Network Information Service (NIS) B.11.31.02 Administrator's Guide

HP-UX 11i v3

Table of Contents

List of Figures

List of Tables

About This Document

This document describes how to configure and administer Network Information Service (NIS ™) on systems running HP-UX 11i v3. It also describes how to troubleshoot NIS.

Intended Audience

This document is intended for system administrators who install and configure UNIX® systems using the Open Network Computing (ONC) suite of products. Readers are expected to be familiar with the following:

- System administration concepts
- UNIX operating system concepts
- Networking concepts

Document Organization

The Network Information Service (NIS) Administrator's Guide is organized as follows:

Typographic Conventions

This document uses the following typographic conventions:

monospace	Computer output, files, directories, software elements such as command options, function names, and parameters.
	Read tunables from the /etc/vx/tunefstab file.
italic	New terms, book titles, emphasis, and variables replaced with a name or value.
%	C shell prompt
$	Bourne/Korn shell prompt
#	Superuser prompt (all shells)
\	Continued input on the following line; you do not type this character.
[ ]	In command synopsis, brackets indicate an optional argument.
	`ls [ -a ]`
		In command synopsis, a vertical bar separates mutually exclusive arguments.
	`mount [suid	nosuid]`

`Ctrl+A`	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the plus.

Related Information

Additional information about NIS is available at:

`http://docs.hp.com`

This website contains the following documents about NIS:

- *HP-UX 11i v3 Release Notes* (5992-1996), September 2007.
- *HP-UX 11i v3 Release Notes* (5991-6469), February 2007.
- *ONC+ Release Notes* (5991-8714), March 2007.
- *ONCplus B.11.31.01 Release Notes* (5992-2334), September 2007.
- *NFS Services Administrator's Guide* (B1031-90061), March 2007.

HP Welcomes Your Comments

HP welcomes your comments concerning this document. We are committed to providing documentation that meets your needs.

Send your comments or suggestions to: netinfo_feedback@cup.hp.com

Include the document title, manufacturing part number, and any comment on the error found in this document. Also, include what we did right, so we can incorporate it into other documents.

1 Introduction

This chapter introduces you to Network Information Service (NIS) and its features. It also discusses the components in an NIS network.

This chapter addresses the following topics:

Overview

NIS is a client/server directory service protocol that enables you to maintain a centralized repository of system information, such as hosts, users, passwords, and other user-defined information.

NOTE: NIS is not supported across WAN links (such as X.25 and SLIP).

NIS Network Components

This section discusses the components in an NIS network.

The NIS network consists of the following components:

- NIS Master Server
- NIS Slave Server
- NIS Client

NIS Master Server

An NIS master server holds the source files for all the NIS maps in the domain. Any changes to the NIS maps must be made on the NIS master server. The NIS master server delivers information to the NIS clients and supplies the NIS slave servers with up-to-date maps. The NIS maps are stored on disk in the NIS master server. An NIS master server can also be an NIS client.

NIS Slave Server

An NIS slave server provides information to the NIS clients and takes the load off the NIS master server. The NIS maps are created on the NIS master server and then transferred to the slave servers. On the slave servers, the maps are stored in the memory. Changes to the NIS maps must be made on the NIS master server, which then transfers the updated copies of the maps to the NIS slave servers.

NIS Client

An NIS client is a system that obtains its configuration information from either an NIS master server or an NIS slave server.

You can configure the NIS client using the `ypinit-c` command. The `ypinit -c` command configures the local host as an NIS client to bind the NIS client to a particular NIS server. When you invoke the `ypinit -c` command, the system prompts you to construct a list of NIS servers in the order of preference. The client tries to bind to the NIS servers in the order specified in the `/var/yp/binding/<domain_name>/ypservers` file.

A Sample NIS Network

Figure 1-1 represents a typical NIS network consisting of the NIS master, the NIS slave server, and clients.

Figure 1-1 Illustration of a Sample NIS Network
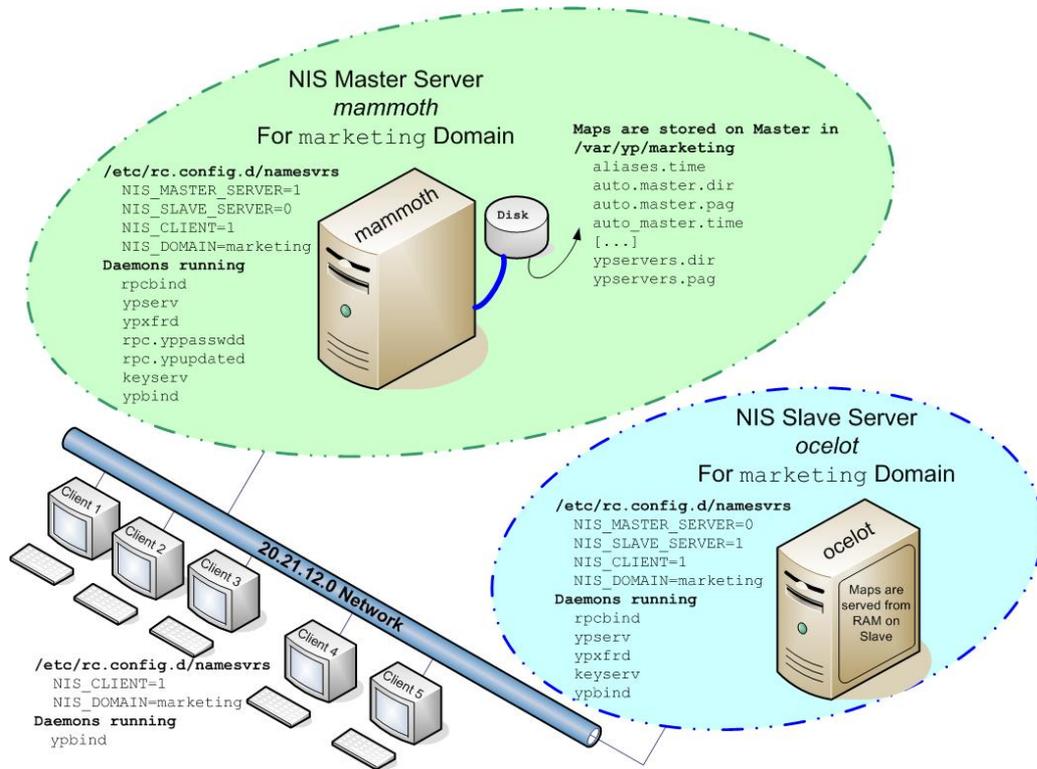
The host name of the NIS master is `mammoth` and the host name of the slave server is `ocelot`. Both the master and the slave servers reside in the `marketing` domain. The client systems `Client 1`, `Client 2`, `Client 3`, `Client 4`, and `Client 5` connect to the master and slave servers to obtain NIS configuration information over the `20.21.12.0` network. The entries in the `/etc/rc.config.d/namesvrs` file on both the master and slave servers are shown in Figure 1-1. The daemons that run on the NIS master server are `rpcbind`, `ypserv`, `ypxfrd`, `rpc.yppasswdd`, `rpc.ypupdated`, `keyserv`, and `ypbind`. The daemons that run on the NIS slave server are `rpcbind`, `ypserv`, `ypxfrd`, `keyserv`, and `ypbind`. Only the `ypbind` daemon runs on the client because the client uses only the `ypbind` daemon to bind to the NIS master server.

NIS Naming Service Components

The NIS naming service consists of the following components:

- NIS Maps
- NIS Domains
- NIS Daemons
- NIS Commands
- Default NIS Source Files

NIS Maps

NIS maps are databases that specify certain system information, such as user names, passwords, and host names, in a database format called `dbm`. The NIS maps are located in the `/var/yp/domainname/` directory on the NIS server. A list of all the default NIS maps is available in the `/etc/rc.config.d/namesvrs` file.

NIS Domains

An NIS domain is a collection of systems that share a common set of NIS maps. Each domain has a domain name and the systems that share a common set of maps belong to the same domain.

Any system can belong to a given domain as long as there is a server for that domain's maps in a reachable network. An NIS client system obtains its domain name and binds to an NIS server as part of its boot process.

NIS Daemons

Table 1-1 lists the daemons that provide the NIS service.

Table 1-1 NIS Daemons

NIS Daemon	Description
ypxfrd	Copies the NIS maps to the local host from an NIS server.
ypserv	Maintains the NIS database called maps, and services the NIS client requests.
ypbind	Maintains binding information for the hosts.
yppasswdd	Changes the password associated with a given login name.
ypupdated	Updates information in the NIS database.

NIS Commands

Table 1-2 lists the commands that NIS supports.

Table 1-2 NIS Commands

Command	Description
domainname	Sets or displays the name of the NIS domain. For more information, see *domainname* (1).
makedbm	Generates an NIS map from an ASCII input file. For more information, see *makedbm* (1M).
ypcat	Prints all the values in an NIS map. For more information, see *ypcat* (1).
ypinit	Sets up an NIS master server, an NIS slave server, or an NIS client. For more information, see *ypinit* (1M).
ypmake	Generates one or more NIS maps from the ASCII files and optionally transfers them to the NIS slave servers. The /var/yp/Makefile and *make* (1) commands support the same functionality. For more information, see *ypmake* (1M).
ypmatch	Prints the values associated with one or more selected keys in an NIS map. For more information, see *ypmatch* (1).
yppasswd	Changes a login password that is stored in the NIS passwd map. For more information, see *yppasswd* (1).
yppoll	Returns the name of the master server for an NIS map and the time at which the map was built. For more information, see *yppoll* (1M).
yppush	Copies an NIS map from the NIS master server to each of the NIS slave servers. The yppush command constructs a list of NIS server host names by reading the NIS map ypservers within the domain. Keys within the ypservers map are the host names of the machines on which the NIS servers run. yppush then sends a "transfer map" request to the NIS server at each host, along with the information needed by the transfer agent (the program that actually moves the map) to call back yppush. See *ypmake*(1M) and *ypinit*(1M) for more information.
ypset	Instructs an NIS client process (ypbind) to bind to a specified NIS server. The ypset command can be used only if ypbind is invoked with the -ypset option. For more information, see *ypset* (1M).

Table 1-2 NIS Commands *(continued)*

Command	Description
ypwhich	Returns the name of the NIS server for the local client, or the name of the NIS master server for one or more NIS maps. For more information, see *ypwhich* (1).
ypxfr	Transfers one or more NIS maps from a master server to the local slave server. A slave server calls ypxfr when yppush is executed on the master server. For more information, see *ypxfr* (1M).

Default NIS Source Files

Table 1-3 lists the source files that NIS uses by default for constructing the NIS maps.

Table 1-3 NIS Source Files

NIS Source File	Description
/etc/hosts	A file that maps Internet addresses to the host names. See *nsswitch.conf* (4) or *hosts*(4).
/etc/passwd	A list of the users on the system, along with the associated passwords, home directories, and other information. See *passwd*(4).
/etc/group	A list of group names, along with encrypted password and numerical group ID. See *group*(4).
/etc/services	A file that associates network services with the port numbers and protocols the services use. See *services*(4).
/etc/protocols	A file that associates network protocols with protocol numbers. See *protocols*(4).
/etc/networks	A file that associates IP addresses with official network names and aliases. See *networks*(4).
/etc/mail/aliases	A list of Sendmail aliases. See *aliases*(5).
/etc/netid	A list of secure remote procedure calls' (RPC) netnames (unix.*UID@domainname* or unix.*hostname@domainname*) for users and hosts outside the NIS domain.

The information in these files is stored in the NIS databases automatically when you configure an NIS master server. Other user-defined files can also be used with NIS to customize configuration.

NIS Binding

In the NIS domain, associating a server with a particular client is called binding.

The ypbind daemon stores information that allows the client processes on the system to communicate with the ypserv process. The ypbind daemon must run on every machine using NIS services, both NIS servers and clients. The ypserv daemon may or may not be running on an NIS client machine, but it must be running somewhere on the network or be available through a gateway.

The ypinit command allows the user to specify a list of servers (in the order of preference) to which the client can bind. Invoking the ypinit command with the -c option prompts the user to construct a list of NIS servers, in the order of preference, to which the client will try to bind. This list of NIS servers is stored in the /var/yp/binding/<domain_name>/ypservers file.

Information Flow in an NIS Network

Figure 1-2 illustrates the flow of information in an NIS network with a single domain.

Figure 1-2 Flow of Information in an NIS Domain

Master Server

maps → Slave Server

maps → Slave Server

Maps are created from configuration files on the master server

Maps are transferred from the master server to the slave servers

data → Client

data → Client

data → Client

Slave Server sends configuration data to the clients

Master Server can also send configuration data directly to the clients
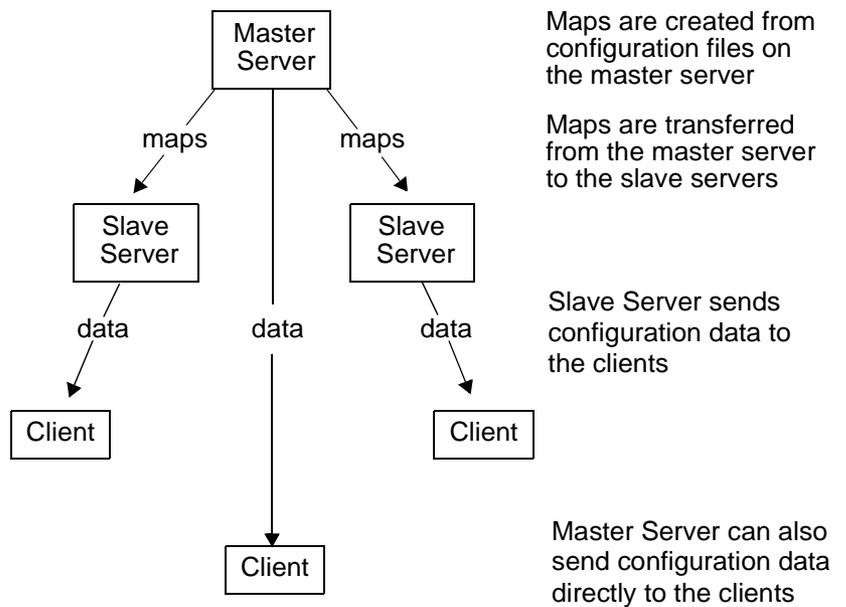
The flow of NIS information from the master server to the client includes the following steps:

1. The NIS maps are created on the master server.
2. The NIS maps are propagated from the master server to the slave server.

NOTE: A host cannot be the master server for more than one NIS domain. However, a master server for one domain can be a slave server for another domain. A host can be a slave server for multiple domains. A client can belong only to one domain.

Figure 1-3 illustrates an NIS network with servers that serve multiple domains.

Figure 1-3 Servers Serving Multiple NIS Domains

Master Server

Master Server | Slave Server

Slave Server

Slave Server

Master Server | Slave Server

Client Client Client Client Client Client Client

☐ Master, Slave, or Client for Domain A

▨ Master, Slave, or Client for Domain B
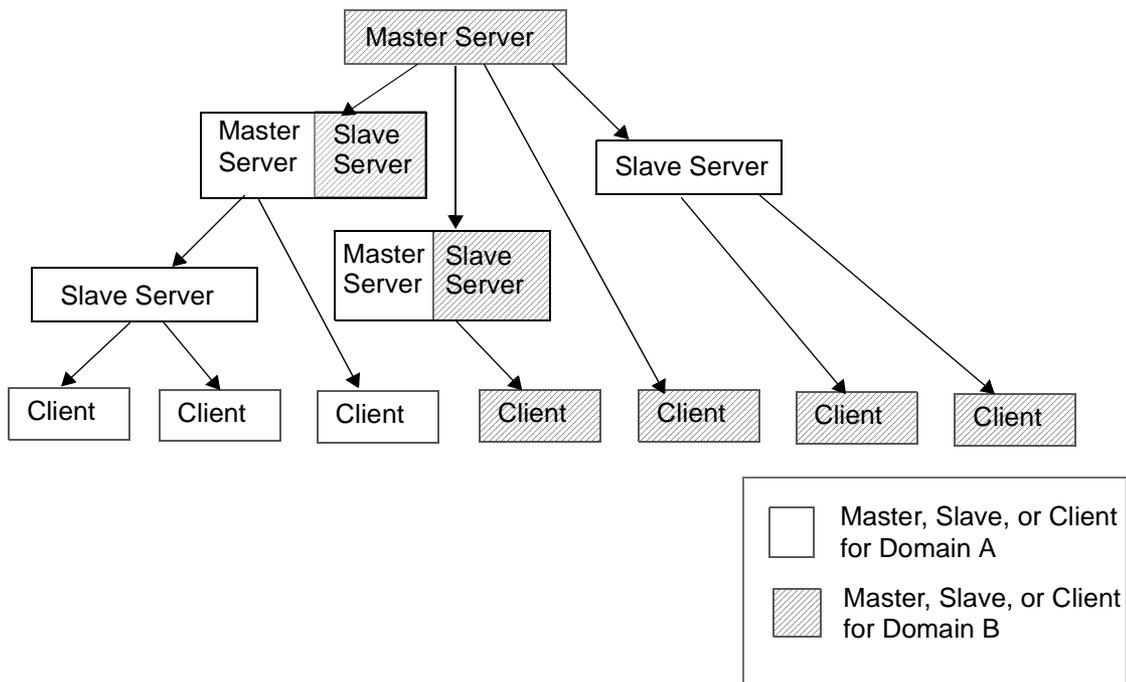
Startup Scripts in NIS Services

This section discusses the startup sequence for the NIS daemons during a normal system boot.

Following are the NIS startup scripts:

- `/sbin/init.d/nis.server`
- `/sbin/init.d/nis.client`

These scripts can be used to start and stop NIS on a system. They also read the `/etc/rc.config.d/namesvrs` file to determine the daemons that must be started, and detect whether the system is a server or a client.

The `nis.server` script is executed at system run level 2 on the NIS server. This script performs the following tasks:

- Starts `rpcbind` if it is not already running.
- Sets the `domainname`.
- Starts the `ypserv` daemon if it is not already running on both the master and the slave servers.
- Starts `ypxfrd` if it is not already running on both the master and the slave servers.
- Starts `yppasswdd` if it is not already running on the master server.
- Starts `ypupdated` if it is not already running on the master server.
- Starts `keyserv` if it is not already running on the master server.

The `nis.client` script is executed at run level 2 on the systems configured as NIS client. This script performs the following tasks:

- Starts the `rpcbind` daemon if it is not already running.
- Sets the `domainname`.
- Starts `ypbind` if it is not already running.
- Starts `keyserv` if it is not already running.

Features in NIS

This section discusses the new NIS features supported on HP-UX 11i v3:

- DNS Forwarding Mode

  DNS forwarding mode can now be enabled through a new option `-d` to the NIS server daemon, `ypserv`. When you run the NIS server in the DNS forwarding mode, if `ypserv` does not find the host or ipnode entry in the NIS host or ipnode database, `ypserv` automatically forwards the host or ipnode queries to DNS. This feature is useful in configurations where certain host or ipnode entries are present in the NIS repository and a different set of entries are present in the DNS repository. For example, if you store the host information for host X in NIS and host Y in DNS. If the `hosts` entry in the `/etc/nsswitch.conf` contains only `nis` (`hosts: nis`), the information for host Y is retrieved even though `dns` is not specified in the `hosts` entry.

  NIS automatically fetches the information from DNS and this operation is transparent to the user or the application. You can achieve this tranparency without changing the `/etc/nsswitch.conf` configuration file.

- Shadow Password Mode

  The NIS password daemon, `rpc.yppasswdd`, now supports the HP-UX shadow password mode. In shadow password mode, the HP-UX system maintains the user passwords in a different file called `shadow,` and replaces the encrypted password in the `passwd` file with `x`. This enhances the password security of the system. You can convert a system to shadow mode using the command `pwconv`, and revert the system using the command `pwunconv`. For more information, see *pwconv*(1M) and *pwunconv* (1M).

The NIS subsystem can now interpret the passwd and shadow files when the system is in shadow mode, that is, the NIS subsystem can recognize a system running in shadow mode, and build, store, and retrieve the password information accordingly.

NOTE: The encrypted password information used for creating NIS password maps is present in the shadow file and is visible in the passwd maps (passwd.byname and passwd.byuid).

To enable this feature, a variable, SHADOW_MODE, is introduced in the /etc/rc.config.d/namesvrs file. The SWFILE option is introduced in the ypmake command to indicate the shadow file. The default value for SWFILE is /etc/shadow. For more information, see *ypmake* (1M).

- Multi-Homed Node

  A multi-homed node is a node that has multiple network interfaces configured. The NIS server can index hosts with multiple entries in the hosts and ipnodes database. While building the NIS maps, ypmake identifies multiple entries with the same hostname, and accommodates the entries in the maps with the aid of the multi and multi.awk scripts. If a host has multiple entries in the database and the ipnodes map is queried, then all the entries are returned. Similarly, if the hosts map is queried, then the address nearest to the NIS client is returned. In previous versions of HP-UX, only the first address listed for a host was built into the NIS maps.

- Support for IPv6 Protocol

  The NIS clients and servers support the IPv6 protocol. Therefore, you can set up an NIS master server, a slave server, or an NIS client that can be identified using an IPv6 address.

  The maps that store host information for NIS are hosts and ipnodes. The /etc/hosts file contains both IPv4 and IPv6 information. When NIS maps are built, NIS filters the IPv4 information from the /etc/hosts file to build the hosts maps that contain only IPv4 information. It also filters /etc/hosts to build the ipnodes maps, such as ipnodes.byaddr and ipnodes.byname, that contain both IPv4 and IPv6 information. Ipnodes are a superset of hosts and act as the hosts database for IPv6 information.

- Alternate Directory for the passwd File

  On HP-UX 11i v3, users may now specify a directory other than the default directory /etc where NIS must look for the passwd file. If the system is running in shadow mode, the shadow file must be present in the same alternate directory specified as the passwd file.

  This feature enhances the security of the system as the administrator can specify a directory for passwd and shadow files which is known to the administrator only.

  Specifying a directory where the passwd and shadow files can be found is mandatory when starting the NIS passwd daemon rpc.yppasswdd on HP-UX 11i v3. The default value is /etc. A new option –D has been introduced to rpc.yppasswdd for this feature in HP-UX 11i v3.

- Long Username and Long Hostname support

  On HP-UX 11i v3, the usernames for NIS users can be up to 256 characters long. The hostnames can be up to 256 characters long for:
  - The hosts on which NIS is running
  - The entries stored and accessed through the hosts and ipnodes maps

- Support for NIS ypbind v3 Protocol

  The NIS client supports the v3 version of the ypbind protocol. This ypbind protocol supports the Transport Independent (TI) RPC.

The ypbind protocol version v1 is obsolete and any request from the NIS client that specifically requests to use the ypbind protocol version v1 is rejected.

- Files for Resolving Map Nicknames

  The NIS commands, ypcat and ypmatch, use the /var/yp/nicknames file to resolve the map nicknames.

  Following are the contents of the /var/yp/nicknames file:

  ```
  passwd passwd.byname
  group group.byname
  networks networks.byaddr
  hosts hosts.byname
  ipnodes ipnodes.byname
  protocols protocols.bynumber
  services services.byname
  aliases mail.aliases
  ethers ethers.byname
  ```

  You can change the nicknames of the most commonly used maps by modifying the fields in the respective /var/yp/nicknames files. For example, you can use byaddr as the nickname for the ipnodes.byaddr map.

2 Planning an NIS Network

This chapter describes how to plan the layout of your NIS network. It tells you how to perform the following tasks:

Determining the Number of NIS Domains Required

For many sites, all hosts can belong to the same domain and it is not necessary to set up more than one domain. However, you might want to create multiple domains in the following situations:

- Sites that are divided into multiple administrative departments (like Marketing and Finance) with a different system administrator for each department where each system administrator is required to maintain a separate NIS domain
- Sites that are divided into multiple administrative departments, where each department requires different configuration data and allows access to different users and hosts. To ensure separation of data and access, you must create a separate NIS domain for each administrative department.

Determining the Number of NIS Servers

Following are some guidelines for determining the number of NIS servers required in an NIS network:

- If you want the client to bind through broadcast method, at least one server must be present in each subnetwork in a domain. However, using the `ypinit -c` command, you can specify a list of servers to which the NIS client can bind.
- In general, a server can serve up to 30 NIS clients if the clients and servers run at the same speed. If the clients are faster than the servers, you will need more servers. If the clients are slower than the servers, each server can serve 50 or more clients.

Determining the Hosts that will be NIS Servers

Following are some guidelines for determining the hosts that can be NIS servers in an NIS network:

- Select servers that are reliable and fast.
- Distribute servers appropriately among client networks.

3 Configuring and Administering an NIS Master Server

This chapter describes how to configure and administer an NIS master server.

An NIS master server holds the source files for all the NIS maps in the domain. Any changes to the NIS maps must be made on the NIS master server. The NIS master server delivers information to the NIS clients and supplies the NIS slave servers with up-to-date maps. An NIS master server can also be an NIS client.

This chapter addresses the following topics:

Creating the Master `passwd` File

To create the master `passwd` file, complete the following steps.

> 📝 **NOTE:** Back up the `/etc/passwd` file before you perform the steps described in this section.

1. Log in to the NIS master server as superuser.
2. Create a new directory, as follows.

   ```
   cd /etc
   mkdir nis
   ```

3. Copy the `/etc/passwd` file from each host in your NIS domain as `/etc/nis/passwd.hostname`, to the host designated at the master server.

   Where:

   `hostname` specifies the name of the host to which the file belongs.

   For example, for a host `ocelot`, the password file will be named as `/etc/nis/passwd.ocelot`.

4. Concatenate all the `passwd` files, including the master server's `passwd` file, into a temporary `passwd` file, as follows.

   ```
   cd /etc/nis
   cat passwd passwd.hostname1 passwd.hostname2...> passwd.temp
   ```

5. Sort the temporary `passwd` file by user name, as follows:

sort -o /etc/nis/passwd.temp -t: -k1,1 /etc/nis/passwd.temp

NOTE: NIS does not require that the `passwd` file be sorted on any of the fields. However, sorting the `passwd` file makes it easier to find duplicate entries.

6. Examine the `/etc/nis/passwd.temp` file, and remove the duplicate entries for the same user name.

NOTE: Ensure that each user in the network has a unique user name.

7. Sort the `passwd.temp` file by user ID, as follows:

sort -o /etc/nis/passwd.temp -t: -k3n,3 /etc/nis/passwd.temp

8. Examine the `/etc/nis/passwd.temp` file, and remove the duplicate entries of user IDs.

NOTE: Ensure that no two users in the network have the same user ID.

9. Move the sorted `/etc/nis/passwd.temp` file to the `/etc/passwd` file.

The `/etc/passwd` file is used to generate the `passwd` map for the NIS domain unless an alternate password file is used as source.

10. Remove all the `/etc/nis/passwd.hostname` files from the master server. For example, for a host `ocelot`, the password file `/etc/nis/passwd.ocelot` must be removed.

For more information about the `passwd` file, see *passwd* (4). For more information about the `sort` command, see *sort* (1).

Creating the Master `group` File

To create the master `group` file, complete the following steps:

NOTE: Back up the `/etc/group` file before you perform the steps described in this section.

1. Log in to the NIS master server as superuser.
2. If you have already created /etc/nis directory in the previous section skip this step.

Create a new directory, as follows.

cd /etc

mkdir nis

3. Copy the `/etc/group` file from each host in the NIS domain as the `/etc/nis/group.hostname` file to the NIS master server.

Where:

hostname specifies the name of the host to which the file belongs.

For example, for a host `ocelot`, the group file will be named as `/etc/nis/group.ocelot`.

4. Concatenate all the `group` files, including the master server's `group` file, into a temporary `group` file, as follows:

cd /etc/nis
cat group group.*hostname1* group.*hostname2*... > group.temp

5. Sort the temporary `group` file by group name, as follows:

sort -o /etc/nis/group.temp -t: -k1,1 /etc/nis/group.temp

NOTE: NIS does not require that the group file be sorted in any of the fields. However, sorting the group file makes it easier to find duplicate entries.

6. Examine the /etc/nis/group.temp file for duplicate group name entries.
7. Merge the groups with the same name into one group to remove the duplicate entries.
8. Sort the temporary group file by group ID, as follows:

   ```
   sort -o /etc/nis/group.temp -t: -k3n,3 /etc/nis/group.temp
   ```

9. Examine the /etc/nis/group.temp file for duplicate group IDs.

NOTE: Ensure that no two groups have the same group ID.

10. Move the sorted /etc/nis/group.temp file to the /etc/group file.

    The /etc/group file is used to generate the group map for the NIS domain.

11. Remove the /etc/nis/group.*hostname* files from the master server. For example, for a host ocelot, the group file /etc/nis/group.ocelot must be removed.

For more information about the group file, see *group* (4). For more information about the sort command, see *sort* (1).

Creating the Master hosts File

To create the master hosts file, complete the following steps:

NOTE: Back up the /etc/hosts file before you perform the steps described in this section.

1. Log in to the NIS master server as superuser.
2. If you have already created /etc/nis directory in the previous section skip this step.

   Create a new directory, as follows.

   ```
   cd /etc
   mkdir nis
   ```

3. Copy the /etc/hosts file from each host in your NIS domain as /etc/nis/hosts.*hostname* to the NIS master server.

   Where:

   hostname specifies the name of the host to which the file belongs. For example, for a host ocelot, the hosts file will be named as /etc/nis/hosts.ocelot.

4. Concatenate all the hosts files, including the master server's hosts file, into a temporary hosts file, as follows:

   ```
   cd /etc/nis/
   cat hosts hosts.hostname1 hosts.hostname2... > hosts.temp
   ```

5. Sort the temporary hosts file by IP address, as follows:

   ```
   sort -o /etc/nis/hosts.temp -b -k1,1 /etc/nis/hosts.temp
   ```

NOTE: NIS does not require that the hosts file be sorted in any particular way. However, sorting the hosts file makes it easier to find the duplicate entries.

6. Examine the /etc/nis/hosts.temp file for duplicate IP addresses, and remove the duplicate entries.

> 📝 **NOTE:** If an IP address needs to be mapped to multiple host names, include them as aliases in a single entry.

7. Sort the `hosts.temp` file by host name, as follows:

    `sort -o /etc/nis/hosts.temp -b -k2,2 /etc/nis/hosts.temp`

8. Examine the `/etc/nis/hosts.temp` file for duplicate host names. A host name may be mapped to multiple IP addresses only if the IP addresses belong to different LAN cards on the same host. If a host name appears in multiple entries mapped to IP addresses on different hosts, remove all the duplicate entries.

9. Examine the `/etc/nis/hosts.temp` file for duplicate aliases.

> 📝 **NOTE:** Ensure that an alias does not appear in more than one entry.

10. Move the sorted `/etc/nis/hosts.temp` file to the `/etc/hosts` file.

    The `/etc/hosts` file is used to generate the `hosts` map for the NIS domain.

11. Remove the `/etc/nis/hosts.hostname` files from the master server. For example, for a host `ocelot`, the hosts file named `/etc/nis/hosts.ocelot` must be removed.

For more information about the `hosts` file, see *hosts* (4). For more information about the sort command, see *sort* (1).

Enabling the NIS Master Server Capability

To enable the master server capability, complete the following steps:

> 📝 **NOTE:** Back up the `/etc/rc.config.d/namesvrs` file and the `/etc/nsswitch.conf` file before you perform the steps described in this section.

1. Log in as superuser to the host that is designated as the master server.
2. Ensure that the `$PATH` environment variable includes the following directory paths:
    *   `/var/yp`
    *   `/usr/lib/netsvc/yp`
    *   `/usr/ccs/bin`
3. Enter the following command to set up the NIS domain name:

    `/usr/bin/domainname domainname`

    Where:

    `domainname` is the name of the NIS domain being set up. For example,
    `/usr/bin/domainname marketing`

    If the hosts use short file names, ensure that the first 14 characters of the `domainname` uniquely identify the NIS domain in the network.

4. In the `/etc/rc.config.d/namesvrs` file, set the `NIS_DOMAIN` variable to the domain name, as follows:

    `NIS_DOMAIN=domainname`

5. In the `/etc/rc.config.d/namesvrs` file, set the `NIS_MASTER_SERVER` and `NIS_CLIENT` variables to 1, as follows:

    `NIS_MASTER_SERVER=1`
    `NIS_CLIENT=1`

    Setting the `NIS_MASTER_SERVER` variable to 1, makes the system an NIS master. Similarly, setting the `NIS_CLIENT` variable to 1, makes the system an NIS client. Although it is a

common practice to have both NIS client and NIS server on the same system, it is not mandatory. You can maintain them on different systems.

6. Enter the following command to create the NIS maps for the domain:

   ```
   /usr/sbin/ypinit -m
   ```

   The `ypinit` script prompts for the names of the slave servers.

7. Enter the names of the slave servers.

8. Run the following command to set up the system as a client:

   ```
   /usr/sbin/ypinit -c
   ```

9. Copy the /etc/nsswitch.nis file to /etc/nsswitch.conf, as follows:

   ```
   cp /etc/nsswitch.nis /etc/nsswitch.conf
   ```

10. Enter the following commands to run the NIS startup scripts:

    ```
    /sbin/init.d/nis.server start
    /sbin/init.d/nis.client start
    ```

The master server now runs as both an NIS master server and an NIS client. This enables the master server to query other servers as a client, and provide information to other hosts as a server. You must also configure the slave servers listed by the `ypinit` script.

For more information on configuring the slave servers, see "Configuring and Administering an NIS Slave Server" (page 35) and the following manpages: *domainname* (1), *ypinit* (1M), and *ypfiles* (4).

For a representation of the NIS network configuration, see Appendix A (page 49).

Verifying the NIS Master Server Configuration

This section describes how to verify the NIS master server configuration. To verify the NIS master server configuration, complete the following steps:

1. Log in to the master server as superuser.
2. Enter the following command to obtain the list of maps in the NIS network:

   ```
   /usr/bin/ypwhich -m
   ```

The `ypwhich -m` command lists all the NIS maps available to the local client, and displays the name of the master server that serves each map. In the following output, the local host is both the client and the master server:

The following output is displayed:

```
auto.master mastername
netid.byname mastername
publickey.byname mastername
mail.byaddr mastername
mail.aliases mastername
netgroup.byhost mastername
netgroup.byuser mastername
netgroup mastername
protocols.byname mastername
protocols.bynumber mastername
servi.bynp mastername
services.byname mastername
rpc.byname mastername
rpc.bynumber mastername
networks.byaddr mastername
```

```
networks.byname mastername
ipnodes.byaddr mastername
ipnodes.byname mastername
hosts.byaddr mastername
hosts.byname mastername
group.bygid masternamegroup.byname mastername
passwd.byuid mastername
passwd.byname mastername
ypservers mastername
```
Where:

mastername is the name of the local host. For example, in an NIS network, if the hostname of the master server is `mammoth`, *mastername* in the previous output will be replace by the hostname, that is, `mammoth`.

Configuring the NIS Master Server to Use a Private `passwd` File

This section describes how to configure the NIS master server to use a private `passwd` file by creating a restricted `/etc/passwd` file that can be used only by the NIS master server. The restricted `/etc/passwd.yp` file can be used to generate the NIS `passwd` maps, which are used by the other hosts in the NIS domain.

NOTE: Back up the `/etc/passwd` file and the `/etc/rc.config.d/namesvrs` file, before you perform the steps described in this section.

To configure the NIS master server to use a private `passwd` file, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Copy the `/etc/passwd` file to the `/etc/passwd.yp` file.
3. Edit the `/etc/passwd` file, and remove users who must not be allowed access to the NIS master server.
4. Edit the `/var/yp/Makefile` file. Replace the following line:

   `PWFILE=$(DIR)/passwd`

   with

   `PWFILE=$(DIR)/passwd.yp`

5. In the `/etc/rc.config.d/namesvrs` file, modify the value of the `YPPASSWDD_OPTIONS` variable from:

   `YPPASSWDD_OPTIONS="/etc/passwd -m passwd PWFILE=/etc/passwd"`

   to:

   `YPPASSWDD_OPTIONS="/etc/passwd.yp -m passwd PWFILE=/etc/passwd.yp"`

6. Stop and start the NIS server daemons as follows:

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

7. Enter the following commands to regenerate the NIS `passwd` maps from the `/etc/passwd.yp` file:

```
cd /var/yp
/usr/ccs/bin/make passwd
```

The `make passwd` command generates both the `passwd.byname` and the `passwd.byuid` maps, and copies them to the slave servers.

If the slave servers are not up and running yet, enter `make` with the `NOPUSH` flag set to `1`:

```
cd /var/yp
/usr/ccs/bin/make NOPUSH=1 passwd
```

For more information, see the following manpages: *passwd* (4), *make* (1), *ypmake* (1M), and *ypinit* (1M).

Setting up an Alternate Directory for the `passwd` File

On HP-UX 11i v3, specifying a directory where the `passwd` (and also the `/etc/shadow` file when in shadow mode) can be found is mandatory when starting the NIS `passwd` daemon, `rpc.yppasswdd`. You can use the `-D` option to specify the directory where the `passwd` and shadow files are found, as follows:

```
# rpc.yppasswdd -D /etc/nis
```

However, the default path for the `passwd` and shadow files is `/etc`. You must use this path to avoid a warning message at system startup.

The `-D /etc/nis` option must be added to the `YPPASSWDD_OPTIONS` variable in the `/etc/rc.config.d/namesvrs` file. You must restart the NIS server each time you modify the `namesvrs` file.

> **NOTE:** Back up the `/etc/rc.config.d/namesvrs` file before you perform the steps described in this section.

To set up an alternate directory for the password file, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Enter the following command to stop the NIS server:
   ```
   #/sbin/init.d/nis.server stop
   ```
3. Modify the `/etc/rc.config.d/namesvrs` file, and set the `YPPASSWDD_OPTIONS` variable, as follows:
   ```
   YPPASSWDD_OPTIONS= -D <directory>
   ```
   Where:

   `directory` is the name of the alternate directory. For example, `/etc/nis`
4. Enter the following command to restart the NIS server:`#/sbin/init.d/nis.server start`

Allowing Selected Clients and Slave Servers To Access The Master Server

The NIS network can be secured by restricting access to the NIS master server.

To selectively allow clients and slave servers to access the NIS master server, complete the following steps:

1. Log in to the NIS master server as superuser.
2. On the NIS master server, create the following file if it does not already exist:

   `/var/yp/securenets`
3. Add the following line to the `/var/yp/securenets` file:

   address_mask   IP_address

Where:

IP_address	Specifies the Internet address of the NIS client, the NIS slave server, or the subnet that requests the NIS information or transfers from the NIS master server
address_mask	Indicates the important bit fields in the *IP_address*.
	If a bit is set in the *address_mask* field, the corresponding bit in the source address of any incoming NIS requests must match the same bit in the *IP_address* field.

If a client or a slave host has multiple network interface cards, you must add the IP addresses of all the network interfaces to the `securenets` file. However, this is not required if the NIS server accepts requests on only one interface.

Example 1

The following entry from a `/var/yp/securenets` file allows only the NIS client at IP address 20.21.12.13 to request information from the NIS master server:

255.255.255.255 20.21.12.13

Each bit is set in the address mask. Only the host whose IP address is 20.21.12.13 is allowed access to the master server.

Example 2

The following entry from a `/var/yp/securenets` file allows any host on the network 20.21.12.13, to request NIS information or transfer NIS maps from the master server:

255.255.255.0 20.21.12.13

The last eight bits in the IP address are ignored, because the last eight bits of the address mask are set to 0. Any host whose IP address begins with 20.21.12 is allowed access to the master server.

> **NOTE:** You must restart the `ypserv` process for the changes made to the `/var/yp/securenets` file to be effective.

4. Enter the following commands to restart the `ypserv` process:

```
/sbin/init.d/nis.server stop
/sbin/init.d/nis.server start
```

For more information, see *securenets* (4).

Verifying the Contents of an NIS Map

To verify the contents of an NIS map, enter the following command:

`/usr/bin/ypcat -k` *mapname*

The `-k` option lists the key for each item in the map and the data associated with the key.

For example, in the `netgroup.byhost` map, the hostname is the key. If the `-k` option is not specified, the `ypcat` command lists all the data associated with each `netgroup` name. However, it does not list the `key`, hostname in this case.

For more information about the `ypcat` command, see *ypcat* (1).

Modifying an NIS Map

To modify an NIS map, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Modify the source file of the NIS map. For example, if you want to modify the NIS `hosts` map, modify the `/etc/hosts` file.

3. Enter the following commands to generate the map and to push it to the slave servers:

```
cd /var/yp
/usr/ccs/bin/make mapname
```

If the slave servers are not running, enter the following command with the NOPUSH flag set to 1, as follows:

```
cd /var/yp
/usr/ccs/bin/make NOPUSH=1 mapname
```

This procedure is effective for all NIS maps except the ypservers map, which does not have a source file. For information about modifying the ypservers map, see "Adding a Slave Server to the NIS Domain" (page 30) or "Removing a Slave Server from the NIS Domain" (page 31).

If you modify the passwd, group, or hosts map, you must regenerate the netid.byname map using the make hosts and make netid commands. However, it is not mandatory to use the make hosts command before rebuilding the netid maps using the make netid command. The netid.byname map maps the users to groups, where the entry for each user is followed by a list of all the groups to which the user belongs. The netid.byname map is generated from the /etc/passwd, /etc/group, and /etc/hosts files.

For more information, see *make*(1), *ypmake*(1M), *yppush*(1M), and *ypxfr*(1M).

Adding an AutoFS Map to the NIS Domain

AutoFS mounts directories automatically when users or processes request access to them. It unmounts directories automatically if the processes remain idle for more than 10 minutes. By default, an NIS domain that you set up contains an auto.master AutoFS map file.

To add an AutoFS map to the NIS domain, for example auto.tools, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Modify the /usr/sbin/ypinit script using a text editor, to add the AutoFS map to the MASTER_MAPS list variable, as follows:

```
MASTER_MAPS="group.bygid group.byname \
  hosts.byaddr hosts.byname netgroup netgroup.byhost \
  netgroup.byuser networks.byaddr networks.byname passwd.byname \
  passwd.byuid protocols.byname protocols.bynumber rpc.bynumber \
  services.byname vhe_list publickey.byname netid.byname mail.byaddr \
  mail.aliases auto.master rpc.byname servi.bynp auto.tools"
```

3. Modify the /var/yp/Makefile file, as follows:
   - Add the AutoFS map to the list of maps that begin with all:, as follows:

   ```
   all: passwd group hosts networks rpc services protocols \
        netgroup aliases publickey netid vhe_list auto.master\
        auto.tools
   ```

   - Duplicate the section that begins with $(YPDBDIR)/$(DOM)/auto_master.time. In this section, change all occurrences of auto.master or auto_master in the newly created section to the name of the map that you are adding. Ensure that some occurrences are auto_tools (the name of the ASCII file), and some are auto.tools (the name of the NIS database).

   ```
   $ (YPDBDIR)/$(DOM)/auto_tools.time: $(DIR)/auto_tools\
           @(sed -e "s/^[ |   ]*//g" -e "/^#/d" -e s/#.*$$//
   $(DIR)/auto_tools $(CHKPIPE)) |\
               $(MAKEDBM) - $(YPDBDIR) /$(DOM)/auto.tools;
           @touch $(YPDBDIR)/$(DOM)/auto_tools.time;
           @echo "updated auto.tools";
           @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM)\ auto.tools; fi
           @if [ ! $(NOPUSH) ]; then echo "pushed\ auto.tools"; fi
   ```

   In the /var/yp/Makefile file, duplicate the section that begins with auto.master:. In this section, change auto.master to auto.tools, and change both occurrences of auto_master.time to auto_tools.time.

```
auto.tools:
    @if [ $(NOPUSH) ]; then $(MAKE) $(MFLAGS) -k \
        $(YPDBDIR)/$(DOM)/auto_tools.time DOM=$(DOM) DIR=$(DIR); \
    else $(MAKE) $(MFLAGS) -k $(YPDBDIR)/$(DOM)/auto_tools.time \
        DOM=$(DOM) DIR=$(DIR) NOPUSH=$(NOPUSH);fi
```

4. Enter the following commands to generate the map:

```
cd /var/yp
/usr/ccs/bin/make auto.tools
```

Removing an AutoFS Map from the NIS Domain

In case the user wants to shift the tools repository from the NIS Master server to a different server, the auto.tools AutoFS map must be removed from the Master server and added to the other server.

To remove an AutoFS map from the NIS domain, for example auto.*tools*, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Modify the /usr/sbin/ypinit script using a text editor, to remove the auto.*tools* map name from the MASTER_MAPS list.
3. Modify the /var/yp/Makefile, as follows:
   - Remove the auto.*tools* map from the list of maps that begin with all:.
   - Remove the auto.*tools* section. For example:

   ```
   $ (YPDBDIR)/$(DOM)/auto_tools.time: $(DIR)/auto_tools\
       @(sed -e "s/^[ |   ]*//g" -e "/^#/d" -e s/#.*$$// \
   $(DIR)/auto_tools $(CHKPIPE)) |\
           $(MAKEDBM) - $(YPDBDIR) /$(DOM)/auto.tools;
       @touch $(YPDBDIR)/$(DOM)/auto_tools.time;
       @echo "updated auto.tools";
       @if [ ! $(NOPUSH) ]; then $(YPPUSH) -d $(DOM) auto.tools; fi
       @if [ ! $(NOPUSH) ]; then echo "pushed auto.tools"; fi
   ```

   - Remove the section that begins auto.*tools*. For example:

   ```
   auto.tools:
       @if [ $(NOPUSH) ]; then $(MAKE) $(MFLAGS) -k \
           $(YPDBDIR)/$(DOM)/auto_tools.time DOM=$(DOM) DIR=$(DIR); \
       else $(MAKE) $(MFLAGS) -k $(YPDBDIR)/$(DOM)/auto_tools.time \
           DOM=$(DOM) DIR=$(DIR) NOPUSH=$(NOPUSH);fi
   ```

4. On the master and on each of the slave servers, remove the map files, *mapname*.dir and *mapname*.pag, from the directory where the maps are stored. This directory is called /var/yp/*domainname*, where *domainname* is the name of the NIS domain.

   For example, if you are removing the auto.*tools* map from the Finance domain, you must enter the following commands on the master server and on each of the slave servers:

   ```
   cd /var/yp/Finance
   rm auto.tools.dir auto.tools.pag
   ```

For more information, see *ypinit* (1M), *make* (1), *ypmake* (1M), and *ypfiles* (4).

Adding a Slave Server to the NIS Domain

To add an NIS slave server to the NIS domain, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Enter the following command:

   ```
   cd /var/yp/domainname
   ```

   Where:

   domainname is the name of the domain to which the slave server is added. For example, for a master server in the marketing domain, the *domainname* will be marketing.

3. Enter the following command to create an editable ASCII text file from the `ypservers` map:

   `/usr/sbin/makedbm -u ypservers > tempfile`

4. Add the name of the new server to the ASCII `tempfile` file, using a text editor. For example, run the following command at the command prompt to specify the slave server in the `marketing` domain:

   `# vi /var/yp/<domain_name>/tempfile`

   In the `vi` editor window, enter the host name of the slave server. For example:

   `ocelot`

5. Enter the following command to regenerate the `ypservers` map from the ASCII file:

   `/usr/sbin/makedbm tempfile ypservers`

6. Log in to the new slave server as superuser, and configure it as an NIS slave server. For more information on configuring an NIS slave server, see "Configuring and Administering an NIS Slave Server" (page 35).

For more information, see *makedbm* (1M) and *ypfiles* (4).

Removing a Slave Server from the NIS Domain

⚠️ **WARNING!** On completing the procedure below, the NIS clients bound to the slave server being removed may hang for some time.

To remove an NIS slave server from the NIS domain, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Enter the following commands to create an editable ASCII text file from the `ypservers` map:

   ```
   cd /var/yp/domainname
   /usr/sbin/makedbm -u ypservers > tempfile
   ```

3. Remove the name of the slave server from the ASCII `tempfile` file, using a text editor.
4. Enter the following command to regenerate the `ypservers` map from the ASCII file:

   `/usr/sbin/makedbm tempfile ypservers`

5. Log in to the slave server as superuser.
6. Remove all the map files from the map directory, and remove the map directory. The map directory is called `/var/yp/domainname`, where `domainname` is the name of the NIS domain.

   For example, if you are removing a slave server from the Marketing domain, you must enter the following commands:

   ```
   cd /var/yp
   rm -rf Marketing
   ```

7. Enter the following command to disable the NIS server capability:

   `/sbin/init.d/nis.server stop`

8. If the slave server is not a slave server in any other NIS domain, use a text editor to set the `NIS_SLAVE_SERVER` variable to 0 in the `/etc/rc.config.d/namesvrs` file, as follows:

   `NIS_SLAVE_SERVER=0`

For more information, see *makedbm* (1M) *ypfiles* (4).

Enabling the DNS Forwarding Mode

The DNS forwarding mode enables the NIS server to contact DNS server to service hosts or ipnodes requests. NIS can automatically fetch the information from DNS and this operation is

transparent to the user or application and can be achieved without any change to the `/etc/nsswitch.conf` configuration file.

To enable the DNS Forwarding mode on the NIS server, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Enter the following command to stop the NIS server:

   `/sbin/init.d/nis.server stop`

3. Add the `-d` option to the YPSERV_OPTIONS variable in the `/etc/rc.config.d/namesvrs` file as follows:

   `YPSERV_OPTIONS="-d"`

4. Enter the following command to restart the NIS server:

   `/sbin/init.d/nis.server start`

Configuring an NIS Master Server in a Domain with Solaris Systems

To configure an NIS master server in a domain with Sun systems, complete the following steps:

1. Log in as superuser to the host that is designated as the master server.
2. If the HP `Makefile` is already customized, move it to the `/var/yp/Makefile.hp`. If not, ignore this step.
3. Copy the Sun `Makefile` in to the `/var/yp` directory on the HP system.

   If the Sun Makefile is not named `Makefile`, use a text editor to set the MAKEFILE_NAME variable to the name of the Sun Makefile in the `/usr/sbin/ypinit` script.

4. If the HP Makefile is already customized, add those changes to the Sun Makefile.
5. In the `/usr/sbin/ypinit` script on the HP host that is designated as the master server, add the `netmasks.byaddr`, `bootparams`, `ethers.byaddr`, and `ethers.byname` maps to the MASTER_MAPS variable.
6. On one of the Sun systems, locate or create an `/etc/ethers` file, an `/etc/bootparams` file, and an `/etc/netmasks` file that contain all the information required by the Sun systems.
7. Copy the `/etc/ethers`, `/etc/bootparams`, and `/etc/netmasks` files to the HP host that is designated as the master server.
8. Follow the steps for enabling the NIS Master Server capability described in the section, "Enabling the NIS Master Server Capability" (page 24)

Enabling the Shadow Password Mode

To enable the shadow password mode on an NIS master server, complete the following steps:

1. Log in to the NIS master server as superuser.
2. Set the SHADOW_MODE variable to 1 in `/etc/rc.config.d/namesvrs` file, as follows:

   `SHADOW_MODE=1`

3. Enter the following command:

   `# pwconv`

   This command moves the system to the shadow password mode.

NOTE: The `pwconv` and `pwunconv` commands enable you to switch the system between the shadow password mode and the normal mode.

4. Enter the following commands to restart the NIS server:
   ```
   /sbin/init.d/nis.server stop
   /sbin/init.d/nis.server start
   ```

NOTE: The encrypted password information used for creating NIS password maps is present in the shadow file and is visible in the passwd maps (`passwd.byname` and `passwd.byuid`).

Controlling NIS Logging

You can control logging for the following NIS processes:

- `ypxfr`
- `ypserv`
- `ypbind`
- `yppasswdd`

Each message logged by these daemons can be identified by the date, time, host name, process ID, and the name of the function that generated the message. You can direct log messages from all these NIS daemons to the same file.

The following subsections describe how to control logging of NIS processes:

- Stopping and Starting the Logging of `ypxfr`

  If `ypxfr` is run interactively from the command line, it logs messages to standard output.

  If `ypxfr` is run by `cron` or by `yppush`, it logs messages to the `/var/yp/ypxfr.log` file, if the file exits.

  To log messages in the `/var/yp/ypxfr.log` file, create it by entering the following command:

  `/usr/bin/touch /var/yp/ypxfr.log`

  To stop logging of `ypxfr` and to remove the `ypxfr.log` file, enter the following command:

  `/usr/bin/rm /var/yp/ypxfr.log`

  The log output of `ypxfr` cannot be redirected.

  For more information see *ypxfr* (1M), *cron* (1M), and *yppush* (1M) manpages.

- Start and Stop Logging of `ypserv`

  By default, the `ypserv` daemon logs messages to the file `/var/yp/ypserv.log`, if it exists.

  Verify whether the `/var/yp/ypserv.log` file exists and start logging by `ypserv` by entering the following command:

  `/usr/bin/touch /var/yp/ypserv.log`

  To stop logging of `ypserv`, remove the`ypserv.log` file by entering the following command:

  `/usr/bin/rm /var/yp/ypserv.log`

  To direct `ypserv` logging to a different file, complete the following steps:

  1. Enter the following commands to stop `ypserv`:

     `# /sbin/init.d/nis.server stop`

  2. Add the `-l` *logfile* option to the `YPSERV_OPTIONS` variable in the `/etc/rc.config.d/namesvrs` file, as follows:

     `YPSERV_OPTIONS="-l /var/yp/nis_log"`

If you specify a log file with the `-l` option, `ypserv` shares the same log file with the other NIS daemons.

3. Enter the following command to restart the NIS server:

```
# /sbin/init.d/nis.server start
```

For more information, see *ypserv* (1M).

- Configuring `ypbind` Logging

  1. Enter the following commands to stop the NIS client:

  ```
  # /sbin/init.d/nis.client stop
  ```

  2. Add the `-l` *logfile* option to the `YPBIND_OPTIONS` variable in the `/etc/rc.config.d/namesvrs` file, as follows:

  ```
  YPBIND_OPTIONS="-l /var/yp/nis_log"
  ```

  If you specify a log file with the `-l` option, `ypbind` shares the same log file with the other NIS daemons.

  3. Enter the following command to restart the NIS client:

  ```
  # /sbin/init.d/nis.client start
  ```

- Configuring `yppasswdd` Logging

  To configure the `yppasswdd` logging, complete the following steps:

  1. Enter the following commands to stop `yppasswdd`:

  ```
  # /sbin/init.d/nis.server stop
  ```

  2. Add the `-l` *logfile* option to the `YPPASSWD_OPTIONS` variable in the `/etc/rc.config.d/namesvrs` file, as follows:

  ```
  YPPASSWDD_OPTIONS="-l /var/yp/nis_log"
  ```

  If you specify a log file with the `-l` option, `yppasswdd` shares the same log file with the other NIS daemons.

  3. Enter the following command to restart the NIS server:

  ```
  # /sbin/init.d/nis.server start
  ```

4 Configuring and Administering an NIS Slave Server

This chapter describes how to configure an NIS slave server.

An NIS slave server provides information to the NIS clients and takes the load off the NIS master server. The NIS maps are created on the NIS master server and then transferred to the slave servers. Changes to the NIS maps must be made on the NIS master server, which then transfers the updated copies of the maps to the NIS slave servers.

This chapter addresses the following topics:

For information on removing the slave server, see "Removing a Slave Server from the NIS Domain" (page 31).

Modifying the Slave Server passwd File

This section describes how to modify the slave server /etc/passwd file to obtain user information from the master server.

NOTE: Back up the /etc/passwd file before you perform the step described in this section.

Remove all the users from the /etc/passwd file, except the superuser, users with administrative privileges, and the system entries required for the system to boot. Usually, system entries have user IDs less than 100.

The changes made to the /etc/passwd file on an NIS slave server must be identical to the changes made on an NIS client.

Following is an example of an /etc/passwd file on an NIS slave server:

```
root:0AnhFBmriKvHA:0:3::/:/bin/ksh
daemon:*:1:5::/:/bin/sh
bin:*:2:2::/bin:/bin/sh
adm:*:4:4::/usr/adm:/bin/sh
uucp:*:5:3::/usr/spool/uucppublic:/usr/lib/uucp/uucico
lp:*:9:7::/usr/spool/lp:/bin/sh
hpdb:*:27:1:ALLBASE:/:/bin/sh
```

For more information, see *passwd* (4).

Modifying Slave Server group File

This section describes how to modify the slave server group file.

NOTE: Backup the /etc/group file before attempting the steps mentioned in the procedure below.

Modifying the slave server group file involves removing all the groups from the /etc/group file, except the entries required for the system to boot. The groups that will be required are the list of groups in the /etc/passwd file.

The changes made to the /etc/group file on an NIS slave server must be identical to the changes made on an NIS client.

Following is an example of an /etc/group file on an NIS slave server:

```
root::0:rootl,sam
other::1:
bin::2:
sys::3:
adm::4:
daemon::5:
mail::6:
lp::7:
```

For more information, see *group* (4).

Enabling the NIS Slave Server Capability

To enable the NIS slave server capability, complete the following steps:

1. Enter the following command on the NIS master server to verify that the NIS master server is configured and running NIS:

    `/usr/bin/ypwhich`

2. Log in as a superuser to the host that is designated as the slave server.

3. Ensure that the $PATH environment variable includes the following directory paths:

    - `/var/yp`
    - `/usr/lib/netsvc/yp`
    - `/usr/ccs/bin`

4. Enter the following command to set the NIS domain name:

    `/usr/bin/domainname domainname`

    Where:

    domainname is the same as the domain name on the NIS master server.

5. In the `/etc/rc.config.d/namesvrs` file, set the `NIS_DOMAIN` variable to the domain name, as follows:

    `NIS_DOMAIN=domainname`

6. In the `/etc/rc.config.d/namesvrs` file, set the `NIS_SLAVE_SERVER` and `NIS_CLIENT` variables to 1, as follows:

    ```
    NIS_SLAVE_SERVER=1
    NIS_CLIENT=1
    ```

7. Enter the following command to set up the NIS slave server and copy the NIS maps from the master server:

    `/usr/sbin/ypinit -s NIS_server_name [DOM=domainname]`

    Where:

    NIS_server_name is the name of the master server. If the slave server serves a different domain from the one set by the `domainname` command, you must specify the *domainname* after *NIS_server_name*.

8. Run the following command to set up the system as a client to itself and the master sever:

    `/usr/sbin/ypinit -c`

9. Copy the /etc/nsswitch.nis file to /etc/nsswitch.conf, as follows:

    `cp /etc/nsswitch.nis /etc/nsswitch.conf`

10. Enter the following commands to start the NIS scripts:

    ```
    /sbin/init.d/nis.server start
    /sbin/init.d/nis.client start
    ```

To receive map updates from the NIS master server, you must add the new slave server to the `ypservers` map on the master server. For information about adding an NIS slave server to the NIS domain, see "Adding a Slave Server to the NIS Domain" (page 30) .

For more information see *domainname* (1), *ypinit* (1M), *ypfiles* (4), and Appendix A (page 49).

Verifying the NIS Slave Server Configuration

To verify the NIS slave server configuration, enter the following command to check whether the NIS slave server is running:

```
/usr/bin/ypwhich
```

The `ypwhich` command must return the host name of the slave server. If the `ypwhich` command does not return the name of the slave server, see "Troubleshooting NIS" (page 45) for more information on troubleshooting.

For more information, see *ypwhich* (1).

Scheduling Regular Map Transfers from the NIS Master Server

The NIS maps are pushed from the NIS master server to the NIS slave server using map transfers. You can schedule map transfers for files that are frequently modified, for example, `/etc/passwd` and `/etc/hosts` sample scripts. The sample scripts are available in `/usr/newconfig/var/yp`. You can customize these scripts to schedule the map transfer.

To schedule map transfers from the NIS master server, complete the following steps:

1. Log in to the slave server as superuser.
2. From the `/usr/newconfig/var/yp` directory, copy the sample scripts `ypxfr_1perday`, `ypxfr_2perday`, and `ypxfr_1perhour` scripts to the `/var/yp` directory, as follows:

   ```
   cp /usr/newconfig/var/yp/ypxfr_1perday /var/yp
   cp /usr/newconfig/var/yp/ypxfr_2perday /var/yp
   cp /usr/newconfig/var/yp/ypxfr_1perhour /var/yp
   ```

3. Create a `crontab` file to invoke these NIS scripts at regular intervals.

   Following is an example of a `crontab` file:

   ```
   0 21 * * * /var/yp/ypxfr_1perday
   30 5,19 * * * /var/yp/ypxfr_2perday
   15 * * * * /var/yp/ypxfr_1perhour
   ```

   The `crontab` file runs the `ypxfr_1perday` script at 9:00 p.m. every night, and the `ypxfr_2perday` script at 5:30 a.m. and 7:30 p.m. every day. It runs the `ypxfr_1perhour` script at 15 minutes past every hour.

4. Enter the following command to add a file to `crontab`:

   ```
   crontab filename
   ```

   Where:

   filename is the `crontab` file created in the previous step.

If you have created customized NIS maps for your domain, you must add the map names to the appropriate scripts or you can use the scripts provided as templates to create your own scripts.

In some domains, frequently transferring huge maps may generate excessive network traffic. To avoid this, you can schedule map transfers less frequently by editing the `/var/yp/ypxfer_1perhour` script. You must create a new script that contains `/etc/passwd` to be run at the desired lower frequency.

If you have multiple slave servers, ensure that the map transfer to all the servers does not happen simultaneously. The time specified in the `crontab` file on each slave server must be different.

For more information, see *cron* (1M), *crontab* (1), and *ypxfr* (1M).

Allowing Selected Clients to Access the Slave Server

To selectively allow clients to access the slave server and prevent access by other clients, complete the following steps:

1. On the NIS slave server, create the following file if it does not exist:

    `/var/yp/securenets`

2. Add entries to the `/var/yp/securenets` file with the following syntax:

    address_mask   *IP_address*

    Where:

    IP_address   Specifies the Internet address of an NIS client, NIS slave server, or the subnet that requests NIS information or transfers NIS maps from the NIS master server.

    address_mask   Indicates the important bit fields in the *IP_address*.

    If a bit is set in the *address_mask* field, the corresponding bit in the source address of any incoming NIS requests must match the same bit in the *IP_address* field.

    Example 1

    The following entry from the `/var/yp/securenets` file allows only the NIS client at IP address 20.21.12.15, to request information from the NIS slave server:

    255.255.255.255 20.21.12.15

    Each bit is set in the address mask. Only the host whose IP address is 20.21.12.15 is allowed access to the slave server.

    Example 2

    The following entry from a `/var/yp/securenets` file allows any host on the network 20.21.12.0, to request NIS information or transfer NIS maps from the slave server:

    255.255.255.0   20.21.12.15

    The last eight bits in the IP address are ignored because the last eight bits of the address mask are set to 0. Any host whose IP address begins with 20.21.12 is allowed access to the slave server.

3. Enter the following commands to stop and restart the `ypserv` process:

    ```
    /sbin/init.d/nis.server stop
    /sbin/init.d/nis.server start
    ```

For more information, see *securenets* (4).

5 Configuring and Administering an NIS Client

An NIS client is a system that obtains its configuration information from either an NIS master server or an NIS slave server.

This chapter addresses the following topics:

- "Modifying the NIS Client `passwd` File" (page 39)
- "Modifying the NIS Client `group` File" (page 39)
- "Enabling the NIS Client Capability" (page 40)
- "Verifying the NIS Client Configuration" (page 41)
- "Guidelines for Changing Passwords in the NIS Domain" (page 41)
- "Preventing an NIS Client from Binding to Unknown Servers" (page 42)
- "Binding an NIS Client to a Server on a Different Subnet" (page 42)

Modifying the NIS Client `passwd` File

This section describes how to modify the NIS client /etc/passwd file. You may need to do this if you want to validate login information with the NIS server instead of the local passwd file.

NOTE: Back up the /etc/passwd file before you perform the step described in this section.

To modify the NIS client `passwd` file, remove all the users from the /etc/passwd file, except the root user, other users with administrative privileges and the system entries required for the system to boot. Usually, system entries have user IDs less than 100.

The changes made to the /etc/passwd file on an NIS client must be identical to the changes made on an NIS slave server.

Following is an example of an /etc/passwdfile on an NIS client:

```
root:0AnhFBmriKvHA:0:3: :/:/bin/ksh
daemon:*:1:5::/:/bin/sh
bin:*:2:2::/bin:/bin/sh
adm:*:4:4::/usr/adm:/bin/sh
uucp:*:5:3::/usr/spool/uucppublic:/usr/lib/uucp/uucico
lp:*:9:7::/usr/spool/lp:/bin/sh
hpdb:*:27:1:ALLBASE:/:/bin/sh
tstusr:*:101:4:Test User:/tstusr:/bin/sh
tstusr1:*:105:4:Test User1:/tstusr1:/bin/sh
tstusr2:*:106:4:Test User2:/tstusr2:/bin/sh
```

For more information, see *passwd* (4).

Modifying the NIS Client `group` File

This section describes how to modify the NIS client `group` file.

NOTE: Back up the /etc/group file before performing the steps described in this section.

To modify the NIS client `group` file, remove all groups from the /etc/group file except the entries required for the system to boot. The groups that will be required are the groups listed in the /etc/passwd file.

The changes made to the /etc/group file on an NIS client must be identical to the changes made on the NIS slave server.

Following is an example of an /etc/group file on an NIS client:

```
root::0:root
other::1:root,hpdb
bin::2:root,bin
```

```
sys::3:root,uucp
adm::4:root,adm
daemon::5:root,daemon
mail::6:root
lp::7:root,lp
tty::10:
nuucp::11:nuucp
users::20:root
nogroup:*:-2:
smbnull::101:
hpsmh::102:
sshd::103:
nisg::104:
nisgr::105:
```

For more information, see *group* (4).

Enabling the NIS Client Capability

To enable the NIS client capability, complete the following steps:

> **NOTE:** Back up the NIS configuration file /etc/rc.config.d/namesvrs file and the /etc/nsswitch.conf file before performing the steps described in this section.

1. Enter the following command on the NIS master or slave server to verify that the NIS server to which the client wants to bind is running:

   `# /usr/bin/ypwhich -m`

2. Log in to the NIS client as superuser.

3. Ensure that the $PATH environment variable includes the following directory paths:
   - `/var/yp`
   - `/usr/lib/netsvc/yp`
   - `/usr/ccs/bin`

4. Enter the following command to set the NIS domain name:

   `/usr/bin/domainname domainname`

   Where:

   `domainname` is the domain name of the server serving the client. Following is an example of the /usr/bin/domainname command:

   `/usr/bin/domainname marketing`

   Where:

   `marketing` is the `domainname`

5. In the /etc/rc.config.d/namesvrs file, set the NIS_DOMAIN variable to the domain name and the NIS_CLIENT variable to 1, as follows:

   `NIS_DOMAIN=domainname`
   `NIS_CLIENT=1`

6. Copy the /etc/nsswitch.nis file to /etc/nsswitch.conf, as follows:

   `cp /etc/nsswitch.nis /etc/nsswitch.conf`

7. To start the NIS client processes, enter the following command:

   `/sbin/init.d/nis.client start`

For more information, see *domainname* (1), *ypbind* (1M), and *nsswitch.conf* (4). For information about how to make the client bind to a particular server, see "Binding an NIS Client to a Server on a Different Subnet" (page 42).

Verifying the NIS Client Configuration

To verify the NIS client configuration, complete the following steps:

1. Log in to the NIS client as superuser.
2. Enter the following command to list the NIS maps and the master server that serves the maps:

   `/usr/bin/ypwhich -m`

The `ypwhich -m` command lists all the NIS maps available to the client, and displays the name of the master server that serves each map.

An output similar to the following is displayed:

```
auto.master mastername
netid.byname mastername
publickey.byname mastername
mail.byaddr mastername
mail.aliases mastername
netgroup.byhost mastername
netgroup.byuser mastername
netgroup mastername
protocols.byname mastername
protocols.bynumber mastername
servi.bynp mastername
services.byname mastername
rpc.byname mastername
rpc.bynumber mastername
networks.byaddr mastername
networks.byname mastername
ipnodes.byaddr mastername
ipnodes.byname mastername
hosts.byaddr mastername
hosts.byname mastername
group.bygid mastername
group.byname mastername
passwd.byuid mastername
passwd.byname mastername
ypservers mastername
```

For example, on an NIS network with master server with hostname `mammoth` the mastername will be replaced by the *hostname* `mammoth`.

If the output is not displayed, see *ypwhich* (1).

Guidelines for Changing Passwords in the NIS Domain

This section lists the guidelines for changing passwords in the NIS domain.

All users in the NIS domain must observe the following rules when changing their password:

- Use `/usr/bin/yppasswd` or `passwd -r nis`, instead of the `passwd` command to change the login passwords.
- Users are encouraged to change their password before they leave for the day. This allows time for the updated NIS maps on the master server to be replicated on the slave servers.

The yppasswd command is a link to the passwd -r nis command. It modifies the passwd file on the NIS master server, regenerates the NIS passwd maps from the updated passwd file, and replicates the NIS passwd maps on the slave servers.

For more information, see *yppasswd*(1), *yppasswdd*(1M), *passwd*(1), *ypxfr*(1M), and *yppush*(1M).

Preventing an NIS Client from Binding to Unknown Servers

To prevent an NIS client from binding to unknown servers, complete the following steps:

1. Create the following file if it does not already exist:

   /var/yp/secureservers

2. Add lines to the file with the following syntax:

   address_mask   IP_address

   Where:

IP_address	Specifies the Internet address of an NIS server or the subnet of an NIS server from which the client accepts NIS information.
address_mask	Indicates the important bits in the *IP_address* field.
	If a bit is set in the *address_mask* field, the corresponding bit in the source address of any incoming NIS requests must match the same bit in the *IP_address* field.

   Example 1

   The following line from a /var/yp/secureservers file allows the NIS client to bind only to the server at IP address 20.21.22.23:

   255.255.255.255 20.21.22.23

   Each bit is set in the address mask. IP address of the NIS server must match the *IP_address* field exactly, for the client to bind to the server.

   Example 2

   The following line from a /var/yp/secureservers file allows the client to bind to any NIS server on the network 20.21.22.0.

   255.255.255.0 20.21.22.23

   The last eight bits in the IP address are ignored, because the last eight bits of the address mask are set to 0. The client binds to any server whose IP address begins with 20.21.22.

3. Enter the following commands to stop and start the ypbind process:

   ```
   /sbin/init.d/nis.client stop
   /sbin/init.d/nis.client start
   ```

If the master or the slave server has multiple interface cards, and these alternate network interface cards are used to contact the server, add the IP addresses of all cards to the secureservers file.

If you start the ypbind daemon with the -ypset option and run the ypset command to bind to an NIS specific server, the /var/yp/secureservers file is ignored and the NIS client may bind to any server. However, if the NIS client fails to bind to an NIS server, the ypbind daemon does not fall back to the broadcast mode. As a result, the NIS client remains unbound. In such cases, you must use the ypinit −c command to bind the NIS client to an NIS server.

For more information, see *ypbind* (1M).

Binding an NIS Client to a Server on a Different Subnet

This section describes how to bind an NIS client to an NIS server on a different subnet or to a specific server on the same subnet.

To bind an NIS client to any of the specified NIS servers on a different subnet, complete the following steps:

1. Log in to the NIS client as superuser.
2. Enter the following command, to construct a list of bind servers:

   `#/usr/sbin/ypinit -c`

   The `ypinit` command when invoked with the `-c` option, configures the local host as an NIS client so that the NIS client attempts to bind to a particular NIS server. Invoking the `ypinit` command with the `-c` option prompts the user to construct a list of NIS servers, in the order of preference, to which the client will try to bind. This list of NIS servers is stored in the `/var/yp/binding/<domain_name>/ypservers file`.

   NOTE: Ensure that the `/etc/hosts` file has appropriate entries for the NIS servers added above.

3. Run the following command to copy the `/etc/nsswitch.nis` file to the `/etc/nsswitch.conf` file:

   `# cp /etc/nsswitch.nis /etc/nsswitch.conf`

4. Enter the following commands to restart the NIS client:

   ```
   # /sbin/init.d/nis.client stop
   # /sbin/init.d/nis.client start
   ```

6 Troubleshooting NIS

This chapter discusses the following known problems and workarounds with NIS:

NIS "Server Not Responding" Message

If the message "NIS Server Not Responding" is displayed, you can do the following:

☐ Run the `/usr/sbin/ping` command on the NIS client to ensure that the NIS server is up and is reachable on the network. If the `ping` command fails either the server is down or the network has a problem. If the server is down, reboot it, or wait for it to start. For information on troubleshooting network problems, see "*Installing and Administering LAN/9000 Software*" available at http://docs.hp.com.

> **NOTE:** If the NIS client hangs during boot because the client is unable to bind to an NIS server you may need to temporarily disable NIS client binding by booting the client in single-user mode and setting the `NIS_CLIENT` variable equal to 0 in the `/etc/rc.config.d/namesvrs` file. When the root cause of the NIS binding problem is resolved, you can change the `NIS_CLIENT` variable back to 1.

To boot your NIS client without waiting for the server to come up, set the `NIS_CLIENT` variable to 0 in the `/etc/rc.config.d/namsvrs` file and then boot your client.

☐ Run the `ps -ef` command on the NIS server to check whether `ypserv` is running. If it is not, complete the following steps to start `ypserv`:

1. In the `/etc/rc.config.d/namesvrs` file on the NIS master server, ensure that the following variables are set:

   `NIS_MASTER_SERVER=1`

2. Run the following command to start up the NIS server:

   `/sbin/init.d/nis.server start`

A User Cannot Log In

If the user cannot login, you can do the following:

☐ If the user has recently changed passwords, ask the user to try logging in with the old password. If the user can log in using the old password, follow these steps:

1. Run the `ps -ef` command on the NIS master server to ensure that the `yppasswdd` daemon is running. If it is not, run the following command to start all the NIS server processes:

   `/sbin/init.d/nis.server start`

2. Check the `cron` scripts on the slave servers to ensure transfers of the `passwd` map from the master server are frequent enough. Once per hour is usually frequent enough, but frequent map transfers may increase network traffic. You might want to schedule map

transfers during intervals when the network load is low, and advise users to make their password changes during these intervals.

☐ Run the following command on the NIS client to determine the master server that provides the `passwd` map to the client:

`/usr/bin/ypwhich -m passwd`

If the server does not respond, see "NIS "Server Not Responding" Message" (page 45) in the troubleshooting chapter.

If the `ypwhich` command returns the name of the NIS master server, log in as root to the master server and ensure that the user has an entry in the `/etc/passwd` file. Then, run the following commands on the master server to generate the NIS `passwd` database from the `/etc/passwd` file (or from a private password file if it is being used), and replicate it to the NIS slave servers:

```
touch /etc/passwd
cd /var/yp
/usr/ccs/bin/make passwd
```

☐ Run the following command to check whether the NIS client has an entry in the `passwd` database on the NIS server to which it is bound:

`/usr/bin/ypmatch username passwd`

If the client has no entry in the `passwd` database, run the following commands on the NIS server to which the client is bound:
```
/usr/sbin/ypxfr passwd.byname
/usr/sbin/ypxfr passwd.byuid
```

This command transfers the `passwd` database from the NIS master server to the slave server where the command was run.

☐ If the user's NIS client is bound to a slave server, use the following procedure to ensure that the slave server is listed in the NIS master server's `ypservers` database:

1. Run the following command on the NIS client to determine the server to which the client is bound:

   `/usr/bin/ypwhich`

2. Log in to the NIS master server, and run the following command:

   `cd /var/yp/domainname`

3. Run the following command on the NIS master server to write the contents of the `ypservers` database to a temporary file:

   `/usr/sbin/makedbm -u ypservers > tempfile`

4. If the NIS slave server is not listed in `tempfile`, use a text editor to add it, and then run the following command to rebuild the `ypservers` database:

   `/usr/sbin/makedbm tempfile ypservers`

5. Log in to the NIS slave server and run the following command:

   `/usr/sbin/ypxfr ypservers`

"Unknown Host" Message

If the message "Unknown Host" is displayed, you can do the following:

☐ Run the following command to trace a lookup of the unknown host:

`/usr/contrib/bin/nsquery hosts hostname`

The trace will indicate the name services (BIND, NIS, or /etc/hosts) that were queried and in what order. If the host is not performing lookups the way you want, see "*NFS Services Administrator's Guide*", for instructions on configuring the Name Service Switch.

☐ Run the following command on the NIS client to determine which master server provides the `hosts` map:

```
/usr/bin/ypwhich -m hosts
```

If the server does not respond, see "NIS "Server Not Responding" Message" (page 45).

If the `ypwhich` command returns the name of the NIS master server, log in as root to the master server and ensure that the unknown host is listed in its /etc/hosts file. Then, run the following commands on the master server to generate the NIS `hosts` database from the /etc/hosts file and replicate it to the NIS slave servers:

```
touch /etc/hosts
cd /var/yp
/usr/ccs/bin/make hosts
```

☐ Run the following command to check whether the unknown host is listed in the `hosts` database on the NIS server to which the client is bound:

```
/usr/bin/ypmatch hostname hosts
```

If the host is not listed in the `hosts` database, run the following commands on the NIS server to which the client is bound:

```
/usr/sbin/ypxfr hosts.byname
/usr/sbin/ypxfr hosts.byaddr
```

This command transfers the `hosts` database from the NIS master server to the server where you issue the command.

☐ If the NIS client is bound to a slave server, use the following procedure to ensure that the slave server is listed in the NIS master server's `ypservers` database:

1. Run the following command on the NIS client to determine the server to which the client is bound to:

   ```
   /usr/bin/ypwhich
   ```

2. Log in as root to the NIS master server and run the following command to change to the directory where the domain databases reside:

   ```
   cd /var/yp/domainname
   ```

3. On the NIS master server, run the following command to write the contents of the `ypservers` database to a temporary file:

   ```
   /usr/sbin/makedbm -u ypservers > tempfile
   ```

4. If the NIS slave server is not listed in `tempfile`, use a text editor to add it, and then run the following command to rebuild the `ypservers` database:

   ```
   /usr/sbin/makedbm tempfile ypservers
   ```

5. Log in to the NIS slave server and run the following command:

   ```
   /usr/sbin/ypxfr ypservers
   ```

An NIS Client Cannot Bind to a Server

If the NIS client is trying to bind to a server and any of the NIS commands return any of the following messages:

```
ypcat: can't bind to an NIS server for domain domainname

ypmatch: can't match key.
        reason: can't communicate with ypbind
```

```
ypwhich: clntudp_create error RPC_PROG_NOT_REGISTERED
```
You can do the following:

- Check if the ypbind process is running:
  ```
  ps -ef | grep ypbind
  ```

- If ypbind is not running, run the following command to start all the NIS client processes:
  ```
  /sbin/init.d/nis.client start
  ```

NIS Returns Incorrect Information

If the message "NIS Returns Incorrect Information" is displayed, you can do one the following:

☐ Run the following command on the NIS client to determine the master server that provides the appropriate NIS map:
```
/usr/bin/ypwhich -m mapname
```
If the server does not respond, see "NIS "Server Not Responding" Message" (page 45).

☐ Log in as root to the NIS master server, and run the following command to check the contents of the appropriate NIS map:
```
/usr/bin/ypcat -k mapname
```
If the map contents are not correct, edit the ASCII file from which the map is generated. Then, run the following commands to regenerate the map and replicate it to the slave servers:
```
cd /var/yp
/usr/ccs/bin/make mapname
```

☐ Run the following command on the NIS client to check the contents of the map on the NIS server to which the client is bound:
```
/usr/bin/ypcat -k mapname
```
If the contents are not correct, log in as root to the slave server that serves the NIS client, and run the following command:
```
/usr/sbin/ypxfr mapname
```
This command transfers the map from the NIS master server to the slave server where you executed the command.

☐ If the NIS client is bound to a slave server, ensure that the slave server is listed in the NIS master server's ypservers database. Follow these steps:
   1. Run the following command on the NIS client to determine the server to which the client is bound to:
      ```
      /usr/bin/ypwhich
      ```
   2. Log in as root to the NIS master server and run the following command to change to the directory where the domain databases reside:
      ```
      cd /var/yp/domainname
      ```
   3. On the NIS master server, run the following command to write the contents of the ypservers database to a temporary file:
      ```
      /usr/sbin/makedbm -u ypservers > tempfile
      ```
   4. If the NIS slave server is not listed in tempfile, use a text editor to add it, and then run the following command to rebuild the ypservers database:
      ```
      /usr/sbin/makedbm tempfile ypservers
      ```
   5. Log in to the NIS slave server and run the following command:
      ```
      /usr/sbin/ypxfr ypservers
      ```

☐ Make sure the slave servers have cron scripts that schedule regular updates of the map.

A Sample NIS Session

This appendix discusses a sample configuration for an NIS network, where we configure an NIS master (hostname: `mammoth`), NIS slave (hostname:`ocelot`) and an NIS client for the domain `marketing`.

The following topics are addressed here:

Configure an NIS Master Server

1. Modify the `/etc/rc.config.d/namesvrs` file and set the fields mentioned as follows:

   ```
   NIS_MASTER_SERVER=1
   NIS_SLAVE_SERVER=0
   NIS_CLIENT=1
   NIS_DOMAIN=marketing
   ```

2. Enter the following command to setup the local host as the master server for all maps:

   ```
   #/usr/sbin/ypinit -m
   ```

   The `ypinit` procedure starts by asking a few questions in order to properly install NIS in your environment.

   ```
   Do you want this procedure to quit on non-fatal errors? [y/n: n] n
   ```

   Remember to correct anything which fails. If you do not, some part of the system (perhaps the NIS itself) may not work.

   ```
   Can the existing directory "/var/yp/marketing" and its contents be destroyed? [y/n: n] y
   ```

   At this point, you must construct a list of the hosts that will be NIS servers (both master and slave) for the `marketing` domain.

   ```
   This machine, mammoth, is in the list of Network Information Service servers.
   ```

   Provide the hostnames of the slave servers, one per line. When you have no more names to add, enter `Ctrl+D` or a blank line.

   ```
   next host to add: mammoth

   next host to add: ocelot

   next host to add
   ```

   The current list of NIS servers looks like this:

   ```
   mammoth
   ocelot
   ```

   ```
   Is this correct? [y/n: y]
   ```

   There will be no further questions. The remainder of the procedure should take 5 to 10 minutes.

   ```
   Building the ypservers database... ypservers build complete.
   ```

   ```
   Running make in /var/yp: updated passwd updated group updated hosts updated ipnodes updated networks
   updated rpc updated services updated protocols updated netgroup updated aliases updated publickey updated
    netid updated auto.master
   ```

   ```
   mammoth has been set up as a master Network Information Service server without any errors.
   ```

   If there are running slave NIS servers, run *yppush* (1M) for any databases which have been changed. If there are no running slaves, run `ypinit` on those hosts which are to be slave servers.

3. Configure NIS Master as an NIS Client.

   ```
   # /bin/rm -f /var/yp/binding/marketing/ypservers
   # /usr/sbin/ypinit -c
   ```

If it is desired that the NIS client has to bind to a particular NIS server, a list of NIS servers must be constructed first. Continue to add the names of NIS servers in the order of preference, one per line. When done with the list, press `Ctrl+D` or the `Return` key on a line by itself.

```
next host to add: mammoth

next host to add:
```

The current list of `ypservers` looks like this:

```
mammoth

Is this correct?  [y/n: y]  y
```

Type `y` or `n` to confirm the output.

📝 **NOTE:** Ensure that the file `/etc/hosts` has appropriate entries for the NIS servers added above.

4. Enter the following command to start the NIS master:

   ```
   #/sbin/init.d/nis.server start
   ```

5. Enter the following command to start the NIS client:

   ```
   #/sbin/init.d/nis.client start
   ```

For detailed information on configuring an NIS Master Server, see

Configure an NIS Slave Server

1. Modify the `/etc/rc.config.d/namesvrs` file and set the fields mentioned as follows:

   ```
   NIS_MASTER_SERVER=1
   NIS_SLAVE_SERVER=0
   NIS_CLIENT=1
   NIS_DOMAIN=marketing
   ```

2. Enter the following command, to create NIS databases on a slave server by copying the databases from NIS master server for the marketing domain:

   ```
   #/usr/sbin/ypinit -s mammoth DOM=marketing
   ```

   The `ypinit` procedure starts by asking a few questions in order to properly install NIS in your environment.

   ```
   Do you want this procedure to quit on non-fatal errors? [y/n: n] n
   ```

   Remember to correct anything which fails. If you do not, some part of the system (perhaps the NIS itself) may not work.

   ```
   Can the existing directory "/var/yp/marketing" and its contents be destroyed? [y/n: n] y
   ```

   There will be no further questions. The remainder of the procedure, for copying the databases from `mammoth`, will take few minutes.

📝 **NOTE:** If your master NIS server `mammoth`, is an HP machine, the NIS databases `ethers.byaddr`, and `ethers.byname` do not exist for you to copy. As a result, you can ignore any `no such map` error messages displayed when the systems tries to transfer those maps.

3. Configure NIS slave server as an NIS Client.

   ```
   # /usr/sbin/ypinit -c
   ```

   If it is desired that the NIS client has to bind to a particular NIS server, a list of NIS servers must be constructed first. Continue to add the names of NIS servers in the order of preference, one per line. When done with the list, press `Ctrl+D` or the `Return` key on a line by itself.

   ```
   next host to add: mammoth

   next host to add: ocelot
   ```

The current list of yp servers looks like this:

```
mammoth
ocelot
Is this correct?  [y/n: y]  y
```

Type y or n to confirm the output.

> **NOTE:** Ensure that the /etc/hosts file has appropriate entries for the NIS servers added above.

4. Enter the following command, to start the NIS slave server:

   ```
   #/sbin/init.d/nis.server start
   ```

5. Enter the following command to start the NIS client:

   ```
   #/sbin/init.d/nis.client start
   ```

For more information on how to configure an NIS Slave Server, see "Enabling the NIS Slave Server Capability" (page 36)

Configure an NIS Client

1. Modify the /etc/rc.config.d/namesvrs file for the fields mentioned below, with the corresponding values mentioned against the fields

   ```
   NIS_MASTER_SERVER=1
   NIS_SLAVE_SERVER=0
   NIS_CLIENT=1
   NIS_DOMAIN=marketing
   ```

2. Configure an NIS client:

   ```
   #/usr/sbin/ypinit -c
   ```

   If you want the NIS client to bind to a particular NIS server, a list of NIS servers must be constructed. Continue to add the names of NIS servers in the order of preference, one per line. Press Ctrl+D or the Return key to exit.

   ```
   next host to add:
   ```

   The current list of yp servers looks like this:

   ```
   mammoth
   ocelot
   Is this correct?  [y/n: y]  y
   ```

   Type y or n to confirm the output.

3. Start the NIS client:

   ```
   # /sbin/init.d/nis.client start
   ```

> **NOTE:** Ensure that the file /etc/hosts has appropriate entries for the NIS servers added above.

For more information about an NIS Client, see "Enabling the NIS Client Capability" (page 40)

B NIS Configuration File /etc/rc.config.d/namesvrs

The /etc/rc.config.d/namesvrs file is the main NIS configuration file, through which the NIS services can be enabled or disabled and appropriate options can be provided for the NIS service. It also contains a list of the NIS maps built in the domain.

Following is a listing of the/etc/rc.config.d/namesvrs file:

```
#
# @(#)B.11.31_LR namesvrs $Revision: $
#
# (C) Copyright 1998, 2000-2006 Hewlett-Packard Development Company, L.P.

unset UNIX95
PRE_U95=true;export PRE_U95;

# The DNS variables "NAMED" and "NAMED_ARGS" are removed from this file.
# These variables are now available in the file /etc/rc.config.d/namesvrs_dns.

###########################################################
# NIS (YP) configuration.  See domainname(1), ypserv(1m) #
###########################################################
#
#  Name server using Network Information Service protocol (Sun's ONC)
#
# NIS_MASTER_SERVER: 1 if this node is the master NIS server, 0 if not.
# NIS_SLAVE_SERVER: 1 if this node is a slave NIS server, 0 if not.
# NIS_CLIENT:  1 if this node is a NIS client, 0 if not.
# NISDOMAIN:  the NIS domain name (see domainname(1)).
# NISDOMAIN_ERR: ???
# MAX_NISCHECKS= max number of bind attempts the NIS client will make
# YPSERV_OPTIONS command line options for ypserv
# YPBIND_OPTIONS command line options for ypbind
# YPPASSWDD_OPTIONS command line options for yppasswdd
# KEYSERV_OPTIONS command line options for keyserv
# YPUPDATED_OPTIONS command line options for ypupdated
# YPXFRD_OPTIONS command line options for ypxfrd
# YPSET_ADDR  IP addr or name of NIS server. Must set YPBIND_OPTIONS
#   to "-ypset"
# LONGNAMES_OPTIONS
# SHORTNAMES_OPTIONS
# SHADOW_MODE  1 if shadow mode is to be enabled, 0 if not.
#
# Note: - Setting SHADOW_MODE is one of the criteria for deciding whether
#   the system should be in shadow mode or standard mode. Applications
#   ultimately checks for the existence of a shadow file also to
#   determine the mode of the system.
#
# Note: - NIS_MASTER_SERVER and NIS_SLAVE_SERVER are mutually exclusive,
#   i.e., only one, not both, should be set if either is set.
# - All NIS servers must also be NIS clients, so if you set either
#   NIS_MASTER_SERVER or NIS_SLAVE_SERVER to 1, you should set
#   NIS_CLIENT to 1, too.
# - Refer to NFS administration manual and ypinit(1M) for the
#   steps required to create NIS servers.
#
# MAX_NISCHECKS
# The maximum number of bind attempts the
# NIS client will make for a NIS server to be
# available before shutting down NIS and
# booting without it. This value should
# not be less than 2.
#
NIS_MASTER_SERVER=0
NIS_SLAVE_SERVER=0
NIS_CLIENT=0
NIS_DOMAIN=
MAX_NISCHECKS=2
YPSERV_OPTIONS=""
YPBIND_OPTIONS=""
```

```
YPPASSWDD_OPTIONS="/etc/passwd -m passwd PWFILE=/etc/passwd"
KEYSERV_OPTIONS=""
YPUPDATED_OPTIONS=""
YPXFRD_OPTIONS=""
YPSET_ADDR=""
LONGNAMES_OPTIONS="group.bygid.dir group.bygid.pag group.byname.dir \
group.byname.pag hosts.byaddr.dir hosts.byaddr.pag \
hosts.byname.dir hosts.byname.pag networks.byaddr.dir \
networks.byaddr.pag networks.byname.dir networks.byname.pag \
passwd.byname.dir passwd.byname.pag passwd.byuid.dir \
passwd.byuid.pag protocols.byname.dir protocols.byname.pag \
protocols.bynumber.dir protocols.bynumber.pag \
rpc.bynumber.dir rpc.bynumber.pag services.byname.dir \
services.byname.pag ypservers.dir ypservers.pag \
ipnodes.byname.pag ipnodes.byname.dir ipnodes.byaddr.pag \
ipnodes.byaddr.dir"

SHORTNAMES_OPTIONS="group.bygi.dir group.bygi.pag group.byna.dir \
group.byna.pag hosts.byad.dir hosts.byad.pag \
hosts.byna.dir hosts.byna.pag netwk.byad.dir \
netwk.byad.pag netwk.byna.dir netwk.byna.pag \
passw.byna.dir passw.byna.pag passw.byui.dir \
passw.byui.pag proto.byna.dir proto.byna.pag \
proto.bynu.dir proto.bynu.pag rpc.bynu.dir \
rpc.bynu.pag servi.byna.dir servi.byna.pag \
ypservers.dir ypservers.pag ip.byna.pag ip.byna.dir \
ip.byad.pag ip.byad.dir"

SHADOW_MODE=0
```

C Using NIS in Compat Mode

This section describes how to use Compat Mode, also called the Compatibility Mode, for controlling the name services used to obtain user and group information.

If you specify *compat* as a name service in the `/etc/nsswitch.conf` file for the `passwd:` and `group:` entries, then NIS consults the local `/etc/passwd` or `/etc/group` file. Any lines in the `/etc/passwd` or `/etc/group` file beginning with a plus (+) or a minus (-) sign directs the lookups to NIS. This usage of compat mode and the plus (+) and minus(-) sign gives the same name service behavior as was provided by default in the HP-UX releases previous to the HP–UX 10.30 release.

For more information on configuring the Name Service Switch, see *NFS Services Administrator's Guide*.

Using Netgroups in the `/etc/passwd` File

In the `/etc/passwd` file, netgroups can be used to indicate whether user information should be looked up in the NIS passwd database.

The following example line from the `/etc/passwd` file indicates that users in the netgroup `animals` should be looked up in the NIS passwd database:

`+@animals`

The `animals` netgroup is defined as follows in the `/etc/netgroup` file:

`animals (-,mickey, ), (-,daffy, ), (-,porky, ), (-,bugs, )`

Note that the `/etc/passwd` file is searched sequentially, so if user `mickey`, `daffy`, `porky`, or `bugs` appears before the `animals` netgroup in the `/etc/passwd` file, the NIS database will never be consulted for information on that user.

The Name Service Switch configuration is used to determine where to look for the contents of a netgroup.

Netgroups can also be used to prevent lookups of certain users in the NIS `passwd` database. The following example lines from the `/etc/passwd` file indicate that if the NIS `passwd` database contains entries for users in the `bears` and `marketing` netgroup, these entries cannot be used on the local system. Any other users can be looked up in the NIS database.

`-@bears:::::::`

`+@documentation:::::::`

`-@marketing:::::::`

These lines result in searching the NIS database for users from the `documentation` netgroup. Access is denied for users from the `bears` and `marketing` netgroup.

For information on the /etc/passwd file, type man 4 passwd at the HP-UX prompt.

Editing the Slave Server's passwd File

- HP recommends you to remove all users from the `/etc/passwd` file except the root user and the system entries required for your system to boot. By convention, system entries usually have user IDs less than 100, so all entries with user IDs of 100 or greater can be evaluated for removal.
- The Name Service Switch configuration file provided for NIS (`/etc/nsswitch.nis`) causes your host to check its local `/etc/passwd` file and then continue to the NIS `passwd` map if the requested information is not in the local file. However, in previous releases, you had to add a plus sign (+) to the `/etc/passwd` file to cause your host to check the NIS passwd database.

  If you want your host to behave as it did before HP-UX release 10.30, add the following entry as the last line in the `/etc/passwd` file:

  `+:::::`

Also, make sure your `/etc/nsswitch.conf` file specifies `compat` as the name service for `passwd`.

The plus sign (+) causes processes to consult NIS for any user information not found in the local `/etc/passwd` file.

For more information about the entries in the `/etc/passwd` file, see *passwd*(4).

⚠ **CAUTION:** An asterisk (*) in the password field prevents all NIS users from logging in when NIS is running.

Following is an example `/etc/passwd` file on an NIS slave server:

```
root:0AnhFBmriKvHA:0:3::/:/bin/ksh

daemon:*:1:5::/:/bin/sh

bin:*:2:2::/bin:/bin/sh

adm:*:4:4::/usr/adm:/bin/sh

uucp:*:5:3::/usr/spool/uucppublic:/usr/lib/uucp/uucico

lp:*:9:7::/usr/spool/lp:/bin/sh

hpdb:*:27:1:ALLBASE:/:/bin/sh

+::::::
```

For more information, type `man 4 passwd` at the HP-UX prompt.

Editing the `passwd` File for Compat Mode

- HP recommends you to remove all groups from the `/etc/group` file except the group entries required for your system to boot.
- The Name Service Switch configuration file provided for NIS (`/etc/nsswitch.nis`) causes your host to check its local `/etc/group` file and then continue to the NIS `group` map if the requested information is not in the local file. However, in previous releases, you had to add a plus sign (+) to the `/etc/group` file to cause your host to check the NIS `group` database.

  If you want your host to behave as it did before HP-UX release 10.30, add the following entry as the last line in the `/etc/group` file:

  ```
  +:::
  ```

  Also, make sure your `/etc/nsswitch.conf` file specifies `compat` as the name service for `group`.

  The plus sign (+) causes processes to consult NIS for any group information not found in the local `/etc/group` file.

  The changes you make to the `/etc/group` file on an NIS slave server are the same changes you make on an NIS client. Following is an example `/etc/group` file on an NIS client:

```
root::0:rootl,sam

other::1:

bin::2:

sys::3:

adm::4:

daemon::5:

mail::6:

lp::7:

+:*:*
```

For more information, type `man 4 group` at the HP-UX prompt.

Logging in Issue with NIS

This section suggests a way to correct a common logging in problem encountered with NIS.

If you are using NIS `compat` mode, make sure the NIS escape entry in the `/etc/passwd` file on the client does not have an asterisk in the password field.

For example, on HP systems, the NIS escape entry in the `/etc/passwd` file is:

```
+::::::
```

Index