peer-name* after the call option.

TABLE 21–3 General Communications Problems That Affect PPP *(Continued)*

Symptom	Problem	Solution
The `pppd debug` output shows that IPCP starts but terminates immediately.	IP addresses might be incorrectly configured.	1. Check the chat script to verify whether the script has incorrect IP addresses. 2. If the chat script is correct, request debug logs for the peer, and check IP addresses in the peer logs.
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. Adjust the configuration if necessary.

How to Diagnose Problems With the PPP Configuration

Some PPP problems can be traced to problems in the PPP configuration files. The following procedure shows you how to isolate and fix general configuration problems.

1. **Become superuser on the local machine or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Call the remote peer. Then enable debugging features.**

   ```
   % pppd debug call peer-name
   ```

3. **Check the resulting log for the configuration problems. For more information, refer to "Common PPP Configuration Problems" on page 461.**

Common PPP Configuration Problems

The following table describes symptoms that are related to log output from the procedure, "How to Diagnose Problems With the PPP Configuration" on page 461.

TABLE 21–4 Common PPP Configuration Problems

Symptom	Problem	Solution
`pppd debug` output contains the error message, `Could not determine remote IP address`.	The `/etc/ppp/peers/`*peer-name* file does not have an IP address for the peer. The peer does not provide an IP address during link negotiation.	Supply an IP address for the peer on the pppd command line or in `/etc/ppp/peers/`*peer-name* by using the following format: :10.0.0.10
`pppd debug` output shows that CCP data compression has failed. The output also indictes that the link is dropped.	The peers' PPP compression configurations might be in conflict.	Disable CCP compression by adding the `noccp` option to `/etc/ppp/options` on one of the peers.

How to Diagnose Modem 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. However, you might have difficulties when determining 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 contain solutions for problems with their particular equipment. The following procedure helps 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 456.**

2. **Display the resulting `/var/log/pppdebug` log to check for faulty modem configuration.**

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, modem problems are indicated.

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, the modem configuration might have problems.

Troubleshooting When you display the resulting `/var/log/pppdebug` log, the following symptoms in the output can indicate a faulty modem configuration. The local machine can hear the peer, but the peer cannot hear the local machine.

- 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` messages that are sent by the local machine.
- The link terminates with a SIGHUP signal.

How to Obtain Debugging Information for Chat Scripts

Use the following procedure for obtaining debugging information from `chat` and suggestions for clearing common problems. For more information, refer to "Common Chat Script Problems" on page 463.

1. **Become superuser on the dial-out machine or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 that is 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`.**

   The following is the main error that occurs with `chat`.

   ```
   Oct 31 08:57:13 deino chat[107294]: [ID 702911 local2.info] expect (CONNECT)
   Oct 31 08:57:58 deino chat[107294]: [ID 702911 local2.info] alarm
   Oct 31 08:57:58 deino chat[107294]: [ID 702911 local2.info] Failed
   ```

   The example shows timeout while waiting for a `(CONNECT)` string. When `chat` fails, you get the following message from `pppd`:

   ```
   Connect script failed
   ```

Common Chat Script Problems

Chat scripts are trouble-prone areas for dial-up links. The following table lists common chat script errors and gives suggestions for fixing the errors. For procedural information, refer to "How to Obtain Debugging Information for Chat Scripts" on page 463.

TABLE 21–5 Common Chat Script Problems

Symptom	Problem	Solution
`pppd debug` output contains `Connect script failed`	Your chat script supplies a user name and password. `ogin:` *user-name* `ssword:` *password* However, the peer that you intended 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. Ask the ISP for the correct login sequence.

The `/usr/bin/chat -v` log contains `"expect (login:)" alarm read timed out`	Your chat script supplies a user name and password.
	`ogin: pppuser`
	`ssword: \q\U`
	However, the peer that you intend 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. Ask the ISP 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. An incorrectly configured login name and password are 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. Ask for the correct login sequence.
`pppd debug` output shows that LCP activates, 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. Fix the password if incorrect. 3. Try to call the peer again. 4. If you still get the message, call the ISP. Ask the ISP for the correct login sequence.

Text from the peer begins with a tilde (~).	Your chat script supplies a user name and password.
	`ogin: pppuser`
	`ssword: \q\U`
	However, the peer that you intend 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. Request the correct login sequence.

TABLE 21–5 Common Chat Script Problems *(Continued)*

Symptom	Problem	Solution
The modem hangs.	Your chat script contains the following line to force the local machine to 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 `LCP: timeout sending Config-Requests`	Your chat script contains the following line to force the local machine to 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 force the local machine to 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 force the local machine to 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 force the local machine to 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`.

▼ How to Diagnose and Fix Serial-Line Speed Problems

Dial-in servers can experience problems because of conflicting speed settings. The following procedure helps you to isolate the cause of the link problem to conflicting serial-line speeds.

The following behaviors cause speed problems:

- You invoked PPP through a program such as `/bin/login` and specified the speed of the line.
- You started PPP from `mgetty` and accidentally supplied the bit rate.

pppd changes the speed that was originally set for the line to the speed that was set by `/bin/login` or `mgetty`. As a result, the line fails.

1 Log in to the dial-in server. Call the peer with debugging enabled.

If you need instructions, see “How to Turn on PPP Debugging” on page 456.

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 that were configured for PPP might potentially be in conflict.

3. **Check if PPP is invoked through a program such as `/bin/login` and the line speed that 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 serial-line speeds to conflict.

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 configuration.**

How to Obtain Diagnostic Information for PPPoE

You can use PPP and standard UNIX utilities to identify problems with PPPoE. When you suspect that PPPoE is the cause of problems on a link, 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 456.**

3. **View the contents of the log file `/var/log/pppdebug`.**

   The following example shows part of a log file that was generated 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 with this procedure.

4 Get diagnostic messages from PPPoE.

```
# pppd connect "/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 that are generated as the PPPoE tunnel is negotiated.

```
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 with this procedure.

5 Run `snoop`. Then 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

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 -----
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
```

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 21–6 Common Leased-Line Problems

Symptom	Problem	Solution
The link does not start.	CSU biopolar violations (CSU BPVs) can be the cause. One end of the link is set up for AMI lines. The other end is set up for 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 sustained traffic is 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 "loop clocking" is in use. On some unstructured leased lines, you might have to supply clocking. North American users should use loop clocking.

Diagnosing and Fixing Authentication Problems

The following table describes solutions for general authentication problems.

TABLE 21–7 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 starts 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.

◆ ◆ ◆ CHAPTER 22

Solaris PPP 4.0 (Reference)

This chapter provides detailed conceptual information about Solaris PPP 4.0. Topics include the following:

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 configuration is very flexible. 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 22–1 Summary of PPP Configuration Files and Commands

File or Command	Definition	For Information
`/etc/ppp/options`	A 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, nonroot users are prohibited from using PPP.	"`/etc/ppp/options` Configuration File" on page 475
`/etc/ppp/options.`*ttyname*	A file that describes the characteristics of all communications over the serial port *ttyname*.	"`/etc/ppp/options.`*ttyname* Configuration File" on page 476
`/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 for Communicating With the Dial-in Server" on page 479
`/etc/ppp/peers/`*peer-name*	A file that contains characteristics of the remote peer *peer-name*. Typical characteristics include the remote peer's phone number and chat script for negotiating the link with the peer.	"`/etc/ppp/peers/`*peer-name* File" on page 480
`/etc/ppp/pap-secrets`	A file that contains the necessary security credentials for Password Authentication Protocol (PAP) authentication.	"`/etc/ppp/pap-secrets` File" on page 492
`/etc/ppp/chap-secrets`	A file that contains the necessary security credentials for Challenge-Handshake Authentication Protocol (CHAP) authentication.	"`/etc/ppp/chap-secrets` File" on page 495
`~/.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 `$HOME/.ppprc` on a Dial-in Server" on page 479
`pppd` *options*	Command and options for initiating a PPP link and describing its characteristics.	"How PPP Options Are Processed" on page 472

Refer to the pppd(1M) man page for details on the PPP files. `pppd` (1M) also includes comprehensive descriptions of all options that are available to the `pppd` command. Sample templates for all the PPP configuration files are available in `/etc/ppp`.

How PPP Options Are Processed

1. The `pppd` daemon parses the following:

All Solaris PPP 4.0operations 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:

- `/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 these criteria:
   - If a serial device is specified on the command line, or a previously processed configuration file, `pppd` uses the name of that device.
   - 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.
   - Otherwise, if `pppd` discovers that standard input is attached to a tty, then the name of the tty is used.
   - If `pppd` still cannot find a serial device, `pppd` 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 nonprivileged 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 configuration files are privileged regardless of their ownership:

- `/etc/ppp/options`
- `/etc/ppp/options.`*ttyname*
- `/etc/ppp/peers/`*peer-name*

The file `$HOME/.ppprc` is owned by the user. Options that are read from `$HOME/.ppprc` and from the command line are privileged only if the user who is invoking `pppd` is `root`.

Arguments that follow 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 who is 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.
`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.

Option	Status	Explanation
`bsdcomp`	Privileged if set in a privileged file or by a privileged user	The nonprivileged user cannot specify a code size that is larger than the privileged user has specified.
`deflate`	Privileged if set in a privileged file or by a privileged user	The nonprivileged user cannot specify a code size that is 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 nonprivileged user.
`init`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an nonprivileged user.
`pty`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an nonprivileged user.
`welcome`	Privileged if set in a privileged file or by a privileged user	Cannot be overridden by an nonprivileged user.
ttyname	Privileged if set in a privileged file	Opened with root permissions regardless of who invokes `pppd`.
	Not privileged if set in a nonprivileged 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 the following:

- **`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. 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 415. If a machine does not require global options, you can 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 Examples of the /etc/ppp/options Files

To find examples of the /etc/ppp/options file, refer to the following:

- For a dial-out machine, see "How to Define Communications Over the Serial Line" on page 415.
- For a dial-in server, see "How to Define Communications Over the Serial Line (Dial-in Server)" on page 422.
- For PAP support on a dial-in server, see "How to Add PAP Support to the PPP Configuration Files (Dial-in Server)" on page 435.
- For PAP support on a dial-out machine, see "How to Add PAP Support to the PPP Configuration Files (Dial-out Machine)" on page 438.
- For CHAP support on a dial-in server, see "How to Add CHAP Support to the PPP Configuration Files (Dial-in Server)" on page 442.

/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 that is read by pppd after parsing any existing /etc/ppp/options and existing $HOME/.ppprc files. 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. Your server might have only one serial line for PPP. Furthermore, the server requires few options. In this instance, you can specify any required 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 the following:

- 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 that are available for PPP in comparison to the number of potential callers. In this 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 system, you can create an `/etc/ppp/options.` *ttyname* file for the serial port that is connected to the modem, or choose not to use `/etc/ppp/options.` *ttyname*.

Note – Solaris PPP 4.0 does not require an `/etc/ppp/options.`*ttyname* file to work correctly. A dial-out machine might have only one serial line for PPP. Furthermore, the dial-out machine might require few options. You can specify any required 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. The template 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 that are 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 Examples of the `/etc/ppp/options.`*ttyname* Files

To find examples of the `/etc/ppp/options.`*ttyname* files, refer to the following:

- For a dial-out machine, see "How to Define Communications Over the Serial Line" on page 415.
- For a dial-in server, see "How to Define Communications Over the Serial Line (Dial-in Server)" on page 422.

Configuring User-Specific Options

This section contains detailed information about setting up users on the dial-in server.

Configuring `$HOME/.pppprc` on a Dial-in Server

The `$HOME/.pppprc` file is intended for users who are configuring preferred PPP options. As administrator, you can also configure `$HOME/.pppprc` for users.

The options in `$HOME/.pppprc` 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 `.pppprc` file is the second file that is checked by the `pppd` daemon.

See "Setting Up Users of the Dial-in Server" on page 420 for instructions about setting up `$HOME/.pppprc` on the dial-in server.

Configuring `$HOME/.pppprc` on a Dial-out Machine

The `$HOME/.pppprc` file is not needed on the dial-out machine for Solaris PPP 4.0 to work correctly. Additionally, you do not need to have a `$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 421 for configuration instructions for `.pppprc`.

Specifying Information for Communicating With the Dial-in Server

To communicate with a dial-in server, you need to gather information about the server. Then 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 users are not allowed to set.

For example, a nonprivileged user cannot override the noauth option if noauth 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 pppd 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 practice 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 by 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 that you might 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 by using the chat script /etc/ppp/myisp-chat.
`user myname`	Use this account name for the local machine. myname 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 that is 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 Examples of the /etc/ppp/peers/*peer-name* Files

To find examples of the /etc/ppp/peers/*peer-name* files, refer to the following:

- For a dial-out machine, see "How to Define the Connection With an Individual Peer" on page 417.
- For a local machine on a leased line, see "How to Configure a Machine on a Leased Line" on page 427.
- For support of PAP authentication on a dial-out machine, see "How to Add PAP Support to the PPP Configuration Files (Dial-out Machine)" on page 438.
- For support of CHAP authentication on a dial-out machine, see "How to Add CHAP Support to the PPP Configuration Files (Dial-out Machine)" on page 444.
- For support of PPPoE on a client system, see "Setting Up the PPPoE Client" on page 446.

Configuring Modem Speed for a Dial-up Link

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 the Solaris Management Console, as described in "Configuring Devices on the Dial-in Server" on page 419.

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 that is 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 the link includes an asynchronous interface 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. The dial-out machine and its remote peers exchange the strings 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 that is 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 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. Print the string.
`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.
`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.
`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 in this location the contents of the string that follows `-U`, which is specified on the command line. Usually, the string contains the password.
`% pppd`	Wait for the % shell prompt, and run the `pppd` command.

`/etc/ppp/myisp-chat.tmpl` Chat Script Template

This release 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 the template 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. Print the string.
`TIMEOUT 10`	Set initial timeout to 10 seconds. The modem's response should be immediate.
`"" "AT&F1"`	Reset the modem to factory defaults.
`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 drops. For &D2, DTR high-to-low transition causes the modem to go "on-hook" or 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 by 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 U.S. 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. Print the string.
`TIMEOUT 10`	Set initial timeout to 10 seconds. The modem's response should be immediate.
`"" 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 416.

```
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. Print the string.
`"" 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 by 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 the prompt 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. However, the PAP login 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.

Script Contents	Explanation
~ \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 is actually not part of PAP authentication.

The phrase ogin:--ogin: pppuser instructs the modem to send the user name pppuser in response to the login prompt 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 about creating PPP user accounts on a dial-in server, refer to "How to Configure Users of the Dial-in Server" on page 421.

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. Print the string.

Script Contents	Explanation
`"" AT&FB40S83.7=1&K44&J3X7S61.3=1S0=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 multilink, 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 "Chat-Script Field in `/etc/uucp/Systems` File" on page 535.

For More Chat Script Examples

A number of web sites offer sample chat scripts and assistance in creating the chat scripts. For example, see http://ppp.samba.org/ppp/index.html.

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. You might use the chat utility as the program to be invoked by `connect`. In this instance, if you store the chat script in an external file through 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. The `chat` utility is provided 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 runs 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.

Example 22–1 Inline Chat Script

You can place the entire chat script conversation on a single line, similar to the following:

```
connect 'chat "" "AT&F1" OK ATDT5551212 CONNECT "\c"'
```

The complete chat script follows the `chat` keyword. The script terminates with `"\c"'`. You use this form in any PPP configuration file or on the command line as an argument to `pppd`.

More 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 to document. You can add comments to the chat file by preceding the comments with the hash (#) sign.

The procedure "How to Create the Instructions for Calling a Peer" on page 416 shows the use of a chat script that is contained in an external file.

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 during link initiation, such as `stty` for parity settings, besides the commands that are contained in a traditional chat script.

This executable chat script logs in to an old-style UNIX system that requires 7 bits with even parity. The system then changes to 8 bits with 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.**

   `# chmod +x /etc/ppp/`*chatprogram*
3. **Invoke the chat program.**

   `connect /etc/ppp/`*chatprogram*

   Chat programs do not have to be located within the `/etc/ppp` file system. You can store chat programs 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 the authentication protocols.

Password Authentication Protocol (PAP)

PAP authentication is somewhat similar in operation to the UNIX `login` program, though PAP 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. PAP also uses `/etc/ppp/pap-secrets` for defining PAP security credentials. These credentials include a peer name, a "user name" in PAP parlance, and a password. PAP credentials also contain 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 in the PAM facility.

The `/etc/ppp/pap-secrets` file has the following syntax.

```
myclient ISP-server mypassword *
```

The parameters have the following meaning.

`myclient`	PAP user name of the caller. Often, this name is identical to the caller's UNIX user name, particularly if the dial-in server uses the `login` option of PAP.
`ISP-server`	Name of the remote machine, often a dial-in server.
`mypassword`	Caller's PAP password.
*	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*, that is, 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

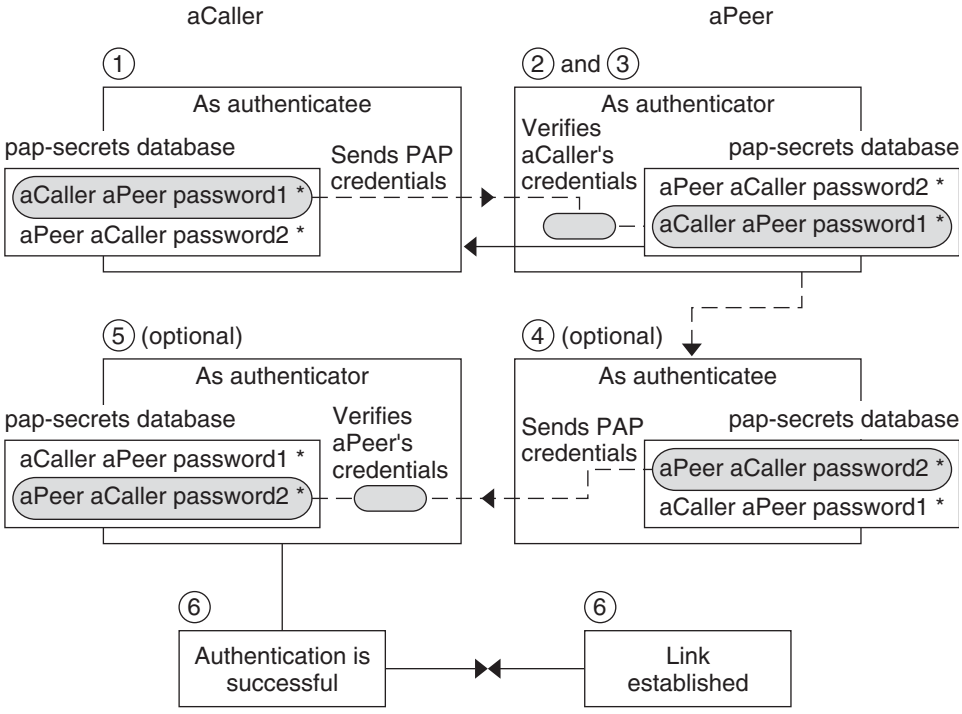
On the server (authenticator), the PAP password can be hidden by doing one of the following:

- 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 performed through the UNIX passwd database or the `pam`(3pam) mechanism.

What Happens During PAP Authentication

PAP authentication occurs in the following sequence.

FIGURE 22–1 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, the peer 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. (Optional) The original caller reads its own `/etc/ppp/pap-secrets` to verify the identity of the remote peer.

Note – If the original caller does require authentication credentials from the remote peer, Step 1 and Step 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 password database. The following shows the format of a `/etc/ppp/pap-secrets` file with the `login` option.

```
joe    *  ""  *
sally  *  ""  *
sue    *  ""  *
```

The parameters have the following meanings.

Caller	`joe`, `sally`, and `sue` are the names of the authorized callers.
Server	Asterisk (*), which indicates that any server name is valid. The `name` option is not required in the PPP configuration files.
Password	Double quotes, which indicate that any password is valid. If a password is in this column, then the password from the peer must match both the PAP password and the UNIX `passwd` database.
IP Addresses	Asterisk (*), which indicates that any IP address is allowed.

Challenge-Handshake Authentication Protocol (CHAP)

CHAP authentication uses the notion of the *challenge* and *response*, which means that 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." The CHAP secret is an arbitrary string that is known to both the caller and the 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.

```
myclient myserver secret5748 *
```

The parameters have the following meanings:

`myclient` CHAP user name of the caller. This name can be the same as or different from the caller's UNIX user name.

`myserver` Name of the remote machine, often a dial-in server.

`secret5748` Caller's CHAP secret.

Note – Unlike PAP passwords, CHAP secrets are never sent over the link. Rather, CHAP secrets are used when the local machines compute the response.

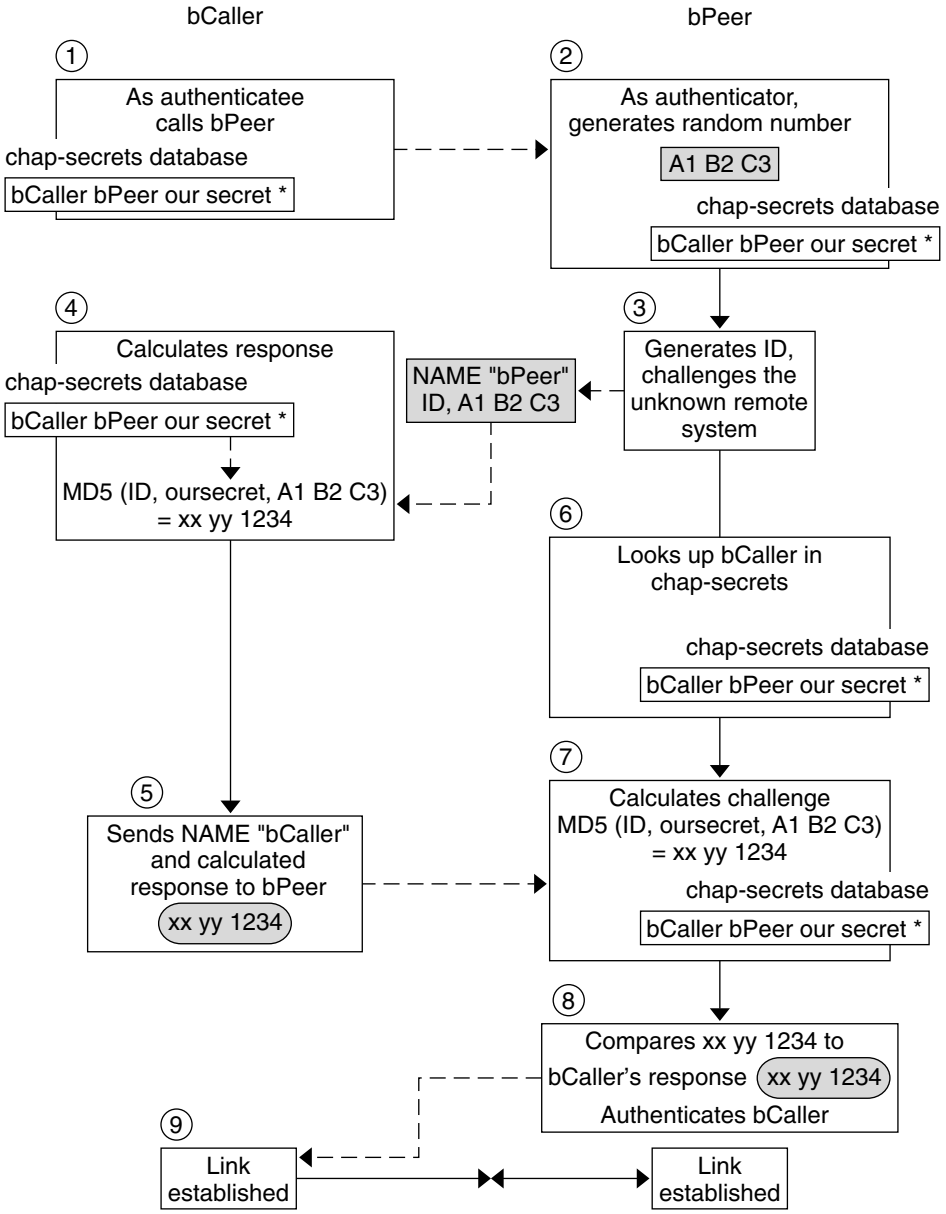
---

* 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 22–2 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 this data 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 of the IP address that is defined in `/etc/ppp/options.`*ttyname*. Dynamic addressing occurs on a per-serial port basis. When a call arrives over a serial line, the caller receives the IP address in the `/etc/ppp/options.`*ttyname* file for the call's serial interface.

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 following entry:

  ```
  :10.1.1.1
  ```

- For serial port `term/b`, create the file `/etc/ppp/options.term.b` with the following entry:

```
:10.1.1.2
```

- For serial port `term/c`, create the file `/etc/ppp/options.term.c` with the following entry:

```
:10.1.1.3
```

- For serial port `term/d`, create the file `/etc/ppp/options.term.d` with the following entry:

```
:10.1.1.4
```

With the previous 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 the following:

- 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 example of an `/etc/ppp/pap-secrets` file that defines static IP addresses.

```
joe    myserver  joepasswd   10.10.111.240
sally  myserver  sallypasswd 10.10.111.241
sue    myserver  suepasswd   10.10.111.242
```

Caller	`joe`, `sally`, and `sue` are the names of the authorized callers.
Server	`myserver` indicates the name of the server.
Password	`joepasswd`, `sallypasswd`, and `suepasswd` indicate the passwords for each caller.
IP Addresses	`10.10.111.240` and `10.10.111.241` and `10.10.111.242` are the IP addresses assigned to each caller.

Here is a example of an `/etc/ppp/chap-secrets` file that defines static IP addresses.

```
account1 myserver secret5748  10.10.111.244
account2 myserver secret91011 10.10.111.245
```

Caller	`account1` and `account2` indicate the names of the callers.
Server	`myserver` indicates the name of the server for each caller.

Password	`secret5748` and `secret91011` indicates the CHAP secret for each caller.
IP Addresses	`10.10.111.244` and `10.10.111.245` are the IP addresses for each caller.

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 following shows an example of this usage.

```
myclient ISP-server mypassword 10.10.111.240/28+
```

The plus sign (+) indicates that the unit number is added to the IP address. Note the following:

- 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 391.
- For tasks for setting up PPPoE tunnels, see Chapter 20, "Setting Up a PPPoE Tunnel (Tasks)."

This section contains detailed information about PPPoE commands and files, which is summarized in the next table.

TABLE 22–2 PPPoE Commands and Configuration Files

File or Command	Description	For Instructions
`/etc/ppp/pppoe`	A 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 502
`/etc/ppp/pppoe.`*device*	A file that contains characteristics of a particular interface that is used by PPPoE for a tunnel	"`/etc/ppp/pppoe.`*device* File" on page 504
`/etc/ppp/pppoe.if`	File that lists the Ethernet interface over which runs the tunnel that is set up by PPPoE	"`/etc/ppp/pppoe.if` File" on page 500

TABLE 22–2 PPPoE Commands and Configuration Files *(Continued)*

File or Command	Description	For Instructions
`/usr/sbin/sppptun`	Command for configuring the Ethernet interfaces that are involved in a PPPoE tunnel	"`/usr/sbin/sppptun` Command" on page 500
`/usr/lib/inet/pppoed`	Command and options for using PPPoE to set up a tunnel	"`/usr/lib/inet/pppoed` Daemon" on page 502

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 PPPoE 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 when the interfaces that are listed are plumbed for use in PPPoE tunnels.

You need to create explicitly `/etc/ppp/pppoe.if`. Type the name of one interface to be configured for PPPoE on each line.

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. 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 that is 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 that you issue the `sppptun` command, the discovery protocol `pppoed` is plumbed on the interface. The second time that you run `sppptun`, the session protocol `pppoe` is plumbed. `sppptun` prints the name of the interface that was just plumbed. You use this name to unplumb the interface, when necessary.

For more information, refer to the `sppptun(1M)` man page.

Examples of `sppptun` Commands for Administering Interfaces

The following example shows how to manually plumb an interface for PPPoE by using `/usr/sbin/sppptun`.

```
# /usr/sbin/sppptun plumb pppoed hme0
hme0:pppoed
# /dev/sppptun plumb pppoe hme0
 hme0:pppoe
```

This example shows how to list the interfaces on an access server that was plumbed for PPPoE.

```
# /usr/sbin/sppptun query
hme0:pppoe
hme0:pppoed
hme1:pppoe
hme1:pppoed
hme2:pppoe
hme2:pppoed
```

This example shows how to unplumb an interface.

```
# 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 PPPoE. The PPPoE access server and client do function in the traditional client-server relationship. This relationship is similar to the relationship of the dial-out machine and dial-in server on a dial-up link. One PPPoE system initiates communications and one PPPoE system answers. By contrast, the PPP protocol has no notion of the client-server relationship. PPP considers both systems equal peers.

The commands and files that set up a PPPoE access server include the following:

/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, 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 file 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
      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 that are 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 that are 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 that is 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 resemble the following.

EXAMPLE 22–2 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
   pppd "debug name internet-server"
```

In this file, the following values apply.

`hme1,hme2,hme3`	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 interpret `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 that is 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. `/etc/ppp/pppoe.`*device* also includes 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 have precedence for that interface over the global parameters that are defined in `/etc/ppp/pppoe`.

The basic syntax of `/etc/ppp/pppoe.`*device* is as follows:

```
service service-name
      service-specific-options
service another-service-name
       service-specific-options
```

The only difference between this syntax and the syntax of `/etc/ppp/pppoe` is that you cannot use the `device` option that is shown in "`/etc/ppp/pppoe` File" on page 502.

`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. The server is 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 the clients 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 by using static IP addresses.

EXAMPLE 22–3 /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 that are 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 force the server to 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 the service section 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 force the server to 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 this access server scenario just discussed, you might set up the following /etc/ppp/options file.

EXAMPLE 22–4 /etc/ppp/options File for an Access Server

```
     auth
     proxyarp
```

EXAMPLE 22–4 `/etc/ppp/options` File for an Access Server *(Continued)*

```
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 found by the `/usr/bin/hostname` command. The `name` option in the previous example changes the server's name to `no-service`. The name `no-service` is 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 no secrets can be found for the client with a server name of `no-service`.

The access server scenario uses the following `/etc/hosts` file.

EXAMPLE 22–5 `/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 22–6 `/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 that is used for CHAP authentication. Note that only clients `joe` and `mary` are listed in the file.

EXAMPLE 22–7 `/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 the tunnel 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 500
- "`/usr/lib/inet/pppoec` Utility" on page 507
- "`pppoe.so` Shared Object" on page 507
- "`/etc/ppp/peers/`*peer-name* File" on page 480
- "`/etc/ppp/options` Configuration File" on page 475

`/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 `chat` utility. You do not invoke `pppoec` directly. Rather, you start `/usr/lib/inet/pppoec` as an argument to the `connect` option of `pppd`.

`pppoe.so` Shared Object

`pppoe.so` is the PPPoE shared object that must be loaded by PPPoE to provide PPPoE capability to access servers and clients. The `pppoe.so` 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. An `/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. `connect` then 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 447.

EXAMPLE 22–8 `/etc/ppp/peers/`*peer-name* to Define a Remote Access Server

```
# cat /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 that are 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.
`noccp`	Turns off CCP compression on the link. **Note –** 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, which is 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 is true for a PPPoE client.

◆ ◆ ◆ CHAPTER 23

Migrating From Asynchronous Solaris PPP to Solaris PPP 4.0 (Tasks)

Earlier versions of the Solaris OS included a 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:

- "Before Converting `asppp` Files" on page 509
- "Running the `asppp2pppd` Conversion Script (Tasks)" on page 512

The chapter 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 380.

Before Converting `asppp` Files

You can use the conversion script `/usr/sbin/asppp2pppd` to convert the files that compose 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 `asppp`, see the *Solaris 8 System Administration Collection, Volume 3*, available from `http://docs.sun.com`.

Example of the `/etc/asppp.cf` Configuration File

The procedure that is shown in "How to Convert From `asppp` to Solaris PPP 4.0" on page 512 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 two minutes of inactivity
`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 of the `/etc/uucp/Systems` File

The procedure that is shown in "How to Convert From asppp to Solaris PPP 4.0" on page 512 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`	Uses `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 of the /etc/uucp/Devices File

The procedure that is shown in "How to Convert From asppp to Solaris PPP 4.0" on page 512 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
```

The file supports any Hayes modem that is connected to serial port cua/b.

Example of the /etc/uucp/Dialers File

The procedure that is shown in "How to Convert From asppp to Solaris PPP 4.0" on page 512 uses the following /etc/uucp/Dialers file.

```
#
#       <Much information about modems supported by Oracle 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 Oraccle 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 (Tasks)

The `/usr/sbin/asppp2pppd` script copies the PPP information in `/etc/asppp.cf` and PPP-related UUCP files to appropriate locations in the Solaris PPP 4.0 files.

Task Prerequisites

Before doing the next task, you should have done the following:

- Installed the Solaris release 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) is from the /etc/uucp/Devices file. The link speed, idle time, authentication information, and peer names are 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 that are created for dial-out machine mojave.

```
File number (1 .. 4):  3
#lock
noauth
```

The information in /etc/ppp/options is 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 mojave and Pgobi.

6 Type 9 to save the created files. Then exit the conversion script.

◆ ◆ ◆ CHAPTER 24

UUCP (Overview)

This chapter introduces the UNIX-to-UNIX Copy Program (UUCP) and its daemons. The following topics are covered:

UUCP enables computers to transfer files and exchange mail with each other. The program also enables computers to participate in large networks such as Usenet.

The Solaris OS 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 compose the system, of which the program `uucp` is only a part. The UUCP utilities range from those utilities that are used to copy files between computers (`uucp` and `uuto`) to those utilities that are 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 when two computers communicate regularly and are physically close, such as within 50 feet of each other. You can use a limited-distance modem to increase this distance somewhat.

Telephone lines — By 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 that is requested by UUCP. The recipient machine must have a modem that is capable of answering incoming calls.

Network — UUCP can also communicate over a network that runs TCP/IP or another protocol family. After your computer has been established as a host on a network, your computer can contact any other host that is 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 accompanied 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 the UUCP software by 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 that is used for the link, establishes the link to the remote computer, and performs the required login sequence and permission checks. Also, `uucico` transfers data files, execute files, and results from logs, 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. This daemon 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. `uuxqt` 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 a password 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. The user ID 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. `uucleanup` 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. `Uutry` invokes the `uucico` daemon to establish a communication link between your machine and the remote computer that you specify. See the `Uutry`(1M) man page for details.

`uucheck` Checks for the presence of UUCP directories, programs, and support files. `uucheck` 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 machines 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 machine. `uucp` 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 the file 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 by using `uuto`. See the `uuto`(1C) man page.
`uux`	Creates the work, data, and execute files that are 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`). `uustat` 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 that compose the UUCP database. These files are in the `/etc/uucp` directory. You need to edit these files to set up UUCP or `asppp` on your machine. The files include the following:

`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, `Dialcodes` can be used by `asppp` as well as UUCP.
`Dialers`	Contains character strings that are required to negotiate with modems to establish connections with remote computers. `Dialers` is used by `asppp` as well as UUCP.
`Grades`	Defines job grades, and the permissions that are associated with each job grade, which users can specify to queue jobs to a remote computer.
`Limits`	Defines the maximum number of simultaneous `uucicos`, `uuxqts`, and `uuscheds` that are permitted on your machine.

`Permissions`	Defines the level of access that is 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 that is needed by the `uucico` daemon, `cu`, and `asppp` to establish a link to a remote computer. This information includes the following: ■ Name of the remote host ■ Name of the connecting device associated with the remote host ■ Time when the host can be reached ■ Telephone number ■ Login ID ■ 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 that are shown in "UUCP Database Files" on page 518. 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 minimum 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 are not 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 each other, you should understand all their contents before you make any changes. A change to an entry in one file might require a change to a related entry in another file. The remaining files that are listed in "UUCP Database Files" on page 518 are not as critically intertwined.

Note – `asppp` uses only the files that are described in this section. `asppp` does not use the other UUCP database files.

◆ ◆ ◆ CHAPTER 25

Administering UUCP (Tasks)

This chapter explains how to start UUCP operations after you have modified the database file that is relevant to your machines. The chapter contains procedures and troubleshooting information for setting up and maintaining UUCP on machines that run the Solaris OS, such as the following:

UUCP Administration (Task Map)

The following table provides pointers to the procedures that are covered in this chapter, in addition to a short description of each procedure.

TABLE 25–1 Task Map for UUCP Administration

Task	Description	For Instructions
Allow remote machines to have access to your system	Edit the `/etc/passwd` file to add entries to identify the machines that are permitted to access your system.	"How to Add UUCP Logins" on page 522
Start UUCP	Use the supplied shell scripts to start UUCP.	"How to Start UUCP" on page 523
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 525
Troubleshoot some common UUCP problems	Use diagnostic steps to check for faulty modems or ACUs.	"How to Check for Faulty Modems or ACUs" on page 527

TABLE 25–1 Task Map for UUCP Administration *(Continued)*

Task	Description	For Instructions
	Use diagnostic steps to debug transmissions.	"How to Debug Transmissions" on page 528

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. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit the `/etc/passwd` file and add the entry to identify the machine that is permitted to access your system.**

   A typical entry that you might put into the `/etc/passwd` file for a remote machine that is 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. Otherwise, you might have to truncate or abbreviate the name.

   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.

3. **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 includes four shell scripts that poll remote machines, reschedule transmissions, and clean up old log files and unsuccessful transmissions. The scripts are as follows:

- `uudemon.poll`
- `uudemon.hour`
- `uudemon.admin`
- `uudemon.cleanup`

These shell scripts should execute regularly to ensure that UUCP runs 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, the file 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit the `/usr/lib/uucp/uudemon.crontab` file and change entries as required.**

3. **Activate the `uudemon.crontab` file by issuing the following command:**

   ```
   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. *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 in place when uudemon.hour is called.

uudemon.hour Shell Script

The default uudemon.hour shell script does the following:

- Calls the uusched program to search the spool directories for work files (C.) that have not been processed. The script then 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 when they were transferred.

By default, uudemon.hour runs twice an hour. You might want uudemon.hour 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 that are listed in the lock files (/var/spool/locks).
- Sends resulting status information to the uucp administrative login by using mail.

uudemon.cleanup Shell Script

The default uudemon.cleanup shell script does the following:

- Collects log files for individual machines from the /var/uucp/.Log directory, merges these files, and places the files 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 that was 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 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Edit the `/etc/uucp/Systems` file to ensure that the entries have the following fields:**

System-Name Time TCP Port networkname Standard-Login-Chat

A typical entry would resemble the following:

```
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 capability 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, a completely different machine could easily run UUCP and have the TCP/IP host name of `rochester`.

The Port field in the `Systems` file should have the entry `-`. This syntax is equivalent to listing the entry as `uucp`. In almost every situation, 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 `in.uucpd` 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`.

4 **Verify that UUCP is enabled.**

```
# svcs network/uucp
```

The UUCP service is managed by the Service Management Facility. To query the status of this service, you can use the `svcs` command. For an overview of the Service Management Facility, refer to Chapter 18, "Managing Services (Overview)," in *System Administration Guide: Basic Administration*.

5 **(Optional) If necessary, enable UUCP by typing the following:**

```
# inetadm -e network/uucp
```

UUCP Security and Maintenance

After you have set up UUCP, maintenance is straightforward. This section explains ongoing UUCP tasks that relate 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 the following:

- Ways that the remote machine can receive files from your machine
- Directories for which the remote machine has read and write permission
- Commands that the remote machine can use for remote execution

A typical `Permissions` entry follows:

```
MACHINE=datsun LOGNAME=Udatsun VALIDATE=datsun
COMMANDS=rmail REQUEST=yes SENDFILES=yes
```

This entry allows files to be sent and be received to and from the "normal" UUCP directories, not from anywhere in the system. The entry also causes the UUCP user name to be validated at login time.

Regular UUCP Maintenance

UUCP does not require much maintenance. However, you must ensure that the `crontab` file is in place, as described in the section "How to Start UUCP" on page 523. Your concern should be the growth of mail files and the public directory.

Email for UUCP

All email messages that are generated by the UUCP programs and scripts are sent 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 problem, create an alias in `/etc/mail/aliases` and redirect that email either to `root` or to yourself and others who are 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 the directory. However, the directory's sticky bit is set, so the directory's 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 ensure that you remove files from it periodically.

If this maintenance 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 risk 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. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Obtain counts and reasons for contact failure by running the following command:**

   ```
   # uustat -q
   ```

3. **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 -lline
   ```

   line is `/dev/cua/a`.

How to Debug Transmissions

If you cannot contact a particular machine, you can check communications to that machine with `Uutry` and `uucp`.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Try to make contact:

```
# /usr/lib/uucp/Uutry -r machine
```

Replace *machine* with the host name of the machine you are unable to contact. This command does the following:

- Starts the transfer daemon (`uucico`) with debugging. You can get more debugging information if you are `root`.
- Directs the debugging output to `/tmp/`*machine*.
- Prints the debugging output to your terminal by issuing the following command:

  ```
  # tail -f
  ```

  Press Control-C to end output. You can copy the output from `/tmp/`*machine* if you want to save the output.

3 If `Uutry` does not isolate the problem, try to queue a job:

```
# uucp -r file machine\!/dir/file
```

file	Use the name of the file that you want to transfer.
machine	Use the name of the machine that you want to copy to.
/dir/file	Specify the location of the file for the other machine.

4 Issue the following command:

```
# Uutry
```

If you still cannot solve the problem, you might need to call your local support representative. Save the debugging output, which can help diagnose the problem.

Note – You might also decrease or increase the level of debugging that is provided by `Uutry` through the `-x` *n* option. *n* indicates the debug level. The default debug level for `Uutry` is 5.

Debug level 3 provides basic information about when and how the connection is established, but not much information about the transmission. Debug level 9, however, 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 other site's administrator and decide when to change the level.

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 information that might be out of date for a machine is the following:

- 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 that your computer attempts to communicate with. These files contain status information about attempted communication and whether the communication was successful.

Checking Basic Information

Several commands are available for checking basic networking information:

- Use the `uuname` command to list those machines that 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 that are needed by `uucp`. This command also checks the `Permissions` file and displays information about the permissions that you have set up.

◆ ◆ ◆ CHAPTER 26

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 that is needed by the `uucico` daemon to establish a communication link to a remote computer. `/etc/uucp/Systems` is the first file that 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.

By using the `Sysfiles` file, you can define several files to be used as `Systems` files. See "UUCP `/etc/uucp/Sysfiles` File" on page 549 for a description of `Sysfiles`.

The following is the syntax for an entry in the `Systems` file:

```
System-Name    Time    Type    Speed    Phone    Chat Script
```

See the following example of an entry in the Systems file.

EXAMPLE 26–1 Entry in /etc/uucp/Systems

```
Arabian     Any  ACUEC 38400 111222  ogin: Puucp ssword:beledi
```

Arabian	Entry for the System-Name field. For more information, see "System-Name Field in /etc/uucp/Systems File" on page 532.
Any	Entry for the Time field. For more information, see "Time Field in /etc/uucp/Systems File" on page 532.
ACUEC	Entry for the Type field. For more information, see "Type Field in /etc/uucp/Systems File" on page 533.
38400	Entry for the Speed field. For more information, see "Speed Field in /etc/uucp/Systems File" on page 534.
111222	Entry for the Phone field. For more information, see "Phone Field in /etc/uucp/Systems File" on page 534.
ogin: Puucp ssword:beledi	Entry for the Chat Script field. For more information, see "Chat-Script Field in /etc/uucp/Systems File" on page 535.

System-Name Field in /etc/uucp/Systems File

This field contains the node name of the remote computer. On TCP/IP networks, this name can be the machine's host name or a name that is created specifically for UUCP communications through the /etc/uucp/Sysname file. See "UUCP /etc/uucp/Systems File" on page 531. In Example 26–1, the System-Name field contains an entry for remote host Arabian.

Time Field in /etc/uucp/Systems File

This field specifies the day of week and time of day when the remote computer can be called. The format of the Time field follows:

```
daytime[;retry]
```

day Portion of Time Field

The *day* portion can be a list that contains some of the following entries.

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*.

time Portion of Time Field

Example 26–1 shows `Any` in the Time field, which indicates that host `Arabian` can be called at any time.

The *time* portion should be a range of times that are specified in 24-hour notation, for example, `0800-1230` for 8:30 a.m. to 12:30 p.m. 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 a.m. and 8 a.m.

retry Portion of Time Field

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. This means 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 no response occurs, 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, the value specified is always the retry time. Otherwise, the back-off algorithm is used.

Type Field in `/etc/uucp/Systems` File

This field contains the device type that should be used to establish the communication link to the remote computer. The keyword that is used in this field is matched against the first field of `Devices` file entries.

EXAMPLE 26–2 Keyword With the Type Field

```
Arabian    Any    ACUEC, g    38400    1112222    ogin: Puucp ssword:beledi
```

You can define the protocol that is used to contact the system by adding the protocol to the Type field. The previous example shows how to attach the protocol g to the device type ACUEC. For information about protocols, see "Protocol Definitions in /etc/uucp/Devices File" on page 543.

Speed Field in /etc/uucp/Systems File

This field, also known as the Class field, specifies the transfer speed of the device that is used in establishing the communication link. The UUCP speed field can contain a letter and speed, such as C1200 or D1200, to differentiate between classes of dialers. Refer to "Class Field in the /etc/uucp/Devices File" on page 540.

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 26–3 Entry in Speed Field

```
eagle    Any    ACU, g    D1200    NY3251    ogin: nuucp ssword:Oakgrass
```

If information is not required for this field, use a dash (-) as a placeholder for the field.

Phone Field in /etc/uucp/Systems File

This field enables you to specify the telephone number, known as a *token*, of the remote computer for automatic dialers, which are known as *port selectors*. The telephone number consists of an optional alphabetic abbreviation and a numeric part. If an abbreviation is used, the abbreviation must be listed in the Dialcodes file.

EXAMPLE 26–4 Entry in the Phone Field

```
nubian    Any    ACU       2400    NY555-1212    ogin: Puucp ssword:Passuan
eagle     Any    ACU, g    D1200   NY=3251       ogin: nuucp ssword:Oakgrass
```

In the Phone field, an equal sign (=) instructs 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 that are 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 to the switch. In this way, the port selector knows the remote machine with which your host wants to communicate, 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 by using the Dialcodes file.

Chat-Script Field in /etc/uucp/Systems File

This field, also known as the Login field, contains a string of characters that is 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 following format:

expect send [expect send] ....

expect represents the string that the local host expects to receive from the remote host to initiate conversation. *send* is the string that the local host sends after the local host 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 the following:

- Login prompt that the local host expects to receive 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 receive from the remote machine
- Password that the local host sends to the remote machine

The *expect* field can be composed of subfields of the following form:

expect[-send-expect]...

The *-send* is sent if the prior *expect* is not successfully read. The *-expect* that follows the *-send* is the next expected string.

For example, with strings `login--login`, the UUCP on the local host expects `login`. If UUCP receives `login` from the remote machine, UUCP goes to the next field. If UUCP does not receive `login`, UUCP sends a carriage return, then looks for `login` again. If the local computer initially does not expect any characters, use the characters "", for NULL string, in the *expect* field. All *send* fields are sent with a carriage return appended unless the *send* string is terminated with a `\c`.

The following is an example of a `Systems` file entry that uses an *expect-send* string:

```
sonora Any ACUEC 9600 2223333 "" \r \r ogin:-BREAK-ogin: Puucpx ssword:xyzzy
```

This example instructs 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 receive `ogin:`, send the login name `Puucpx`. When you receive `ssword:` (for `Password:`), send the password `xyzzy`.

The following table lists some useful escape characters.

TABLE 26–1 Escape Characters Used in the Chat-Script Field of the Systems File

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 forward, whenever a character is transmitted, UUCP waits for the character to be received before continuing its checks.
\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 that is 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, the dial-in server disconnects the link to the remote machine and then calls back the remote machine. 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 receive the characters, INITIATED, from the dial-in server. After the characters, INITIATED, have been matched, the dialing facility flushes any subsequent characters that the dialing facility 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.

A string of characters does not need to directly precede or follow the \H, as shown in the previous sample string.

Hardware Flow Control in /etc/uucp/Systems File

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 enables hardware flow control in a Systems file entry:

```
unix Any ACU 2400 12015551212 "" \r ogin: Puucp ssword:Passuan "" \ STTY=crtscts
```

This pseudo-send string can also be used in entries in the Dialers file.

Setting Parity in /etc/uucp/Systems File

In some situations, 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. See this expect-send couplet in the following example:

```
unix Any ACU 2400 12015551212 "" P_ZERO "" \r ogin: Puucp ssword:Passuan
```

The following are parity couplets that can follow the expect-send couplet, "" P_ZERO:

"" P_EVEN Sets the parity to even, which is the default

"" P_ODD Sets the parity to odd

"" P_ONE Sets the parity bit to 1

These parity couplets can be inserted anywhere in the chat script. The parity couplets apply to all information in the chat script that follows "" P_ZERO, the expect-send couplet. A parity couplet can also be used in entries in the Dialers file. The following example includes the parity couplet, "" P_ONE:

```
unix Any ACU 2400 12015551212 "" P_ZERO "" P_ONE "" \r ogin: Puucp ssword:Passuan
```

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 include high-speed modems), direct links, and network connections.

An entry in the /etc/uucp/Devices file has the following syntax:

```
Type   Line   Line2   Class   Dialer-Token-Pairs
```

The following is an entry in the Devices file for a U.S. Robotics V.32bis modem that is attached to port A and is running at 38,400 bps.

```
ACUEC   cua/a   -   38400   usrv32bis-ec
```

ACUEC	Entry in the Type field. For more information, see "Type Field in /etc/uucp/Devices File" on page 538.
cua/a	Entry in the Line field. For more information, see "Line Field in the /etc/uucp/Devices File" on page 540.
-	Entry in the Line2 field. For more information, see "Line2 Field in the /etc/uucp/Devices File" on page 540.
38400	Entry in the Class field. For more information, see "Class Field in the /etc/uucp/Devices File" on page 540.
usrv32bis-ec	Entry in the Dialer-Token-Pairs field. For more information, see "Dialer-Token-Pairs Field in the /etc/uucp/Devices File" on page 541.

Each field is described in the next section.

Type Field in /etc/uucp/Devices File

This field describes the type of link that the device establishes. The UUCP Type field can contain one of the keywords that is 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. Create 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

The port selector is a variable that is replaced in the Type field by the name of a port selector. Port selectors are devices that are attached to a network that prompts 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 544 for more information.

System-Name Variable

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 with an entry in /etc/uucp/Systems for the computer *System-Name*.

Type Fields in Devices File and Systems File

Example 26–5 shows a comparison of the fields in /etc/uucp/Devices and the fields in /etc/uucp/Systems. The keyword that is 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 instance a V.32bis modem. This value is matched against the Type field in the Systems file, which also contains the entry ACUEC. See "UUCP /etc/uucp/Systems File" on page 531 for more information.

EXAMPLE 26–5 Comparison of Type Fields in Devices file and Systems File

The following is an example of an entry in the Devices file.

```
ACUEC cua/a - 38400 usrv32bis-ec
```

The following is an example of an entry in the Systems file.

```
Arabian Any ACUEC 38400 111222 ogin: Puucp ssword:beledi
```

Line Field in the /etc/uucp/Devices File

This field contains the device name of the line (known as port) that is associated with the Devices entry. If the modem that is associated with a particular entry were attached to the /dev/cua/a device (serial port A), the name that is 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
```

Line2 Field in the /etc/uucp/Devices File

This field is a placeholder. Always use a hyphen (-) here. 801–type dialers, which are not supported in the Solaris OS, use the Line2 field. Non-801 dialers do not normally use this configuration, but still require a hyphen in this field.

Class Field in the /etc/uucp/Devices File

The Class field contains the speed of the device, if the keyword ACU or Direct is used in the Type field. However, the Class field can contain a letter and a speed, such as C1200 or D1200, to differentiate between classes of dialers, such as Centrex or Dimension PBX.

This differentiation 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 network handles the external communications. In such a situation, you must distinguish which line or lines should be used for internal communications and which should be used for external communications.

The keyword that is used in the Class field of the Devices file is matched against the Speed field of the Systems file.

EXAMPLE 26–6 Class Field in the Devices file

```
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 that is 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.

Dialer-Token-Pairs Field in the /etc/uucp/Devices File

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 the *dialer* portion is not present, it is taken from a related entry in the `Systems` file. The *token* portion can be supplied immediately after the dialer portion.

The last dialer-token pair might not be present, depending on the associated dialer. In most situations, 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 list shows the special dialer types.

`TCP`	TCP/IP network
`TLI`	Transport Level Interface Network (without STREAMS)
`TLIS`	Transport Level Interface Network (with STREAMS)

See "Protocol Definitions in /etc/uucp/Devices File" on page 543 for more information.

Structure of the Dialer-Token-Pairs Field in the /etc/uucp/Devices File

The DTP field can be structured four different ways, depending on the device that is associated with the entry.

See the first way that the DTP field can be structured:

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 26–7 Dialers Field for Directly 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 instance, the phone number) is taken from the Phone field of a `Systems` file entry. (`\T` is implied, as described in Example 26–9.)

See the second and third ways that the DTP field can be structured:

- **Direct link** – For a direct link to a particular computer, the DTP field of the associated entry would contain the keyword `direct`. This condition is true for both types of direct-link entries, `Direct` and *System-Name*. Refer to "Type Field in /etc/uucp/Devices File" on page 538.
- **Computers on the same port selector** – If a computer with which you intend 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 26–8 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 designation 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 531 for details. This type of DTP contains an escape character (`\D`), which ensures that the content of the Phone field is not interpreted as a valid entry in the `Dialcodes` file.

See the fourth way that the DTP field can be structured:

Modems that are 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 (the fifth and seventh fields of the entry) is used to match entries in the `Dialers` file, as follows.

EXAMPLE 26–9 UUCP Dialers Field for Modems Connected to Port Selector

```
develcon ""     ""    \pr\ps\c  est:\007    \E\D\e      \007
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 a 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. 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 by using the `/etc/uucp/Dialcodes` file. This escape character is normally placed in the `/etc/uucp/Dialers` file for each caller script that is associated with a modem, such as Hayes, and U.S. Robotics. Therefore, the translation does not occur until the caller script is accessed.
- `\D` – Indicates that the Phone (*token*) field should not be translated by 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 that are associated with network switches `develcon` and `micom`.

Protocol Definitions in `/etc/uucp/Devices` File

You can define the protocol to use with each device in `/etc/uucp/Devices`. This specification 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 531 for details. If you do specify the protocol, you must use the following 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 26–2 Protocols Used in `/etc/uucp/Devices`

Protocol	Description
`t`	This protocol is commonly used for transmissions over TCP/IP and other reliable connections. `t` assumes error-free transmissions.
`g`	This protocol is UUCP's native protocol. `g` 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, such as TCP/IP.
`f`	This protocol is used for transmission over X.25 connections. `f` 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 enacted over a whole file only. If a transport fails, the receiver can request retransmission or retransmissions.

Here is an example that shows a protocol designation for a device entry:

```
TCP,te - - Any TCP -
```

This example indicates that, for device TCP, you should try to use the t protocol. If the other end of the transmission 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 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 occur on a line before the line can be made available for transferring data. This conversation, known as a chat script, is usually a sequence of ASCII strings that is transmitted and is expected. A chat script is often used to dial a phone number.

As shown in the examples in "UUCP /etc/uucp/Devices File" on page 538, the fifth field in a Devices file entry is an index into the Dialers file or a special dialer type, such as 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 syntax:

```
dialer   substitutions   expect-send
```

The following example shows the entry for a U.S. Robotics V.32bis modem.

EXAMPLE 26–10 Entry in /etc/uucp/Dialers File

```
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
```

usrv32bis-e
: Entry in the Dialer field. The Dialer field matches the fifth and additional odd-numbered fields in the Devices file.

`=,-, ""`
Entry in the Substitutions field. The Substitutions field is a translation string. The first of each pair of characters is mapped to the second character in the pair. This mapping is usually used to translate = and - into whatever the dialer requires for "wait for dial tone" and "pause."

`dA\pT&FE1V1X1Q0S2=255S12=255&A1&H1&M5&B2&W\r\c OK\r`
Entry in Expect-Send field. The Expect-Send fields are character strings.

`\EATDT\T\r\c CONNECT\s14400/ARQ STTY=crtscts`
More of the Expect-Send field.

The following example shows sample entries in the `Dialers` file, as distributed when you install UUCP as part of the Solaris installation program.

EXAMPLE 26–11 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
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
```

EXAMPLE 26–11 Excerpts From /etc/uucp/Dialers *(Continued)*

```
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 that are commonly used in the send strings in the Dialers file.

TABLE 26–3 Backslash Characters for /etc/uucp/Dialers

Character	Description
\b	Sends or expects a backspace character.
\c	No newline or carriage return.
\d	Delays for approximately 2 seconds.
\D	Phone number or token without Dialcodes translation.
\e	Disables echo checking.
\E	Enables echo checking for slow devices.
\K	Inserts a Break character.
\n	Sends newline.
\nnn	Sends octal number. Additional escape characters that can be used are listed in the section "UUCP /etc/uucp/Systems File" on page 531.
\N	Sends or expects a NULL character (ASCII NUL).
\p	Pauses for 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 that is given by the remainder of the line works as listed:

- "" – Waits for nothing, which means proceed to the next step.
- \d – Delays 2 seconds, then sends 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) 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 – \E enables echo checking. From this point forward, whenever a character is transmitted, UUCP waits for the character to be received before proceeding. Then, UUCP sends the phone number. The \T means to take the phone number that is passed as an argument. The \T applies the Dialcodes translation and the modem function translation that is specified by field 2 of this entry. Then \T sends a P and a carriage return.
- > – Waits for a >.
- 9\c – Sends a 9 without a newline.
- OK – Waits for the string OK.

Enabling Hardware Flow Control in the /etc/uucp/Dialers File

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. 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.

Setting Parity in the /etc/uucp/Dialers File

In some situations, 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
```

The following are parity couplets that can follow the expect-send couplet:

"" P_EVEN — Sets the parity to even, which is the default

"" P_ODD — Sets the parity to odd

"" P_ONE — Sets the parity to one

This pseudo-send string can also be used in entries in the Systems file.

Other Basic UUCP Configuration Files

You can use files in this section 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 file to provide additional information about a basic phone number that is used by several systems at the same site.

Each entry has the following syntax:

```
Abbreviation   Dial-Sequence
```

Abbreviation — This field provides the abbreviation that is used in the Phone field of the Systems file.

Dial-Sequence — This field provides the dial sequence that is passed to the dialer when that particular Systems file entry is accessed.

Compare the fields in the two files. The following are the fields in the Dialcodes file.

```
Abbreviation   Dial-Sequence
```

The following are the fields in the Systems file.

```
System-Name   Time   Type   Speed   Phone   Chat Script
```

The following table contains sample content for the fields in a `Dialcodes` file.

TABLE 26–4 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 following entry:

```
NY5551212
```

When `uucico` reads `NY` in the `Systems` file, `uucico` searches the `Dialcodes` file for `NY` and obtains the dialing sequence `1=212`. `1=212` is the dialing sequence that is needed for any phone call to New York City. This sequence 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 such as `jt7867` in the `Systems` file. When `uucico` reads the entry that contains `jt7867` in the `Systems` file, `uucico` 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 about `cu`, see the cu(1C) man page. You can use `Sysfiles` for the following:

- 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 the file more convenient to split into several smaller files.

The syntax of the `Sysfiles` file is as follows:

`service=`*w* `systems=`*x*`:`*x* `dialers=`*y*`:`*y* `devices=`*z*`:`*z*

w Represents `uucico`, `cu`, or both commands 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 that it is presented

y Represents one or more files to be used as the `Dialers` file

z Represents 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 being called does not have an entry in that file, or if the entries in the file fail, then both commands check /etc/uucp/Local_Systems.

As specified in 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 by using the uuname command or the cu list by using the uuname -C command. The following is another example of the file, which shows that the alternate files are consulted first and the default files are consulted if necessary:

```
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*. The node name 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 that contains the UUCP node name for your system.

UUCP /etc/uucp/Permissions File

The /etc/uucp/Permissions file specifies the permissions that remote computers have for login, file access, and command execution. Some options restrict the remote computer's ability

to request files and its ability to receive files that are 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 composed of options that are delimited by a 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 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 as follows:

- `LOGNAME` – Specifies the permissions that become effective when a remote computer logs in to (calls) your computer.

  Note – When a remote machine calls you, its identity is questionable unless the remote machine has a unique login and verifiable password.

- `MACHINE` – Specifies permissions that become effective when your computer logs in to (calls) a remote computer.

`LOGNAME` entries contain a `LOGNAME` option. `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 that is granted to remote computers, you should consider the following:

- All login IDs that are used by remote computers to log in for UUCP communications must appear in one and only one `LOGNAME` entry.
- Any site that is called with 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 that are 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 be 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. REQUEST=no, the default value, is used if the REQUEST option is not specified. The REQUEST option can appear in either a LOGNAME entry, so that the remote computer calls you, or a MACHINE entry, so that you call the remote computer.

UUCP SENDFILES Option

When a remote computer calls your computer and completes its work, the remote computer can attempt to retrieve the work that your computer has queued for it. The SENDFILES option specifies whether your computer can send the work that is queued for the remote computer.

The string SENDFILES=yes specifies that your computer can send the work that is queued for the remote computer if 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, but it is not allowed to initiate a call to this particular remote computer. See "UUCP /etc/uucp/Systems File" on page 531 for more information.

The string SENDFILES=call specifies that files that are 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 sent to remote computers. If the option is used with a MACHINE entry, the option 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 can set the

MYNAME option of the Permissions file. Suppose that you want your organization to be known as widget. If all your modems are connected to a machine with the host name gadget, you can have an entry in gadget's Permissions file that reads as follows:

```
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 reads as follows:

```
MACHINE=world MYNAME=widget
```

You can also use the MYNAME option for testing purposes, as this option 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 556.

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 entering or exiting. To grant permission to deposit files in /usr/news as well as in 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 path 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 following entry permits reading any file except those files in the /etc directory (and its subdirectories) Remember, these options are prefixes.

```
READ=/ NOREAD=/etc WRITE=/var/spool/uucppublic
```

This entry 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 occurs until the calling system is called back. The reasons to set up CALLBACK are as follows:

- 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 back the remote computer before any file transfers can occur.

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 that corresponds 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 is started.

UUCP COMMANDS Option

Caution – The COMMANDS option can compromise the security of your system. Use this option 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 the requests to be transferred to the remote computer. Files and commands are sent to the target computer for remote execution, which 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 the remote system 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 following entry overrides the COMMAND default so that the computers that are named owl, raven, hawk, and dove can now execute rmail, rnews, and lp on your computer.

```
MACHINE=owl:raven:hawk:dove COMMANDS=rmail:rnews:lp
```

In addition to the names as just specified,you can have full path names of commands. For example, the following entry specifies that command rmail uses the default search path.

```
COMMANDS=rmail:/usr/local/rnews:/usr/local/lp
```

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 that are specified in the entry is executed. If you use this value, you give the remote computers full access to your machine.

Caution – This value 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.

Here is the string with the ALL value added:

```
COMMANDS=/usr/local/rnews:ALL:/usr/local/lp
```

This string illustrates two points:

- The ALL value can appear anywhere in the string.
- The path names that are 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, such as cat and uucp with the COMMANDS option. Any command that reads or writes files is potentially dangerous to local security when the command is 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 following string ensures that if any machine other than widget or gadget tries to log in as Uwidget, the connection is refused.

```
LOGNAME=Uwidget VALIDATE=widget:gadget
```

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 that are associated with this entry are protected. If an outsider obtains that information, that particular VALIDATE option can no longer be considered secure.

Carefully consider which remote computers you are granting 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 following LOGNAME entry 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:

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

If an outsider obtains the uucpfriend login and password, masquerading is easy.

But what does this entry have to do with the COMMANDS option, which appears only in MACHINE entries? This entry links the MACHINE entry (and COMMANDS option) with a LOGNAME entry that is associated with a privileged login. This link is needed because the execution daemon is not running while the remote computer is logged in. Actually, the link 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 that is 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 that is listed, you are really calling either eagle, owl, or hawk. Therefore, any files that are 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 that are 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 when the common options are the same. For example, the two sets of entries that follow share the same REQUEST, READ, and WRITE options:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
READ=/ WRITE=/
```

and

```
LOGNAME=uupz REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/
```

You can merge these entries, 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 following command, the forwarding operation works only if machine willow permits machine oak to execute the uucp program.

```
% uucp sample.txt oak\!willow\!pine\!/usr/spool/uucppublic
```

The machine oak also must permit your machine to execute the uucp program. The machine pine, as the last machine designated, does not have to permit the uucp command because the machine 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 as follows:

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. The script 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 that is 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. This file 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 a blank space. The last field in the entry is composed of subfields that are also separated by spaces. If an entry occupies 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 administrator-defined user-job-grade name of up to 64 characters.

UUCP System-job-grade Field

This field contains a single-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. 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 in compliance with the restriction on the maximum job size.

While no maximum number exists for the user job grades, the maximum number of system job grades that are 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
```

If this configuration is in a `Grades` file, these two *User-job-grade* fields 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 the 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 that are described in the following list.

nnnn	Integer that specifies the maximum job size for this job grade
n`K`	Decimal number that represents the number of kilobytes (`K` is an abbreviation for kilobyte)
n`M`	Decimal number that represents the number of megabytes (`M` is an abbreviation for megabyte)
`Any`	Keyword that specifies that no maximum job size exists

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 26–5 Permit-type Field

Keyword	ID List Contents
`User`	Login names of users who are permitted to use this job grade
`Non-user`	Login names of users who are 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, such as `uucp` or `cu`. `Devconfig` entries define the STREAMS modules that are used for a particular device. These entries have the following format:

`service=`*x* `device=`*y* `push=`*z*[:*z*...]

x can be cu, uucico, or both services 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 situations, the default values are acceptable and no changes are needed. If you want to change them, however, use any text editor.

The format of the Limits file is as follows:

service=*x* max=*y*:

x can be uucico, uuxqt or uusched, and *y* is the limit that is 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 to run 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 that is not found when any of the Systems files starts a conversation. This program logs the conversation attempt and drops the connection.

Caution – If you change the permissions of the remote.unknown file so that the file 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. The program logs the conversation attempt but fails to make a connection. If you change the permissions of this file so that the file cannot execute (`chmod 000 remote.unknown`), your system accepts any conversation requests. This change is not trivial. You should have good reasons for making this change.

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 following format:

  `TM.`*pid.ddd*

  pid is a process ID and *ddd* is a sequential three-digit number that starts at 0.

  When the entire file is received, the `TM.`*pid.ddd* file is moved to the path name that is specified in the `C.`*sysnxxxx* 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 26–6 UUCP Lock Files

File Name	Description
`LCK.`*sys*	*sys* represents the name of the computer that is using the file
`LCK.`*dev*	*dev* represents the name of a device that is 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, such as when a 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, such as file transfers or remote command executions, has been queued for a remote computer. The names of work files have the following format:

  `C.`*sysnxxxx*

sys is the name of the remote computer, *n* is the ASCII character that represents the grade (priority) of the work, and *xxxx* is the four-digit job sequence number that is assigned by UUCP. Work files contain the following information:

- Full path name of the file to be sent or be requested.
- Full path name of the destination or user or file name.
- User login name.
- List of options.
- Name of associated data files 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* – *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 subsequent number. *yyy* is used when several `D.` files are 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 that represents the grade (priority) of the work. *xxxx* is a four-digit sequence number that is assigned by UUCP. Execute files contain the following information:

  - Requester's login and computer name
  - Names of files that are 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 that are associated with UUCP.

UUCP ASSERT Error Messages

The following table lists ASSERT error messages.

TABLE 26–7 ASSERT Error Messages

Error Message	Description or 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 situations, this error is fatal.
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.
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 error occurs when a problem arises with the sequence file access, which 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 description.
BAD LINE	A bad line is in the `Devices` file. 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 description.
RETURN FROM fixline ioctl	An `ioctl(2)`, which should never fail, failed. A system driver problem has occurred.
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. This error must be fixed immediately.
PKCGET READ	The remote machine probably hung up. No action is needed.
PKXSTART	The remote machine aborted in a nonrecoverable way. This error can usually be ignored.

TABLE 26–7 ASSERT Error Messages *(Continued)*

Error Message	Description or Action
TOO MANY LOCKS	An internal problem has occurred. Contact your system vendor.
XMV ERROR	A problem with some file or directory has occurred. The spool directory is the probable cause, 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 26–8 UUCP STATUS Messages

Error Message	Description/Action
OK	Status is acceptable.
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 particular machine failed. The cause could be a wrong login or password, wrong number, a slow machine, or failure in executing the Dialer-Token-Pairs script.
CONVERSATION FAILED	The conversation failed after successful startup. This error usually means that one side went down, the program aborted, or the line (link) was dropped.
DIAL FAILED	The remote machine never answered. The cause 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 error 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 564.
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.

TABLE 26–8 UUCP STATUS Messages *(Continued)*

Error Message	Description/Action
`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. The remote machine could be trying to call your machine. If the remote machine has an older version of UUCP, the process that was talking to your machine might have failed, leaving the `LCK` file. If the remote machine 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.
`REMOTE REJECT AFTER LOGIN`	The login that was 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 error is usually the same as `DIAL FAILED`. However, if this error 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 that are produced by the `/usr/include/sysexits.h` file. Not all are currently used by `uucp`.

TABLE 26–9 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 data format 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 problem could also include errors like “No message” to a mailer.
67	Address Unknown	The user that was specified did not exist. This error might be used for mail addresses or remote logins.
68	Host Name Unknown	The host did not exist. This error is used in mail addresses or network requests.

TABLE 26–9 UUCP Error Messages by Number *(Continued)*

Message Number	Description	Meaning
69	Service Unavailable	A service is unavailable. This error can occur if a support program or file does not exist. This message also can simply indicate that something does not work and the cause currently is not identifiable.
70	Internal Software Error	An internal software error has been detected. This error should be limited to non-operating system-related errors, if possible.
71	System Error	An operating system error has been detected. This error is intended to be used for conditions like "cannot fork", "cannot create pipe." For instance, this error includes a `getuid` return of a user who does not exist in the `passwd` file.
72	Critical OS File Missing	A system file such as `/etc/passwd` or `/var/admin/utmpx` does not exist, cannot be opened, or has an error, such as a 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 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 message is not intended for file system problems, which should use `NOINPUT` or `CANTCREAT`, but rather for higher-level permissions. For example, `kre` uses this message 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.

PART VI

Working With Remote Systems Topics

This section provides instructions for administering an FTP Server and for accessing remote systems in the Solaris environment.

◆ ◆ ◆ CHAPTER 27

Working With Remote Systems (Overview)

This section includes information on working with remote files.

What Is the FTP Server?

The FTP Server is based on wu-ftpd. Originally developed by Washington University in Saint Louis, wu-ftpd is widely used for distribution of bulk data over the Internet and is the preferred standard for large FTP sites. For information on the licensing terms, refer to the materials that are incorporated at `/var/sadm/pkg/SUNWftpu/install/copyright`.

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 a Solaris release, TCP/IP configuration is established automatically during startup. For more information, see *System Administration Guide: IP Services*.

Recent Changes to the FTP Service

Previous releases include several changes to the FTP service. The changes include enhancements to the FTP server, and changes to the `ftpcount`, `ftpwho` and `ftp` commands.

The enhancements to the FTP server improve scalability and transfer logging. These options are covered in "Configuration Help for Busy Sites" on page 598 and in the `ftpaccess(4)` man page. In specific:

- The `sendfile()` function is used for binary downloads
- New capabilities supported in the `ftpaccess` file
  - `flush-wait` controls the behavior at the end of a download or directory listing
  - `ipcos` sets the IP Class of Service for either the control or data connection
  - `passive ports` can be configured so that the kernel selects the TCP port to listen on
  - `quota-info` enables retrieval of quota information
  - `recvbuf` sets the receive (upload) buffer size used for binary transfers
  - `rhostlookup` allows or disallows the lookup of the remote hosts name
  - `sendbuf` sets the send (download) buffer size used for binary transfers
  - `xferlog` format customizes the format of the transfer log entry
- `-4` option which makes the FTP server only listen for connections on an IPv4 socket when running in standalone mode

In addition, `ftpcount` and `ftpwho` now support the `-v` option, which displays user counts and process information for FTP server classes defined in virtual host `ftpaccess` files. See the `ftpcount(1)` and `ftpwho(1)` man pages for more information.

The FTP client and server now support Kerberos. For more information refer to the `ftp(4)` man page and to "Kerberos User Commands" in *System Administration Guide: Security Services*.

The `ftp` command has been changed. By default, a Solaris FTP client connected to a Solaris FTP server lists both directories as well as plain files when the `ls` command is issued to the client. If the FTP server is not running in the Solaris OS, directories may not be listed. To allow for the default Solaris behavior when connecting to non-Solaris FTP servers, the `/etc/default/ftp` file can be edited appropriately on each Solaris client. To make the change for individual users, the `FTP_LS_SENDS_NLST` environment variable can be set to `yes`. For more information see the `ftp(4)` man page.

The `ftpd` daemon is managed by the Service Management Facility. Administrative actions on this service, such as enabling, disabling, or restarting, can be performed using the `svcadm` command. The service's status for all daemons can be queried using the `svcs` command. For an overview of the Service Management Facility, refer to Chapter 18, "Managing Services (Overview)," in *System Administration Guide: Basic Administration*.

◆ ◆ ◆ CHAPTER 28

Administering the FTP Server (Tasks)

This chapter includes tasks that are described in the following table to set up and administer an FTP server.

Administering the FTP Server (Task Map)

TABLE 28–1 Task Map: Administering the FTP Server

Task	Description	For Instructions
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 576 "How to Control the Number of Invalid Login Attempts" on page 577 "How to Disallow FTP Server Access to Particular Users" on page 578 "How to Restrict Access to the Default FTP Server" on page 579 "How to Define FTP Server Classes" on page 575

TABLE 28–1 Task Map: Administering the FTP Server *(Continued)*

Task	Description	For Instructions
Set up FTP server logins	Establish login accounts for real, guest and anonymous users.	"How to Set Up Real FTP Users" on page 580 "How to Set Up Guest FTP Users" on page 581 "How to Set Up Anonymous FTP Users" on page 582 "How to Create the `/etc/shells` file" on page 582
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 584 "How to Create Messages to Be Sent to Users" on page 584 "How to Configure the README Option" on page 585
Configure access to files on the FTP server	Use the `/etc/ftpd/ftpaccess` file to specify classes of users who are 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 227 "Controlling Uploads and Downloads on the FTP Server" on page 587
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 591 "How to Enable Complete Virtual Hosting" on page 592
Start the FTP server	Change the service properties to start the FTP server in `nowait`, standalone mode or foreground mode.	"How to Start an FTP Server Using SMF" on page 594 "How to Start a Standalone FTP Server in the Background" on page 595 "How to Start a Standalone FTP Server in the Foreground" on page 595
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 596
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 597 "How to Use `greeting text` to Verify `ftpaccess`" on page 597 "How to Check the Commands Executed by FTP Users" on page 598

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 log in 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 who are permitted access from a particular host.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add entries for anonymous, guest, and real users in the `ftpaccess` file.

`class` *class typelist addrglob*[*addrglob...*]

`class` Keyword that is used to define FTP users.

class A name that is 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 28–1 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 who 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 that are 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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add the following entries to the `ftpaccess` file:

`limit` *class n times* [*message-file*]

`limit`	Keyword that is used to restrict simultaneous logins by the specified number of users of a defined class at certain connection times.
class	A name that is 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 28–2 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 that are 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 that are 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 who is logged in at a particular 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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add the following entries to the `ftpaccess` file.

`loginfails` *n*

loginfails	Keyword that is used to assign the number of login failures that are permitted before the FTP connection is terminated
n	Number of times a login can fail

Example 28–3 Controlling the Number of Invalid Login Attempts

```
loginfails 10
```

The preceding example states that the user is disconnected from the FTP server after 10 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 with a statement that the match was found in a deprecated file. The message also recommends the use of `/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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add entries to the `/etc/ftpd/ftpusers` file for users who are not allowed to log in to the FTP server.**

Example 28–4 Disallowing 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 `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 that is 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 following entry:**

      `defaultserver deny` *username* `[`*username...*`]`

`defaultserver`	Keyword that is 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 who are not listed on the `deny` line, add the following line:**

      `defaultserver allow` *username* `[`*username...*`]`

   c. **To prevent access by anonymous users, add the entry:**

      `defaultserver private`

Example 28–5 Restricting Access to the Default FTP Server

```
defaultserver deny *
defaultserver allow username
```

The previous example states that the FTP server denies access to all users except `anon` users and those users who are 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 that run 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 user name 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 log in to the FTP server by using the either `ftp` or `anonymous` as a user name. 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 that can be created for each guest user.

  Real and guest users log in by using individual accounts with passwords that are known only to one person. Anonymous users log in to a well-known account that is potentially available to anyone. Most large-scale file distribution is created by using the anonymous account.

▼ How to Set Up Real FTP Users

To enable access for real users to the FTP server, follow these instructions:

1. **Verify that the user has an account that is set up with a user name and password that can be used to establish a terminal session.**

   For more information, see Chapter 4, "Managing User Accounts and Groups (Overview)," in *System Administration Guide: Basic Administration*.

2. **Confirm that the real user is a member of a class in the `ftpaccess` file.**

   For information on the user classes that are defined in the `ftpaccess` file, see "How to Define FTP Server Classes" on page 575.

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`) that is 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, do the following:

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 Chapter 4, "Managing User Accounts and Groups (Overview)," in *System Administration Guide: Basic Administration*.

   Note – In this procedure, **`/home/guests/./guest1`** is used as the home directory name for a user who is 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 user's 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 575 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` is listed in the `/etc/shells` file. See "How to Create the `/etc/shells` file" on page 582.**

Example 28–6 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, follow these instructions:

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 575 for further information.

Example 28–7 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 or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Create the `/etc/shells` file.**

3. **Edit `/etc/shells`. Add the full path to each shell on a single line.**

Example 28–8 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 that are 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 that is 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 host name is not defined in the `ftpaccess` file, the default host name for the local machine is used.

How to Customize Message Files

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit your message file to include magic cookies as appropriate.**

   See `ftpaccess(4)` for a list of cookies you can use.

Example 28–9 Customizing Message Files

The following is an example of a message file that 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-related or application-related messages are displayed on screen. The `ftpaccess` file lists the events that trigger associated `message` statements.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add the following entries to the `ftpaccess` file:**

   `message` *message-file* `[`*when* `[`*class* ...`]]`

`message`	Keyword that is 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 that is 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 28–10 Creating 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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add the following entries to the `ftpaccess` file.

`readme` *message-file* `[when [`*class*`...]]`

`readme` Keyword that is 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 that is 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 send messages to users. See `ftpaccess(4)`.

Example 28–11 Configuring the README Option

```
readme  README*     login
readme  README*     cwd=*
```

The previous example states that any files that match `README*` are listed at login or when a directory is changed. Here is a sample login that is based on the settings that are 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 the release. Use the standard 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, do the following:

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 28–12 Controlling File Access Commands

The following are examples of permissions that are 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 the following:

- 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 that are started 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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add the following entries to the `ftpaccess` file.

To enable users to upload files, add the following 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 that is applied to users who have a home directory (the argument to `chroot()`) of the *root-dir*. The *root-dir* can 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 that is used to specify any number of `class=<classname>` restrictions. If restrictions are specified, the upload clause only becomes effective 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 that are uploaded into `dirnames`.
group	Group that is associated with files that are uploaded into `dirnames`.
mode	Parameter that is 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 that is 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 file names.

Example 28–13 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 the following:

- FTP user accounts that use `chroot` to `/export/home/ftp` can upload to the `/incoming` directory. Uploaded files are owned by user `ftpadm` and the group `ftpadmin`. The 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 cannot start with a . (dot) or - (dash). If a file name fails this filter, the `/etc/ftpd/filename.msg` message is displayed if the FTP Administrator has created the message file. This message is followed by an FTP server error message.

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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add the following entries to the `ftpaccess` file to prevent users from retrieving files.

`noretrieve [absolute|relative] [class=`*classname*`]... [-]` *filename ...*

`noretrieve`	Keyword that is 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 that is used to specify `class=<classname>` of users to which `noretrieve` restrictions apply
filename	Name of file the user is not permitted to retrieve

Example 28–14 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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add the following entries to the `ftpaccess` file.

```
virtual address root|banner|logfile path
virtual address hostname|email string
```

`virtual`	Keyword that is used to enable virtual server capabilities
address	IP address of the virtual server
`root`	The root directory of the virtual server
`banner`	Banner file that is displayed when a connection is made to the virtual server
`logfile`	Record of file transfers that are made to and from the virtual server
path	Variable that is used to specify the location of directories and files on the virtual server
`email`	Email address that is used in message files and in the `HELP` command
`hostname`	Name of the host that is shown in the greeting message or status command
string	Variable that is 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 to 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 28–15 Enabling Limited Virtual Hosting in the `ftpaccess` File

```
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.

Example 28–16 Enabling Limited Virtual Hosting on the Command Line

The `ftpaddhost(1M)` script with the `-l` option is provided to configure limited virtual hosts.

In the following example, `ftpaddhost` is run with `-l -b -x` options to configure 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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 that are customized for the virtual host

Note – While it is possible to use `hostname` as the *address* of the virtual server, you are strongly encouraged to 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 28–17 Enabling Complete Virtual Hosting in the `ftpservers` file

```
#
# 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.

Example 28–18 Enabling Complete Virtual Hosting from the Command Line

The `ftpaddhost(1M)` script with the `-c` option is provided to configure complete virtual hosts.

In the following example, `ftpaddhost` is run with `-c -b -x` options to configure complete 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 -c -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 three ways:

- As a `nowait` server that is started by `inetd`
- As a standalone server run in the background
- As a standalone server run in the foreground from the `inittab` file

A standalone server always has the quickest possible response time, and is intended for large servers that are 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 an FTP Server Using SMF

By default, the SMF service is configured to start the FTP server using the `nowait` mode. If the site handles many connections, the FTP server can also be run in standalone mode. See the `in.ftpd`(1M) man page for information on additional command-line options.

1 Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Verify the `wait` property for the FTP server.

The line reporting that `wait=FALSE` indicates that the server is started in `nowait` mode.

```
# inetadm -l network/ftp
SCOPE    NAME=VALUE
         name="ftp"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/sbin/in.ftpd -a"
         user="root"
default  bind_addr=""
default  bind_fail_max=-1
default  bind_fail_interval=-1
default  max_con_rate=-1
default  max_copies=-1
default  con_rate_offline=-1
default  failrate_cnt=40
default  failrate_interval=60
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
```

3 Start the FTP server.

```
# svcadm enable network/ftp
```

How to Start a Standalone FTP Server in the Background

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Disable the FTP server.**

   ```
   # svcadm disable network/ftp
   ```

3. **Start the standalone FTP server.**

   ```
   # /usr/sbin/in.ftpd -a -S
   ```

   Add the line to an FTP server startup script. See "Using Run Control Scripts" in *System Administration Guide: Basic Administration* for information on creating a system startup script.

How to Start a Standalone FTP Server in the Foreground

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Disable the FTP server.**

   ```
   # svcadm disable network/ftp
   ```

3. **Add an entry to the `inittab` file to start the service.**

   The new entry in `/etc/inittab` should look something like the following:

   ```
   ftpd:3:respawn:/usr/sbin/in.ftpd -a -s
   ```

4. **Tell `init` to re-examine `/etc/inittab`.**

   This command should start the FTP service.

   ```
   # init q
   ```

Shutting Down the FTP Server

The `ftpshut(1M)` command closes down the FTP server at a particular time.

When you run `ftpshut`, a file is generated from command-line options that specify when shutdown occurs, 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 that is 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 or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 Add the following entries to the `ftpaccess` file.

`shutdown` *path*

`shutdown`	Keyword that is used to specify the *path* to a file that is checked regularly for whether the FTP server is scheduled to be shut down
path	Location of the file that was 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 that is specified to display copyright and version information, then terminate
`-l`	Flag that is used to adjust the time that new connections to the FTP server are denied
`-d`	Flag that is used to adjust the time that existing connections to the FTP server are disconnected.
`time`	Shutdown time that is 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 that are useful for debugging to the location that is specified for daemon messages in the `/etc/syslog.conf` file. If a problem occurs with the FTP server, check this file first 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, follow these instructions:

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.**

```
# svcadm refresh system/system-log
```

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, do the following:

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, do the following:**

   a. **Confirm that the `ftpaccess` file is in the correct location. Use the `strings(1)` command to obtain 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)`, `ftpservers(4)`, `strings(1)`, `syslog.conf(4)`, and `pgrep(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 who are specified in *typelist*.**

```
log commands typelist
```

2 **Check messages that are written to the location specified in the `/etc/syslog.conf`.**

Configuration Help for Busy Sites

The following list includes some suggestions to improve performance on busy FTP sites.

1. Sites which typically support many simultaneous connections should run the FTP server in standalone mode, see "Starting the FTP Server Automatically" on page 593.
2. Use `vmstat` and other system utilities to monitor the system hosting the FTP server. If the system runs low on resources place a limit on the number of simultaneous connections, see "How to Set User Login Limits" on page 576. For more information on system monitoring, see Chapter 13, "Monitoring System Performance (Tasks)," in *System Administration Guide: Advanced Administration*.
3. If you impose a connection limit, consider using the `limit-time` and `timeout idle` capabilities in the `ftpaccess` file to stop users from hogging connections. If you don't impose a connection limit, specify the `-Q` option to `in.ftpd`.
4. If you don't need ftp login and logout records in `/var/adm/wtmpx`, specify the `-W` option to `in.ftpd`.

5. To reduce the load on the system hosting the FTP server, increase the transfer buffer sizes using the `recvbuf` and `sendbuf` capabilities in the `ftpaccess` file. If large buffer sizes are selected it may be necessary to increase the data activity timeout using the `timeout data` capability in the `ftpaccess` file.
6. The FTP server reads from various databases including the hosts, passwd, group and services. Slow lookups may cause a significant delay logging into the FTP server, configuring the `files` source first in `nsswitch.conf` minimizes the lookup times. For more information, see the `nsswitch.conf(4)` man page.
7. By default the FTP server attempts to lookup the remote host's name, which can be slow causing a significant delay logging in. The `rhostlookup` capability in the `ftpaccess` file can be used to stop this lookup. However be aware that if the remote host's name is not looked up, only its IP address is matched when using other capabilities in the `ftpaccess` file and when matching entries in the `ftphosts` file. Also the remote host's IP address will be used in messages and in place of the %R magic cookie. See the description of the `rhostlookup` capability in the `ftpaccess(4)` man page for more details.
8. Retrieving quota information may also cause a significant delay when logging into the FTP server, so only use the `quota-info` capability in the `ftpaccess` file if you make use of the quota magic cookies. See the `ftpaccess(4)` man page for a list of the quota magic cookies.

◆ ◆ ◆ CHAPTER 29

Accessing Remote Systems (Tasks)

This chapter describes all the tasks that are 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.

Accessing Remote Systems (Task Map)

This chapter provides tasks that are described in the following table to log in and copy files from remote systems.

TABLE 29–1 Task Map: Accessing Remote Systems

Task	Description	For Instructions
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 606 "How to Find Out If a Remote System Is Operating" on page 607 "How to Find Who Is Logged In to a Remote System" on page 607 "How to Log In to a Remote System (`rlogin`)" on page 608 "How to Log Out From a Remote System (`exit`)" on page 609

TABLE 29–1 Task Map: Accessing Remote Systems *(Continued)*

Task	Description	For Instructions
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 610 "How to Close an ftp Connection to a Remote System" on page 611 "How to Copy Files From a Remote System (ftp)" on page 612 "How to Copy Files to a Remote System (ftp)" on page 614
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 618

Logging In to a Remote System (rlogin)

The rlogin command enables you to log in to a remote system. After you are 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 in to 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 are prompted for a password, unless you set up the /etc/hosts.equiv or .rhosts file. If the network tries to authenticate you, you are not asked for a password, because 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 one of the following is true:

- Your system name and user name appear in the remote system's /etc/hosts.equiv file.
- Your system name and user name appear 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 that is pointed to by the remote system's /etc/nsswitch.conf file contains information about you.

Note – Network authentication generally supersedes system authentication.

/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 that is 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 previous entry 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 that is preceded by a plus sign (+) means that all the machines in that netgroup are considered trusted.

A group name that is preceded by a minus sign (–) means that none of the machines in that netgroup is 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.

The use of hosts that are not trusted 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.

.rhosts File

The .rhosts file is the user equivalent of the /etc/hosts.equiv file. This file 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 that are 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 users' 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 that grants 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 change would effectively prevent a user from risking system security by using a .rhosts file irresponsibly. The change 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 606 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

If 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 or 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 who is 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. However, 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 following table.

TABLE 29–2 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 assigns 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 resembles the following:

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

Then when you log in to a remote system, you see a similar prompt and your working directory is 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 substitute for your local system at the beginning of the prompt. The remote file system is parallel to your home directory.

Effectively, if you change directory to /home and then run ls, you see the following:

```
earth(home/smith): cd ..
earth(/home): ls
smith  jones
```

How to Search for and Remove .rhosts Files

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

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 file name.
-print	Prints the current path name.
-exec rm {} \;	Tells the find command to apply the rm command to all files that are identified by using the matching file name.

   The find command starts at the designated directory and searches for any file that is named .rhosts. If it finds such as file, find prints the path on the screen and removes it.

Example 29–1 Searching for and Removing .rhosts Files

The following example searches and removes .rhosts files in all the user's home directories that are 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 timeout of 20 seconds. Effectively, if it does not receive a response within 20 seconds, it returns the third message. You can force ping to wait longer (or less) by typing 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 who are 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 29–2 Finding Who Is Logged In to a Remote System

The following example shows the short output of rusers.

EXAMPLE 29–2 Finding Who Is Logged In to a Remote System *(Continued)*

```
$ rusers pluto
pluto    smith  jones
```

In the following example, the long version of `rusers` shows 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 by using the `rlogin(1)` command.

```
$ rlogin [-l user-name] system-name
```

`rlogin`	(No options) Logs you in to the remote system *directly*, effectively, with your current user name
`-l` *user-name*	Logs you into the remote system *indirectly*, effectively, with the user name you supply

If the network attempts to authenticate you, you are not prompted for a password. If the remote system attempts to authenticate you, you are 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 that is running on the remote system, and whether you have mail waiting for you in your home directory.

EXAMPLE 29–3 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
```

EXAMPLE 29–3 Logging In to a Remote System (rlogin) *(Continued)*

```
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 29–4 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 following table.

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. However, rlogin provides access to a richer set of file manipulation commands than ftp provides.

Authentication for Remote Logins (ftp)

Authentication for ftp remote login operations can be established by one of the following methods:

- 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 29–3 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 610.
`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 Ensure that you have ftp authentication.

You must have `ftp` authentication, as described in "Authentication for Remote Logins (ftp)" on page 609.

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 are displayed.

3 Type your user name.

```
Name (remote-system:user-name): user-name
```

4 If prompted, type your password.

```
331 Password required for user-name:
Password: password
```

If the system you are accessing has an established anonymous ftp account, you are 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 that are supplied by the ftp interface, including help. The principal commands are summarized in Table 29–3.

Example 29–5 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 610.

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. **Ensure that 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 copies each file individually, asking you for confirmation each time.

8. **Close the `ftp` connections.**

   `ftp> bye`

Example 29–6 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).
filea
files
ps_data
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 type the ftp command is 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 610.

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. **Ensure that you have write permission to the target directory.**

   ```
   ftp> ls -l target-directory
   ```

5. **Set the 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 copies each file individually, asking you for confirmation each time.

8. **To close the ftp connection, type bye.**

   ```
   ftp> bye
   ```

Example 29–7 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 or her 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).
filea
filef
files
ps_data
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 or her 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 this command 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 that is created by the automounter and to operations relative to a user's home directory. Because rcp performs the same operations without these constraints, this section describes 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. Ensure that 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 Path Names	Abbreviated Path Names
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 that are mounted on a particular system. In the previous example, 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 it might reside. In the previous first example, the abbreviated path name identifies the same file, myfile.txt, but uses "~" symbol to indicate the jones home directory:

```
~ = mars:/home/jones
```

The examples on the second line demonstrate the user of absolute and abbreviated path names after a remote login. No difference is evident for the abbreviated path name. However, 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 605.

The following table provides a sample of absolute and abbreviated path names that are recognized by the C shell. The sample 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 29–4 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*
	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*

TABLE 29–4 Allowed Syntaxes for Directory and File Names *(Continued)*

Logged in to	Syntax	Description
	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 Ensure that 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 in to the remote system with the `rlogin` command, as described in "How to Log In to a Remote System (rlogin)" on page 608. 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 29–4 and as illustrated in the following examples.

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 in to. For explanations of these symbols, see Table 29–4.

Example 29–8 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. Here, 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.

Example 29–9 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 allow for 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 the login is to a remote system.

Example 29–10 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 file name 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 29–11 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 allow for the remote login.

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

In this instance, the “~” symbol can be used to denote the current user's home directory, even though it is on the local system. The “ . ” symbol refers to the working directory on the remote system because the user is logged in to the remote system. Here is an alternative syntax that performs the same operation:

```
pluto(/home/jones): rcp earth:/home/smith/notice.doc /home/jones
```

PART VII

Monitoring Network Services Topics

This section provides step-by-step instructions for monitoring network services.

◆ ◆ ◆ CHAPTER 30

Monitoring Network Performance (Tasks)

This chapter describes how to monitor network performance. The following is a list of the step-by-step instructions in this chapter.

Monitoring Network Performance

Table 30–1 describes the commands that are available for monitoring network performance.

TABLE 30–1 Network Monitoring Commands

Command	Description
`ping`	Look at the response of hosts on the network.
`spray`	Test the reliability of your packet sizes. This command can tell you whether the network is delaying packets or dropping packets.
`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 that are 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 the following:

- Loose cables or connectors
- Improper grounding
- No termination
- Signal reflection

For more information about this command, see `ping(1M)`.

EXAMPLE 30–1 Checking the Response of Hosts on the Network

The simplest version of `ping` sends a single packet to a host on the network. If `ping` receives the correct response, the command prints the message *host* `is alive`.

```
$ ping elvis
elvis is alive
```

With the `-s` option, `ping` sends one datagram per second to a host. The command then prints each response and the time that was required for the round trip. An example follows.

```
$ ping -s pluto
64 bytes from pluto (123.456.78.90): icmp_seq=0. time=3.82 ms
64 bytes from pluto (123.456.78.90): icmp_seq=5. time=0.947 ms
64 bytes from pluto (123.456.78.90): icmp_seq=6. time=0.855 ms
^C
----pluto PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss

round-trip (ms) min/avg/max/sttdev = 0.855/1.87/3.82/1.7
```

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 do not use a delay, you might deplete the 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 30–2 Sending Packets to Hosts on the Network

The following example sends 100 packets to a host (`-c 100`), with a packet 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 enable 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

To display network status information, such as statistics about the state of network interfaces, routing tables, and various protocols, use 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 that are 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
eri0   1500 loopback    venus      1628480   0   347070    16   39354    0
```

This display shows the number of packets that a machine has transmitted and has 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 that is greater than 5 to 10 percent can indicate a problem.

Calculate the error rate for the input packets by dividing the number of input errors by the total number of input packets (`Ierrs/Ipkts`). The error rate for the output packets is the number of output errors divided by the total number of output packets (`Oerrs/Opkts`). If the input error rate is high, at 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  eri0
224.0.0.0          pluto                  U        3      0  eri0
default            mars-gate              UG       0  14142
```

The fields in the `netstat -r` report are described in the following table.

TABLE 30–2 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 by using a redirect.
`Ref`		Shows the current number of routes that share the same link layer.
`Use`		Indicates the number of packets that were sent out.
`Interface`		Lists the network interface that is 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. The client application is blocked or suspended until the server has completed the call and has returned the results. One of the major factors that affects 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 the client 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 the following:

- Overloaded servers that complete requests too slowly
- An Ethernet interface that is dropping packets
- Network congestion, which slows the packet transmission

The following table describes the `nfsstat` options to display client and server statistics.

TABLE 30–3 Commands for Displaying Client/Server Statistics

Command	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 the following table.

TABLE 30–4 Output From the `nfsstat -c` Command

Field	Description
`calls`	The total number of calls that were sent.
`badcalls`	The total number of calls that were rejected by RPC.

TABLE 30–4 Output From the `nfsstat -c` Command *(Continued)*

Field	Description
`retrans`	The total number of retransmissions. For this client, the number of retransmissions is less than 1 percent, or approximately 10 timeouts out of 6888 calls. These retransmissions 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, at over 10 percent, then there could be 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 the following table.

TABLE 30–5 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 closely at the cabling and connectors.

Glossary

asppp A version of PPP that was included with the operating system from the Solaris 2.4 to the Solaris 8 releases. `asppp` supported asynchronous PPP communications only.

asynchronous PPP A form of PPP that runs over asynchronous serial lines, which transfer data one character at a time. The most common form of PPP configuration, the dial-up link, uses asynchronous PPP communications.

authentication The act of verifying the identity that is supplied over the network by a remote user or entity, such as a program. Some authentication protocols enable you to build databases of authentication credentials from potential users. Other authentication protocols use certificate chains of trust that are generated by a certificate authority for authentication purposes. These credentials can authenticate users when they try to communicate with you or use your site's services.

broadcast A data-link layer procedure that is used to transmit packets to every machine on a subnet. Broadcast packets are typically not routed beyond the subnet.

Callback Control Protocol (CBCP) A proprietary Microsoft PPP extension that is used to negotiate a callback session. Solaris PPP 4.0 supports only the client (initial caller) side of this protocol.

channel service unit (CSU) A synchronous telecommunications device that provides a local interface to a leased telecommunications line and terminates that line. In the United States, a CSU terminates a T1 line and provides a DS1 or DSX interface. Internationally, the CSU is typically owned by the telephone company provider.

See also CSU/DSU and data service unit (DSU).

CHAP The Challenge-Handshake Authentication Protocol is an authentication protocol that can be used to verify the identity of a caller on a PPP link. CHAP authentication uses the notion of the *challenge* and *response*, where the machine that receives a call challenges the caller to prove its identity.

See also password authentication protocol (PAP).

CHAP secret An ASCII or binary string that is used for identification purposes and is known to both peers on a PPP link. The CHAP secret is stored in clear text in a system's `/etc/ppp/chap-secrets` file but is never sent over the PPP link, not even in encrypted form. The CHAP protocol verifies that a hash of the CHAP secret that is used by a caller matches a hash of the CHAP secret entry for the caller in the recipient's `/etc/ppp/chap-secrets` file.

chat script Instructions that tell a modem how to establish a communications link between itself and a remote peer. Both the PPP and UUCP protocols use chat scripts for establishing dial-up links and dial-back calling.

Compression Control Protocol (CCP) A subprotocol of PPP that negotiates the use of data compression on the link. Unlike header compression, CCP compresses all data within packets that are sent on the link.

CSU/DSU A synchronous telecommunications device that combines the CSU and DSU devices and is used on a leased-line PPP link. The CSU/DSU translates signals from a peer to a leased line. Most CSU/DSUs do not need a chat script to establish the link. CSU/DSUs are often configured by the leased-line provider.

See also channel service unit (CSU) and data service unit (DSU).

data service unit (DSU) A synchronous telecommunications device that is used on a leased-line PPP link. The DSU converts between data-framing formats that are used on telecommunications lines and provides a standard data communications interface.

See also channel service unit (CSU) and CSU/DSU.

dial-in server The peer that negotiates and establishes the recipient end of a dial-up PPP link after receiving a call from a dial-out machine. Though the term "dial-in server" is in common use, the dial-in server does not function in accordance with the client-server paradigm. Rather, it is simply the peer that responds to the request to set up a dial-up link. After it is configured, a dial-in server can receive calls from any number of dial-out machines.

dial-out machine The peer that initiates the call to establish a dial-up PPP link. After it is configured, the dial-out machine can call any number of dial-in servers. The dial-out machine typically provides authentication credentials before the dial-up link can be established.

dial-up PPP link A PPP connection that involves a peer and a modem at either end of a telephone line or similar communications medium, such as a medium that is provided by ISDN. The term "dial-up" refers to the sequence in link negotiation when the local modem dials up the remote peer by using the peer's telephone number. The dial-up link is the most common and least expensive PPP configuration.

Directory Agent (DA) Optional SLP agent that stores and maintains a cache of service advertisements that are 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.

expect-send A scripting format that is used in PPP and UUCP chat scripts. The chat script begins with the text or instruction to *expect* from the remote peer. The next line contains the response to be *sent* from the local host after it receives the correct expect string from the peer. Subsequent lines repeat the expect-send instructions between local host and peer until all instructions that are required to establish communications are successfully negotiated.

extended accounting A flexible way to record resource consumption on a task or process basis.

Internet Protocol Control Protocol (IPCP) A subprotocol of PPP that negotiates the IP addresses of the peers on the link. IPCP also negotiates header compression for the link and enables the use of the network layer protocols.

Internet Protocol Version 6 Control Protocol (IPV6CP) See Internet Protocol Control Protocol (IPCP).

ISDN terminal adaptor (TA) A signal-adapting device that provides a modem-like interface for a dial-up PPP link over an ISDN network. You use the same Solaris PPP 4.0 configuration files to configure an ISDN TA as you use for configuring a standard modem.

leased-line PPP link A PPP connection that involves a host and a CSU/DSU that are connected to a synchronous network medium leased from a provider. OC3 and T1 are common examples of leased-line media. Though easier to administer, leased-line links are more expensive than dial-up PPP links and, therefore, are less common.

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 10, "Incorporating Legacy Services").

link In PPP, the communications connection that is negotiated and established between two peers. Solaris PPP 4.0 supports two types of links: dial up and leased line.

link control protocol (LCP) A subprotocol of PPP that is used to negotiate the initial set of link parameters between the peers. Part of the function of LCP is to test the link integrity, so many link-related problems manifest themselves as LCP failure.

Microsoft CHAP (MS-CHAP) A proprietary Microsoft authentication protocol for PPP. Solaris PPP 4.0 supports versions 1 and 2 of this protocol in both client and server mode.

multicast A network layer procedure that is used to send datagram packets to multiple machines on an IP network. Packets are not handled by every machine as is the situation with broadcast routing. Multicast requires that routers be configured with special routing protocols.

password authentication protocol (PAP) An authentication protocol that can be used to verify the identity of a caller on a PPP link. PAP uses a cleartext password that is passed over the link, which makes it possible to store the password on one of the endpoint machines. For example, PAP can use the login and password entries in the UNIX `passwd` database on the machine that receives a call to verify the identity of the caller.

See also CHAP.

peer In PPP, an individual computer at one end of a PPP communications link, which consists of two peers that are connected by communications media. You can configure many types of computing equipment as a peer, such as a workstation, personal computer, router, or mainframe.

point-to-point protocol (PPP) A data-link layer protocol that provides a standard method for transferring datagrams over point-to-point media. A PPP configuration consists of two endpoint computers called *peers*, and the telephone lines or another bidirectional link that the peers use for communication. The hardware and software connection between the two peers is considered the *PPP link*.

PPP is composed of a number of subprotocols, including PAP, CHAP, LCP, and CCP. Numerous PPP implementations are available.

PPP over Ethernet (PPPoE) A proprietary protocol from RedBack Networks that enables hosts to run PPP sessions over an Ethernet link. PPPoE is commonly used with Digital Subscriber Line (DSL) services.

scope A grouping of UAs and SAs that are arranged administratively, topologically, or in some other manner. You can use scopes to modify how you provision access to services across the enterprise.

service advertisements Information that is 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 reregistered.

Service Agent (SA) The SLP agent that maintains service advertisements for networked services. 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 that is required to use the service.

SLP daemon (slpd) The daemon process that acts as a DA or an SA server in the Oracle Solaris implementation of SLP. Service processes on the host register service advertisements with `slpd` instead of 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. You can optionally configure `slpd` as a DA.

synchronous PPP A form of PPP that runs over synchronous digital lines, which transfer data as a continuous stream of raw bits. The leased-line PPP link uses synchronous PPP.

trusted callers In PPP, remote peers that a dial-in server grants access to by including the peers' security credentials in the server's PAP or CHAP secrets database.

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.

Index

B

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X