

Chapter 10

Network Services



***HP-UX Handbook
Revision 13.00***

TERMS OF USE AND LEGAL RESTRICTIONS FOR THE HP-UX RECOVERY HANDBOOK

ATTENTION: PLEASE READ THESE TERMS CAREFULLY BEFORE USING THE HP-UX HANDBOOK. USING THESE MATERIALS INDICATES THAT YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THESE TERMS, DO NOT USE THE HP-UX HANDBOOK.

THE HP-UX HANDBOOK HAS BEEN COMPILED FROM THE NOTES OF HP ENGINEERS AND CONTAINS HP CONFIDENTIAL INFORMATION.

THE HP-UX HANDBOOK IS NOT A PRODUCT QUALITY DOCUMENT AND IS NOT NECESSARILY MAINTAINED OR UP TO DATE. THE HP-UX HANDBOOK IS HERE MADE AVAILABLE TO HP CONTRACT CUSTOMERS FOR THEIR INTERNAL USE ONLY AND ON THE CONDITION THAT NEITHER THE HP-UX HANDBOOK NOR ANY OF THE MATERIALS IT CONTAINS IS PASSED ON TO ANY THIRD PARTY.

Use of the HP-UX Handbook: Hewlett-Packard Company ("HP") authorizes you to view and download the HP-UX Handbook only for internal use by you, a valued HP Contract Customer, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials. You may not modify the HP-UX Handbook in any way or publicly display, perform, or distribute or otherwise use them for any public or purpose outside your own business. The materials comprising the HP-UX Handbook are copyrighted and any unauthorized use of these materials may violate copyright, trademark, and other laws. If you breach any of these Terms, your authorization to use the HP-UX Handbook automatically terminates and you must immediately destroy any downloaded or printed materials.

Links To Other Web Sites: Links to third party Web sites provided by the HP-UX Handbook are provided solely as a convenience to you. If you use these links, you will leave this Site. HP has not reviewed all of these third party sites and does not control and is not responsible for any of these sites or their content. Thus, HP does not endorse or make any representations about them, or any information, software or other products or materials found there, or any results that may be obtained from using them. If you decide to access any of the third party sites linked to this Site, you do this entirely at your own risk.

Disclaimer: THE HP-UX HANDBOOK IS PROVIDED "AS IS" WITHOUT ANY WARRANTIES OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. HP further does not warrant the accuracy and completeness of the materials in the HP-UX Handbook. HP may make changes to the HP-UX Handbook at any time without notice. The HP-UX Handbook may be out of date, and HP makes no commitment to update the HP-UX Handbook. Information in the HP-UX Handbook may refer to products, programs or services that are not available in your country. Consult your local HP business contact for information regarding the products, programs and services that may be available to you.

Limitation of Liability: IN NO EVENT WILL HP, ITS SUPPLIERS, OR OTHER ANY THIRD PARTIES MENTIONED IN THE HP-UX HANDBOOK BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOST PROFITS, LOST DATA OR BUSINESS INTERRUPTION) ARISING OUT OF THE USE, INABILITY TO USE, OR THE RESULTS OF USE OF THE HP-UX HANDBOOK, WHETHER BASED ON WARRANTY, CONTRACT, TORT OR ANY OTHER LEGAL THEORY AND WHETHER OR NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS DOES NOT APPLY IN CASE OF INTENT OR IF LIABILITY IS LEGALLY STIPULATED. IF YOUR USE OF THE HP-UX HANDBOOK RESULTS IN THE NEED FOR SERVICING, REPAIR OR CORRECTION OF EQUIPMENT OR DATA, YOU ASSUME ALL COSTS THEREOF.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Applicable Laws: These Terms will be governed by and construed in accordance with the laws of the State of California, without giving effect to any principles of conflicts of laws.

General: HP may revise these Terms at any time by updating this posting. *Revised Oct 2013*

© Copyright 2000-2006, 2013 Hewlett-Packard Development Company, L.P

FEEDBACK or QUESTIONS:

(please use subject syntax:

please email essam.ackleh@hp.com

HP-UX Handbook v13.00 Chapter <YY> - <Feedback Title>

TABLE OF CONTENTS

Introduction	4
Host Based Network Support	4
Name Resolution	4
Name Service Switching	5
The system as a DNS client	6
Verifying hostname resolution	7
The system as a NIS client	8
The system as a LDAP-UX client	9
Well-Known Network Services	10
NFS	11
What is configurable on an NFS client?	12
What can be configured on the NFS Server?	13
Involved Scripts and Processes	15
Information required for troubleshooting	15
Common errors include:	16
AutoFS	16
What information can we get from automountd?	17
Commonly used network services	18
telnet	18
ftp	19
r-commands	20
ssh	21
Permissions for the Client Files and Directories	24
Additional Information	26
General Knowledge	26
Internet Standards – RFC	26

Introduction

This document is a short introduction to some commonly used network services on HP-UX systems. It is not intended to be an exhaustive survey of services. Since most, if not all, network services rely heavily on solid and consistent name resolution, this is addressed first. This is followed by a general description of some commonly seen network services.

Host Based Network Support

Name Resolution

A central service being used by almost every program is name resolution, the correlation between names and IP addresses. The functions `gethostbyname()` and `getnamebyaddr()` (implemented within libc) are the basis of this service in a IPv4-only environment. In the dual stack IPv4-IPv6 environment found in HP-UX 11iv2 and higher these are replaced by `getipnodebyname()` and `getipnodebyaddr()` functions.

These basic functions can use different information sources:

- the local file: `/etc/hosts`
- the Network Information System (NIS, previously called “Yellow Pages”)
- the Domain Name System (DNS)
- Lightweight Directory Access Protocol (LDAP)

Rarely, there may be a NIS+ configuration on HP-UX on versions prior to HP-UX 11.31; it is, however, a dead technology. In such a configuration, `/etc/nsswitch.conf` would contain the service switch `nisplus`.

NIS is still available on all supported versions of HP-UX and is still used by customers (even some large customer environments) but suffers from scalability issues that make it undesirable for large enterprise environments. For common necessary unix information such as UID, GID, and GECOS information in a large enterprise environment, LDAP is the *de facto* industry standard.

For hostname-only information DNS would be still be considered the *de facto* industry standard.

Name Service Switching

Most, if not all, network services are designed to be Name Service Switch (NSS) aware.

For instance, the telnetd daemon in the process of setting up an incoming telnet connection, will as part of the program logic, verify the source IP address on the incoming request. To do this the telnetd program opens /etc/nsswitch.conf if it is present and follows the instructions for ipnodes first and then hosts. In the case below, /etc/hosts (files) will be searched first and then dns.

```
passwd:      files ldap
group:       files ldap
hosts:        files [NOTFOUND=continue] dns
ipnodes:     files
services:    files
networks:   files
protocols:  files
rpc:         files
publickey:  files
netgroup:   files
automount: files
aliases:    files
```

If there is no /etc/nsswitch.conf configured then the choices provided in the file are made by the program. The /etc/nsswitch.hp_defaults lists the following as the default for hostname resolution:

```
hosts:        dns [NOTFOUND=return] nis [NOTFOUND=return] files
ipnodes:     dns [NOTFOUND=return] nis [NOTFOUND=return] files
```

ADVICE: It is essential that /etc/nsswitch.conf be configured to safeguard against unintended and undesirable resolution behavior. A poor design (resulting from blindly taking the defaults) can cause performance problems for applications.

Example:

Environment: DNS caching-only nameserver with no /etc/nsswitch.conf and application using loopback for inter process communication (IPC).

Default /etc/nsswitch.conf values:

```
hosts:      dns [NOTFOUND=return] nis [NOTFOUND=return] files
ipnodes:    dns [NOTFOUND=return] nis [NOTFOUND=return] files
```

If the application actually uses the name localhost instead of the loopback IP address of 127.0.0.1 it means that every query to localhost will mean a query to DNS which will not be answered by the nameserver so the query gets sent to another nameserver for outside queries which return NXDOMAIN (no answer) or possibly SERVFAIL return code. The application gets to wait on this for each attempted IPC.

ADVICE: The entry “files” corresponds to the simplest method to resolve names on the system; this is the /etc/hosts file. The file should contain at least entries for localhost, loopback, the hostname, and the local IP addresses, e.g.:

```
# grep loopback /etc/hosts
127.0.0.1   localhost   loopback

# grep `hostname` /etc/hosts
15.140.10.113  grcdg089
```

If you do not use IPv6 addresses in your environment, it is a good idea to modify the **Database** entry for ipnodes in /etc/nsswitch.conf.

```
ipnodes: files
```

This will not harm your IPv4 network, but it will prevent the system from asking the nameserver for the IPv6 address, which may cause unexpected results.

Also, consider using /etc/hosts as the first source for all queries. Since all essential hostname/IP mappings should be in /etc/hosts, the dependence on an outside source for resolution can be minimized. (For example, /etc/fstab NFS entries which are mounted at every system startup or an xntpd time server which is configured in startup scripts)

The system as a DNS client

The existence of /etc/resolv.conf is sufficient.

```
# cat /etc/resolv.conf
domain      grc.hp.com
nameserver  15.137.22.252
```

Simple single level qualified hostnames will be expanded to complete DNS names with the designation specified in the domain statement. If information for systems from different

subdomains should be found, a search statement can be used (instead of the domain statement), e.g.:

```
search    grc.hp.com, bbn.hp.com
```

Up to three *nameserver* statements can be entered into this file. These *must be* specified by IP address, not hostname.

Current versions of libc allow a custom timeout for fallback through directives (*retry* and *retrans*), see man 4 resolver page for details.

The options *retrans* and *retry* are a good choice where nameservers are not always reliable.

```
retrans <value in milliseconds>
```

```
retry <number of retries>
```

Verifying hostname resolution

Historically, *nslookup* has been the tool of choice for resolving host names; however, this tool is not a single standard (vendor implementation details vary) and suffers from other drawbacks.

One major drawback is that *nslookup* does not use the same library as resolver calls and uses a different algorithm for probing nameservers. The *nslookup* tool will often falsely report a very long response time for looking up a host address which can be quickly resolved by using a service (such as telnet or ping) which uses the function calls previously described. As previously stated various versions are available from different vendors. HP's version was enhanced to be NSS aware.

The *nslookup* tool is not, going forward, the best tool to use for DNS queries. A better alternative is HP's proprietary tool *nsquery* which is also NSS aware.

For instance,

```
$ nsquery hosts `hostname`
```

Using "files dns" for the hosts policy.

Searching /etc/hosts for rx7620b

Hostname: rx7620b.example.com

Address: 10.226.90.20

Switch configuration: Terminates Search

The reasoning behind this recommendation is that nsquery uses the same resolver routines as a typical application; therefore, it will act similarly if there is an unresponsive nameserver referenced in /etc/resolv.conf.

The DNS BIND distribution provides the tool dig which is an industry standard and very useful for seeing the complete DNS server's response to queries.

Often it is good to check for consistency between forward and reverse lookups.

```
# nsquery hosts <suspect name>
# nsquery hosts <expected IP>
```

Please be aware that the different commands handle their own caching mechanisms for names and resolver policies to improve performance (after changing /etc/nsswitch.conf, some long running processes such as inetd may need to be restarted to see the changes).

The system as a NIS client

Run *domainname* to specify the NIS domain the client should join and start the ypbnd process afterwards. Add the following lines to /etc/rc.config.d/namesrvs to start the NIS client during boot-up:

```
NIS_CLIENT=1
NIS_DOMAIN=<Name of the NIS-Domain>
```

The startup script /sbin/init.d/nis.client will use this information to start the ypbnd process.

The ypbnd process used to broadcast to find a NIS server within its local subnet. It then bound itself to the first server that answered. If a server from another subnet should be used, some special options like ypset or ypsetme had to be specified (see ypbnd and ypset man-pages; additional entries in /etc/rc.config.d/namesrvs were required). This behavior is documented in every manual covering NIS.

New functionality became available with some patches. Current installations can use the command "ypinit -c". It asks for a list of NIS servers, to be stored in /var/yp/binding/ypservers. If this file exists, the ypbnd process will not use broadcasts to find a NIS server; instead it will contact the listed servers one by one and only use broadcasts in case none of the listed servers answers.

The improved ypinit command is `usr/sbin` located at `/usr/newconfig/`.

NIS Troubleshooting, first steps:

You should check:

Determine to which domain this NIS client is bound

```
# domainname  
  
Determine the server to which the client bound  
# ypwhich  
  
Determine which NIS maps are available  
# ypwhich -m  
  
Display the contents of a map  
# ypcat -k <name of NIS map>  
  
Match a specific entry in a map  
# ypmatch `hostname` hosts
```

The system as a LDAP-UX client

The LDAP-UX client is valuable with other services such as NFS because it has largely taken over the role that NIS has historically played. It provides a way to centrally administer UID and GID information that is vitally important to be consistent throughout the enterprise. A directory schema RFC 2307 has been constructed for LDAP to provide this information for NFS and also for AutoFS.

To verify potential problems with LDAP-UX some commands are useful.

```
# ldapcfinfo -t {passwd|group|pam}  
  
INFO:      CFI_CONFIG_SUCCESS:  
  
        "{passwd|group|pam}" service appears properly configured for LDAP-UX  
operation.  
  
# pwget (no options)  enumerates all user's passwd  
# grget (no options)  enumerates all groups  
# listusers  
# logins  
  
# nsquery passwd ldap20 ldap  
  
Using "ldap" for the passwd policy.  
Searching ldap for ldap20  
User name: ldap20  
User Id: 100020  
Group Id: 200  
Gecos:  
Home Directory: /home/ldap  
Shell: /sbin/sh  
Switch configuration: Terminates Search
```

Well-Known Network Services

A standard HP-UX installation is capable of using several *network services*. Many but not all of these are, or may be, launched from the Internet super daemon, inetd. Those which are not are configured (in HP-UX) to start from configuration scripts in /etc/rc.config.d directory.

The names of the network services defined by the Internet Assigned Numbers Authority (IANA) are provided in /etc/services and may also be served by either NIS or LDAP through their entries through /etc/nsswitch.conf directives.

A TCP-based *network service* will have a LISTEN port; the output of netstat shows the ports offering a TCP service

```
# netstat -a | grep LISTEN
```

tcp	0	0	localhost.49162	*.*	LISTEN
tcp	0	0	*.135	*.*	LISTEN
tcp	0	0	*.2121	*.*	LISTEN
tcp	0	0	*.ovbbccb	*.*	LISTEN
tcp	0	0	localhost.49183	*.*	LISTEN
tcp	0	0	*.shell	*.*	LISTEN
tcp	0	0	*.telnet	*.*	LISTEN
tcp	0	0	*.login	*.*	LISTEN
tcp	0	0	*.daytime	*.*	LISTEN
tcp	0	0	*.exec	*.*	LISTEN
tcp	0	0	*.auth	*.*	LISTEN
tcp	0	0	*.22	*.*	LISTEN
tcp	0	0	*.discard	*.*	LISTEN
tcp	0	0	*.echo	*.*	LISTEN
tcp	0	0	*.time	*.*	LISTEN

The file /etc/services, lists most of the well-known services but the HP-UX distribution does not map ssh in this file so it's port (22) is not listed by name, the same is true of DCE endpoint (135). Otherwise, we can see that the higher port numbers listed are not well-known services because they are listening above port 1023.

For the services started from inetd, the inetd process itself actually owns all of the LISTEN ports. Inetd will launch the service from a library routine upon demand. For instance, /etc/inetd.conf will have an entry such as the following:

```
telnet      stream tcp6 nowait root /usr/lbin/telnetd  telnetd -b /etc/issue
```

Upon receiving a telnet request, inetd will launch /usr/lbin/telnetd to handle the incoming connection.

Some security is provided in HP-UX via /var/adm/inetd.sec:

```
service_name { allow | deny } { hostaddrs | hostnames | netaddrs | netnames }
```

example (only allow loopback telnet connections)

```
telnet allow 127.0.0.1
```

This file applies only to services launched out of inetd (as defined in /etc/inetd.conf), it is not a general security mechanism.

`man inetd.conf` shows the syntax of the file. Lines starting with “#” are comments. `inetd` also requires information from `/etc/services` and `/etc/protocols` or corresponding NIS-maps (see `man nsswitch.conf`) to determine a service and its corresponding port.

Recommended inetd usage:

Changes to the `/etc/inetd.conf` file are only activated after executing

```
# inetd -c      (or after restarting the inetd process)
```

Toggle on/off connection logging of `inetd` services to `/var/adm/syslog/syslog.log`.

```
# inetd -l
```

Turning on/of debugging of the `inetd` (undocumented feature)

```
# inetd -b
```

Stopping the `inetd` process. Do not use `kill -9`

```
# inetd -k
```

For details on the services started by `inetd` (example `telnetd`) refer to their corresponding man pages.

NFS

Basic Functionality

When a client system mounts a filesystem via NFS the following happens:

- The client system generates an RPC request to see if and on which port on the server system a mount daemon (`rpc.mountd`) is reachable (request to the portmapper on port 111).

- The portmapper of the server system communicates the port information to the client where the `rpc.mountd` can be reached.
- The client system sends a mount request for the corresponding file system to the server's mount daemon.
- The mount daemon checks, if the file system can be exported to the client. If so, it then sends the client the file handle of the corresponding file system. Otherwise, it answers with "access denied".
- If the client gets the file handle, it generates an RPC request to gather information about the port on the server systems where the NFS daemon (`nfsd` for UDP communication / `nfsktcpd` for TCP communication) is reachable.
- The portmapper of the server system communicates to the client, which port can be reached by `nfsd` (HP-UX default is port 2049).
- The client system sends an NFS request for the corresponding file handle to the server's NFS daemon.
- The NFS daemon searches for the data belonging to the file handle and sends it to the client.
- The client has mounted the file system. For each additional access to the mounted file system inquiries to the special file handle are sent to the NFS daemon of the server.

All available HP-UX versions support NFS version 2 (NFSv2) and version 3 (NFSv3) with network protocols UDP and TCP (use current patches!). Additionally, HP-UX 11.31 is able to handle NFSv4 communication.

The central configuration file for the NFS functionality is `/etc/rc.config.d/nfsconf` file. It contains the parameters defining basic functionality, and – depending on the version of the operating system and patch level – additional configuration possibilities.

What is configurable on an NFS client?

Well, actually nothing! The kernel of the system must contain the NFS code, but that is always the case nowadays. This delivers basic functionality. For additional functionality the following additional processes are required:

<code>rpc.lockd</code> and <code>rpc.statd</code>	for file locking
<code>biod</code>	I/O Organization
<code>automountd</code>	if you like or need it

The startup script ensures that the required processes are started, if the following settings are

made in `/etc/rc.config.d/nfsconf`:

```
NFS_CLIENT=1
```

The service may be started and stopped using

```
# /sbin/init.d/nfs.client start    or  
# /sbin/init.d/nfs.client stop
```

Enter file systems in `/etc/fstab` to make them available via NFS automatically upon reboot. Temporary ones may be made available using the `automount` daemon.

The file `/etc/rc.config.d/nfsconf` contains additional options to customize the system: Options to start `automountd` and a value for the number of boids (do not change it, if no problems are visible)

```
NUM_NFSIOD=4
```

```
AUTOFS=1
```

What can be configured on the NFS Server?

On the server side is more work to do. We need the following processes:

<code>rpcbind</code>	the portmapper
<code>rpc.mountd</code>	the mount daemon
<code>nfsd</code>	the UDP protocol need more of these processes. Rule of thumb: 4 nfsds per CPU and at least 32 in total
<code>nfsktcpd</code>	it handles the TCP-based NFS communication; it will start up to 10 kernel threads per client, if this is required to speed up performance

Additional functionality requires additional daemons.

<code>rpc.statd</code>	for file locking
<code>rpc.lockd</code>	for file locking
<code>rpc.pcnfsd</code>	to handle PCNFSD protocol requests

The line

```
NFS_SERVER=1
```

in the file `/etc/rc.config.d/nfsconf` allows us to start and stop the daemons processes using

```
# /sbin/init.d/nfs.server start  
# /sbin/init.d/nfs.server stop
```

It also ensures that the NFS server gets started during the system boot procedure.

Please check `/etc/rc.config.d/nfsconf` for additional options to customize the system

behavior.

Performance discussions show how important it is to adjust `NUM_NFSD=<number>`, if the UDP protocol is used for NFS communication.

It determines the number of `nfsd` processes, which handle UDP-NFS requests –the main factor of all NFS related load today. The number should be greater than 4 times the number of available CPUs; and if the system is an NFS client of itself, at least greater than 32.

```
PCNFS_SERVER=1
```

is only required if there are DOS or Windows clients, which use the special PCNFSD protocol.

You also need to maintain the `/etc/exports` (*/etc/dfs/dfstab in HP-UX 11.31 and above systems*) file. This is the location to enter the file systems to be made available to NFS clients.

The “`exportfs -a`” command triggers the `rpc.mountd` to read this file and make the file systems accessible. The `exportfs` command going forward will be replaced by the `showmount` command starting at HP-UX 11.31 (transitional support is still available for using the `exportfs` command), when called without any option, shows what the `mountd` has currently exported (see man pages). To show what the NFS server has exported use:

```
# exportfs -v  
# share          (HP-UX 11.31 and above)
```

As a NFS client to show file systems available from any NFS server use:

```
# showmount -e hostname
```

Warning: File systems managed by a Serviceguard package and available via NFS usually do not appear in `/etc/exports`, because the package startup performs the export via the “`exportfs -i`” command in its package startup script.

Involved Scripts and Processes

Client	Server
Administration Scripts:	
/etc/rc.config.d/nfsconf	/etc/rc.config.d/nfsconf
/sbin/init.d/nfs.client [stop start]	/sbin/init.d/nfs.server[stop start]
Worker Processes:	
	rpcbind
	rpc.mountd
biod (a sufficient number)	nfsd / nfsktcpd
automount (d) (if required)	
File Locking Processes:	
rpcbind	rpcbind
rpc.lockd	rpc.lockd
rpc.statd	rpc.statd

Information required for troubleshooting

The following information is required to start troubleshooting.

On the Client	On the Server
OS Version & Patch level	OS Version & Patch level
Name resolution for the server and client (Name after IP and IP after Name)	Name resolution for the server and client (Name after IP and IP after Name)
File system structure (what is mounted where?)	File system structure (what is mounted where?)
Status and structure of the mount points	Status and structure of the mount points
nettl-trace of the problem event	nettl-trace of the problem event
showmount -e <server>	exportfs
nfsstat -m	
nfsstat -c	nfsstat -s
netstat -s	netstat -s
mount -v	mount -v
ping -o <server> -n 3	ping -o <client> -n 3
rpcinfo -p <server>	
rpcinfo -u <server> mountd	
rpcinfo -u <server> nfs	
rpcinfo -t <server> nfs	
ping <server> 8192	ping <client> 8192

Most of the above information can be collected using an internal HP support tool, [nfsinfo](#). This tool is HP's intellectual property and is proprietary. It should be used outside of HP only in the context of a support case. This tool does not perform a nettl trace which needs to be run while recreating the problem. The basic techniques involved in nettl tracing are covered in the previous chapter of this handbook.

nfsinfo can be found at the following link:

<http://teams3.sharepoint.hp.com/teams/esssupport/InsideESSSupport/InsideWTEC/NETUX/Pages/Tools.aspx#onc> (HP internal)

Common errors include:

nfs server ... not responding - The answer from the server does not reach the client in time. It has to be checked, whether the request reaches the server, or the server is unable to handle the request, to organize the data and to send an answer back to the client, or the answer reaches the client and the client is unable to handle it.

... access denied- rpc.mountd does not allow the request from the specific client. On the client use “showmount -e <server name>” to check which file systems are exported by the server. Login to server from this client to check how the server recognizes the client with “who - R am I”

AutoFS

The automountd daemon implements AutoFS and replaces the legacy automount daemon. The automount program is used to install automatic mount points. The intent of AutoFS is to make mounting of NFS file systems more flexible. Configuration file /etc/rc.config.d/nfsconf:

```
AUTOFS=1
```

The central configuration file for the daemon is:

```
/etc/auto_master
```

This file contains the information or directs to other alternative files – *maps* – with the information regarding the file systems to be mounted.

Useful hints:

- With NIS configuration becoming increasingly rare, Enhanced AutoFS in conjunction with LDAP-UX may contain the maps used by automountd which was previously commonly implemented via NIS.
- Straightforward configuration guidance can be found in the book *Managing NFS/NIS* by Hal Stern.
- NFS manuals for HP-UX can be found by referring to the HP-UX Networking Software for the appropriate level of HP-UX that is being worked with.
<http://h20000.www2.hp.com/bizsupport/TechSupport/Product.jsp?lang=en&cc=us&askId=101&prodClassId=10008&contentType=SupportManual&docIndexId=64255&prodTypeId=18964&prodCatId=427973&prodSubCatId=4155222>
- Systems after HP-UX 11iv1 do not offer the legacy automount daemon; however, this version offers both. The *AUTOFs* automount daemon is mandatory if “newer” NFS-features are used (NFSv3, NFSv4, Files >2GB).
- Patching for NFS has gone to web releases which can be obtained from
<https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=ONCplus> which is a link off the Software Depot home <http://software.hp.com>.
- If troubleshooting a AutoFS problem it is often useful to create a mount point directory and attempt to mount the NFS server’s exported filesystem to isolate whether the problem is NFS or AutoFS. If the problem is with automountd then debug logging of that process and a nettl trace are very useful. Output from running the nfsinfo support script should be run after the AutoFS debug logging.

WARNING: Never use “`kill -9`” on an `automountd` daemon! Because the daemon administers information for the kernel it will corrupt the entire system and to cleanup such a situation a reboot may be required.

What information can we get from `automountd`?

You can trace a problem by turning on process logging. It can be enabled by a start option, or in the `nfsconf` file `AUTOMOUNTD_OPTIONS="-T"`, or using

```
# kill -SIGUSR2 <process id>      or
# cd /net/=<debug level>
```

(the later only if `/net` is in place for the special map `/net -hosts`). The information will be written to `/var/adm/automount.log`.

You can turn off logging via

```
# cd /net/=0                      or again with
# kill -SIGUSR2 <process id>
```

If the automountd process no longer responds to the SIGUSR2 signal, it most often will also not respond to the SIGTERM signal. In this case, there is no other way to fully restore the functionality of the daemon than to **reboot**.

Commonly used network services

telnet

The telnet service has historically been the application that provides a pseudo-term (which is a substitute for a hard-wired console interface) for a login account to use to access a server. This service is quickly being replaced by ssh because of the lack of security with telnet. The telnet user is given only a cursory inspection of its identity (forward and reverse lookup of incoming IP address to prevent spoofing) and it provides no privacy since everything, including, the clear text password can be picked up with a network trace.

It is launched via the inetd process and is configured via /etc/inetd.conf

```
telnet    stream tcp6 nowait root /usr/lbin/telnetd  telnetd
```

Telnet is well-known service:

```
telnet    23/tcp          # Virtual Terminal Protocol
```

Common problems include:

- Connection refused – Usually there is no inetd running to accept the connection request
- telnet: Unable to connect to the remote host: Connection timed out – Usually the telnet request is not reaching the destination telnet server. Could be routing or the address does not really exist.
- Long pause between supplying account name at `login:` prompt and being prompted for `Password:` - This is a hostname resolution issue on the telnetd server of getting a name service response back. Check for forward and reverse resolution of the telnet client's hostname and IP address.
- Long pause after providing a password and getting a command prompt – Likely an issue of large or corrupt utmp file or something wrong in the login user's profile.

ftp

The ftp service has historically been the file transfer service of choice. While still popular, this service is not well suited for environments where security is a big concern. The ftp client (host) is given only a cursory inspection of its identity (forward and reverse lookup of incoming IP address to prevent spoofing) and it provides no privacy since everything, including, the clear text password can be picked up with a network trace.

It is launched via the inetd process and is configured via /etc/inetd.conf

```
ftp      stream tcp6 nowait root /usr/lbin/ftpd    ftpd -l
```

One special characteristic of ftp is that it requires two TCP ports whose well-known service numbers are:

```
ftp-data  20/tcp          # File Transfer Protocol (Data)
ftp       21/tcp          # File Transfer Protocol (Control)
```

Port 20 is the control connection

Another unique feature of ftp is that it can be configured for anonymous access which means that anyone is permitted to pick up files placed in public directories. The ftp server is configured to constrain file system access for the anonymous user to only those public directories.

Common problems include:

- Performance - Could be file system performance or the network infrastructure. It is useful to test to eliminate file system performance issues. Check file transfer time versus transferring a similar amount of bytes using no file systems.

The example below is sending one million full TCP segments which means the Ethernet frame will be at capacity without performing IO operations on the ftp client or the ftp server. The count field here should be adjusted to the size of the file being sent for testing purposes. If this test eliminates network performance file system patches and known problems should be examined. If the problem is network, data should be gathered via support tools such as [linkinfo](#) and [lanshow](#). (HP internal only [download lan link](#) tools by OS level)

```
# ftp <target host>
ftp> put "|dd if=/dev/zero bs=1500 count=1000000" /dev/null
200 PORT command successful.
150 Opening BINARY mode data connection for /dev/null.
1000000+0 records in
1000000+0 records out
226 Transfer complete.
```

1500000000 bytes sent in 13.91 seconds (105324.47 Kbytes/s)

- Firewall issues- Once a control connection is established to ftp on the server the issue, ftp can operate in either ACTIVE mode or PASSIVE mode. In ACTIVE mode ftp client firewalls will be asked to allow an incoming connection coming from a privileged port (20). In a PASSIVE connection the ftp client asks for anonymous ports on the ftp server. To understand the implications, there is a good tutorial on an external site:

<http://slacksite.com/other/ftp.html> (external link)

- Restricting user access – There are many ways provided to allow/disallow what an ftp client user login or anonymous user can do once logged in. The mechanism for configuring this is documented in the `ftpaccess` man page. This is advanced configuration which will not be covered in this document.

r-commands

The Berkley distribution is the source for the commands rexec, rlogin, and remsh. The client sending the r-command is given only a cursory inspection of its identity (forward and reverse lookup of incoming IP address to prevent spoofing) and it provides no privacy since everything, including, the clear text password can be picked up with a network trace. There are Kerberos versions of klogin and kshell which will not be covered in this document.

They are launched via the inetd process and are configured via /etc/inetd.conf

```
login    stream tcp6 nowait root /usr/lbin/rlogind  rlogind -B /etc/issue
shell   stream tcp6 nowait root /usr/lbin/remshd  remshd
exec    stream tcp6 nowait root /usr/lbin/rexecd  rexecd
```

These are well-known services:

```
exec    512/tcp        # remote execution, passwd required
login    513/tcp        # remote login
shell   514/tcp cmd    # remote command, no passwd used
```

The exec service is mentioned here only since it is not widely used. The main attraction of the r-commands is the ability to login without providing a password. This is provided by means of establishing host equivalency via `hosts.equiv` file for non-privileged users or via providing `.rhosts` in the home directory of the desired root or non-privileged user.

In either `/etc/hosts.equiv` or `$HOME/.rhosts` the format is the same.

```
[hostname] [username] [#comment]
```

Common problems include:

- Unexpected prompting for password with rlogin- Host equivalency is not properly established because hostname resolution does not resolve to a hostname which matches the /etc/hosts.equiv or \$HOME/.rhosts file entry.
- remshd: Login incorrect. – Host equivalency is not properly established.

From rlogin/remsh client login to server (providing password) and enter command `who -mR`. This produces a display similar to:

```
bob pts/0 Aug 28 06:49 (rxg16u07.example.com)
```

```
# grep rxg16u07 /etc/hosts.equiv or $HOME/.rhosts
```

There must be an exact match.

```
rxg16u07.example.com bob
```

- remsh connection timed out - The remshd process must establish a connection back to the remsh client to pass stdout from the command sent. A firewall may be blocking the establishment of this connection.

ssh

The ssh service is very popular since it provides strong identity verification of the incoming client connection (the IP is whom it claims to be and not a “man in the middle”), integrity of the data passed (it has not been altered), and privacy (logins and session information are encrypted using strong security mechanisms). It can be configured to allow for login without providing a password and when configured this way can be substituted for the r-commands. The ssh service provides the file transfer capabilities sftp and scp.

The ssh service is launched through the script `/sbin/init.d/secsh` [start/stop]. The sshd daemon forks off sshd processes to handle incoming connection requests.

The ssh service is not mapped in the HP-UX distributed /etc/services file, however, it is also a well-known service which operates on port 22.

It offers several authentication mechanisms: Password, Public-Key, Kerberos, Host-Based, Keyboard-Interactive, and User-Specific.

The two most common, Password and Public-Key, will be considered in this document.

Password authentication is simple and should work with no configuration set up. Although a password is sent across the network it is not sent in clear text as in the case of telnet or rlogin or ftp.

Public-Key authentication is set up by means of generating a mathematically related Private/Public key pair on the ssh client. The client must keep the Private key secret but the Public key can be freely distributed.

The scenario goes like this: the ssh client sets up a key pair in the \$HOME/.ssh subdirectory of the user who wants login access to the sshd server using the ssh-keygen command and then transfers the Public key to the \$HOME/.ssh directory of the desired user and then appends the it to the authorized_keys file.

The user is prompted for the location to save the Private/Public key pair, and then the user must decide whether to provide a passphrase. If none is given here, all future authentication is without password provided. This makes ssh suitable for secure non-interactive file transfers and tasks; however, there it is slightly less secure than providing the passphrase and changing it every so often.

```
# ssh-keygen
Generating public/private rsa key pair.
Please be patient.... Key generation may take a few minutes
Enter file in which to save the key (//.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in //..ssh/id_rsa.
Your public key has been saved in //..ssh/id_rsa.pub.
The key fingerprint is:
60:b4:22:fa:1e:03:ff:20:9d:12:4e:99:e1:82:b7:bc root@host
The key's randomart image is:
+-[ RSA 2048]----+
|   .   |
|   ..  |
| ...+  |
|o+. o .|
|*=. S   |
|+O o    |
|o.@     |
| + *    |
| E .    |
|          |
```

Although not commonly implemented, the ssh-agent program can be launched at the beginning of a login session to provide a passphrase only once. This permits the use of passphrase but eliminates some of the inconvenience of having a passphrase. The use of the ssh-agent is not discussed in this document.

Common problems include:

- Uncertainty whether the ssh server is the right host- How does the client know they are logging into the authentic host? On initial login, a RSA (or DSA) key fingerprint is presented. This should correspond to the host key for the desired machine.

```
# ssh avalon
```

The authenticity of host ‘avalon (10.90.90.90)’ can’t be established.

RSA key fingerprint is 71:94:55:39:c3:f2:01:2c:16:4c:78:15:b8:75:85:46.

Are you sure you want to continue connecting (yes/no)?

If the server’s host public key is available, this can be verified with ssh-keygen:

```
# ssh-keygen -l -f /etc/opt/ssh/ssh_host_rsa_key.pub
2048 71:94:55:39:c3:f2:01:2c:16:4c:78:15:b8:75:85:46 /etc/opt/ssh/ssh_host_rsa_key.pub (RSA)
```

- Can’t log in- There are many potential causes of this. If the authentication is Public-Key the first thing to verify is the permissions of both the ssh server and client. If these permissions are not restrictive enough, permission could be denied.

Permissions for the Client Files and Directories

File/Directory	Permissions
\$HOME (home directory)	drwx----- or drwxr--r--
\$HOME/.ssh	drwx----- or drwxr--r--
\$HOME/.ssh/id_rsa and id_dsa	-rw-r--r-- or -rw-----
\$HOME/.ssh/id_rsa.pub and id_dsa.pub	-rw-r--r-- or -rw-----
\$HOME/.ssh/config	-rwx-----

Permissions for the Server Files and Directories

File/Directory	File Permission
\$HOME (home directory)	drwx----- or drwxr--r---
\$HOME/.ssh	drwx----- or drwxr--r---
\$HOME/.ssh/authorized_keys and \$HOME/.ssh/authorized_keys2	-rw-r--r-- or -rw-----

- Still can't log in - Further debug logging will be required. A first pass at debug logging of the sshd server (the client does not control whether it is authenticated, the server must determine) can be taken as follows:

```
server# script /tmp/server.out
Script started, file is /tmp/server.out
server# /usr/sbin/sshd -ddd -p 2222 -e
```

...then from the ssh client, recreate login problem:

```
client# script /tmp/client.out
Script started, file is /tmp/client.out
client# ssh -p 2222 -vvv username@server
client# exit
Script done, file is /tmp/client.out
```

```
server# exit  
Script done, file is /tmp/server.out
```

If this does not reveal the source of the problem more intense analysis should be done using `tusc` and `sshinfo` support script. A support case should be logged.

- Client receives warning:

WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!

.....

Someone could be eavesdropping on you right now (man-in-the-middle attack)!

It is also possible that the RSA host key has just been changed.

The fingerprint for the RSA key sent by the remote host is 91:a7:55:1b:c6:0b:e7:29:6c:f0:6c:42:c5:bd:36:7c.

Please contact your system administrator.

Add correct host key in **/.ssh/known_hosts** to get rid of this message.

Offending key in **/.ssh/known_hosts:59**

RSA host key for MyRemoteSSHDHostname has changed and you have requested strict checking.

Host key verification failed.

Need to verify the cached server host key on the client and determine if there is a legitimate reason the host key on the server changed.

```
client# ssh-keygen -F avalon -f known_hosts
# Host avalon found: line 23 type RSA
avalon ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAqDzswemRwP6nmIyIc5fNfqOQgtS4Shvx560XiICT
MXjJhND9FjMSciOG6zspoKTm5+xWvcZwf0xtZcvlegQTpH2nirZD2EqTmhxnkK3MQcGv
FL/MpB4hFvME9AnMBjG1XnW5Qm0CR5f0gP4cH9caK0gStBKD9NFwyxUe6z2TVOIBuX8
4XY1i7jem/QCYJSIS/AnwWNw0EhXtBN+MXTasm741qt3bq0E3QVpGtz18emL7/nT9EEgpe
E8bFH4ucJx2b59O3tceYPk4EQG5sddvXtrC3euu7dEFNxyQU3+tooeYwYa3p5RreKnYD++vB
0PMYGh9sEgTWCGHVu3ROay0Q==
```

```
client# grep avalon known_hosts > avalononly
client # ssh-keygen -l -f avalononly
2048 71:94:55:39:c3:f2:01:2c:16:4c:78:15:b8:75:85:46 avalon,10.226.90.43 (RSA)
```

Additional Information

General Knowledge

- *TCP/IP Network Administration* by C. Hunt (O'Reilly)
- *Managing NFS and NIS* by H. Stern (O'Reilly)
- *DNS and BIND* by Albitz/Liu (O'Reilly)

Internet Standards – RFC

- <http://www.rfc-editor.org> (non HP)
- <http://www.isc.org> (non HP)