

ULTRIX

digital

Guide to the Yellow Pages Service

Order Number: AA-ME00B-TE

Guide to the Yellow Pages Service

Order Number: AA-ME00B-TE

June 1990

Product Version: ULTRIX, Version 4.0 or higher

Yellow Pages is a distributed data lookup service for sharing information between systems on a network. This guide describes the Yellow Pages service.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013.

© Digital Equipment Corporation 1990
All rights reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital	DECUS	ULTRIX Worksystem Software
CDA	DECwindows	VAX
DDIF	DTIF	VAXstation
DDIS	MASSBUS	VMS
DEC	MicroVAX	VMS/ULTRIX Connection
DECnet	Q-bus	VT
DECstation	ULTRIX	XUI
	ULTRIX Mail Connection	

Ethernet is a registered trademark of Xerox Corporation.

Network File System and NFS are trademarks of Sun Microsystems, Inc.

UNIX is a registered trademark of AT&T in the USA and other countries.

Contents

About This Guide

Audience	vii
Organization	vii
Related Documents	vii
Conventions	viii
New and Changed Information	viii

1 Introduction to the Yellow Pages Service

1.1 Overview of the Yellow Pages Service	1-1
1.2 How the YP Service Operates	1-2
1.2.1 Naming Domains	1-2
1.2.2 Storing Data	1-2
1.2.3 Default YP Files	1-3
1.2.3.1 The /etc/hosts File	1-3
1.2.3.2 The /etc/passwd File	1-4
1.2.3.3 Other YP Files	1-4
1.2.4 The /etc/svc.conf File	1-4

2 Setting Up and Modifying Yellow Pages

2.1 Overview of Setting Up a YP Server and Client	2-1
2.2 Prerequisite Information	2-2
2.3 Setting Up YP Manually	2-2
2.3.1 Setting Up a YP Master Server	2-2
2.3.1.1 Establish the YP Domain	2-3
2.3.1.2 Build the Default YP Maps	2-4
2.3.1.3 Start the YP Server Daemons	2-4
2.3.1.4 Modify the Default YP Files	2-4
2.3.1.5 Start the YP Password Server	2-5
2.3.1.6 Edit the /etc/rc.local File	2-5

2.3.1.7	Create the YP Servers Map	2-5
2.3.2	Modify the /etc/svc.conf File	2-5
2.3.2.1	Edit the /etc/svc.conf File with svcsetup	2-5
2.3.2.2	Editing the /etc/svc.conf File Manually	2-7
2.3.3	Setting Up a YP Slave Server	2-7
2.3.3.1	Establish the YP Domain	2-8
2.3.3.2	Obtain Copies of the YP Maps	2-8
2.3.3.3	Start the YP Server Daemons	2-8
2.3.3.4	Modify the Default YP Database Files	2-8
2.3.3.5	Edit the /etc/rc.local File	2-8
2.3.3.6	Modify the /etc/svc.conf File	2-9
2.3.3.7	Edit the /usr/lib/crontab File	2-9
2.3.3.8	Add the New Slave Server to the Domain	2-9
2.3.4	Setting Up a YP Client	2-9
2.3.4.1	Establish the YP Domain	2-10
2.3.4.2	Start the YP Port Mapper Daemon	2-10
2.3.4.3	Modify the Default YP Database Files	2-10
2.3.4.4	Edit the /etc/rc.local File	2-10
2.3.4.5	Modify the /etc/svc.conf File	2-10
2.4	Altering YP Client Local Files	2-10
2.4.1	The networks, protocols, rpc, services, and netgroup Files	2-11
2.4.2	The /etc/hosts.equiv File	2-11
2.4.3	The /rhosts File	2-11
2.4.4	The /etc/hosts File	2-12
2.4.5	The /etc/passwd File	2-12
2.4.6	Reducing the /etc/group File to a Single Line	2-13
2.5	Modifying and Creating YP Maps	2-13
2.6	Propagating YP Maps	2-14
2.6.1	Using make to Propagate YP Maps	2-14
2.6.2	Using makedbm to Propagate YP Maps	2-15
2.6.3	Using ypxfr to Propagate YP Maps	2-15
2.6.3.1	Running ypxfr from cron	
2.6.3.2	Running ypxfr Manually	
2.7	Modifying the YP Environment	2-16
2.7.1	Adding YP Servers to the Domain	2-16
2.7.2	Removing YP Slave Servers from the Domain	2-17
2.7.3	Changing a YP Map's Master Server	2-17
2.7.4	Adding Users to a YP Client	2-18
2.7.4.1	Edit the YP Master Server's /etc/passwd File	2-18
2.7.4.2	Update the YP Map	2-19
2.7.4.3	Make a Home Directory	2-19

2.7.4.4	Set Up the New User's Environment	2-20
2.7.4.5	Propagate the Updated YP Map	2-20

3 Maintaining and Managing the Yellow Pages Service

3.1	System Security with YP	3-1
3.1.1	Using the -S Option	3-1
3.1.2	Global and Local YP Files	3-1
3.1.3	Local System Files with Pointers to YP Maps	3-2
3.2	YP Map Access Policies	3-3
3.3	Special YP Password Change	3-4
3.4	Using Netgroups	3-4

4 Troubleshooting the Yellow Pages Service

4.1	How to Solve Problems on a YP Client	4-1
4.1.1	Solving the Problem of Commands That Hang	4-1
4.1.2	Problems Caused by YP Service Unavailability	4-3
4.1.3	Determining why the ypbind Process Exits	4-4
4.1.4	Why the ypwhich Command Might Be Inconsistent	4-4
4.2	How to Solve Problems on a YP Server	4-4
4.2.1	Updating a YP Database	4-6
4.2.2	Determining Why the ypserv Process Exits	4-6

A Appendix

About This Guide

Yellow Pages is a distributed data lookup service for sharing information between systems on a network. This guide describes the Yellow Pages service.

The objective of this guide is to provide introductory, setup, and troubleshooting information for the Yellow Pages Service (YP). This guide will assist you in developing YP management procedures and presents guidelines from which you can develop specific procedures for your site.

Audience

This guide is meant for the person responsible for maintaining networks on an ULTRIX operating system. This person is usually the system manager, but could be a network manager or the system manager who is also a user of a MicroVAX processor. This guide assumes that the reader is familiar with the ULTRIX system commands, the system configuration, the naming conventions, and an editor such as `vi` or `ed`. It assumes that the reader knows the names and addresses of the other systems on the network.

Organization

This guide consists of four chapters, an appendix, and an index. The chapters are:

- Chapter 1 Introduces YP and provides the background information needed before you can set up and run YP on your system.
 - Chapter 2 Describes how to manually set up YP on your system. The description is included for those who want to understand how YP operates and which files are affected by YP.
 - Chapter 3 Describes how to maintain and manage the YP service. System security and YP map access policies are discussed.
 - Chapter 4 Describes the basic approach to solving YP-related problems. It discusses various system problems you may encounter and explains how to solve them. In addition, this chapter lists the YP error messages and suggested solutions.
- Appendix A Lists and describes common commands.

Related Documents

You should have available the related hardware documentation for your system. You should also have the ULTRIX documentation set, including the ULTRIX Reference Pages.

Conventions

The following conventions are used in this guide:

<code>%</code>	The default user prompt is your system name followed by a right angle bracket. In this manual, a percent sign (<code>%</code>) is used to represent this prompt.
<code>#</code>	A number sign is the default superuser prompt.
user input	This bold typeface is used in interactive examples to indicate typed user input.
<code>system output</code>	This typeface is used in interactive examples to indicate system output and also in code examples and other screen displays. In text, this typeface is used to indicate the exact name of a command, option, partition, pathname, directory, or file.
UPPERCASE lowercase	The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function definitions must be typed exactly as shown.
<code>rlogin</code>	In syntax descriptions and function definitions, this typeface is used to indicate terms that you must type exactly as shown.
[]	In syntax descriptions and function definitions, brackets indicate items that are optional.
{ }	In syntax descriptions and function definitions, braces enclose lists from which one item must be chosen. Vertical bars are used to separate items.
. . .	In syntax descriptions and function definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
<code>cat(1)</code>	Cross-references to the <i>ULTRIX Reference Pages</i> include the appropriate section number in parentheses. For example, a reference to <code>cat(1)</code> indicates that you can find the material on the <code>cat</code> command in Section 1 of the reference pages.

New and Changed Information

This guide is a revision. New and changed information includes the following:

- The file, `/etc/svc.conf` now performs many of the actions formerly done by the file, `/etc/svcorder`. References to this change appear throughout this guide.
- A new option, `-S`, is available with the `ypbind` command with the entries for `/etc/ypbind` and `/etc/portmap` in the `/etc/rc.local` file. This option provides increased security.

This chapter introduces the network database service called Yellow Pages (YP) and discusses the following topics:

- Overview of the Yellow Pages service
- How the YP service operates

1.1 Overview of the Yellow Pages Service

The Yellow Pages Service is a distributed database lookup service for sharing information between systems on a network.

Information distributed by YP is stored in sets of a database files called maps. A named set of YP maps is known as a domain. A domain is a distinct network namespace and describes an area under the administrative control of YP. You can think of a YP domain as a set of systems that share the same set of files, the YP maps. A YP domain is different from an Internet or sendmail domain.

YP is based on a client/server model. Unlike other client/server relationships, such as NFS, YP clients and servers are not machines, but are processes. YP domains are organized into a master server, one or more slave servers, and numerous clients. The master server is the only one whose database can be modified. Server databases are automatically updated to keep their information consistent with the master database.

Each domain can have only one master server. A YP server holds all the maps of a YP domain in a subdirectory of `/etc/yp` which is named after the domain. For example, if the name of the domain is `market`, maps for the `market` domain are in `/etc/yp/market`. You can determine your YP domain by executing the `domainname` command.

Note

The `/etc/yp` directory is symbolically linked to the `/var/yp` directory.

Each system on the network belongs to a default domain determined by an entry in the `/etc/rc.local` file at boot time with the `domainname` command.

The YP maps contain the information that YP serves, and each map contains a set of keys and associated values. For example, the `hosts` map contains all host names on a network as keys and the corresponding Internet addresses as values. Each YP map has a map name, used by programs to access data in the map.

Programs must know the format of the data in the map. Most maps are derived from ASCII files such as `/etc/passwd`, `/etc/group`, `/etc/hosts`, and `/etc/networks`. Maps are implemented by `dbm` files located in the `/var/yp/domainname` directory on the YP servers. See `dbm(3x)` in the ULTRIX Reference Pages for more information.

Sometimes YP clients are served by YP servers on the same system, and other times by YP servers running on a different system. If a remote system running a YP server process exits, client processes can obtain the YP service from another system. This feature makes the YP service almost always available.

In the YP environment, only a few systems have a set of YP databases. YP makes the database set available over the network. A YP client system runs YP processes and requests data from databases on other systems. Two kinds of systems have databases: a YP slave server and a YP master server. For any map, one YP server is designated the master, and all changes to the YP map should be made on that system. The changes then propagate from master to slaves.

YP clients do not need to know the location of data, or how it is stored. Instead, they use a network protocol to communicate with a database server that knows those details.

1.2 How the YP Service Operates

The YP service provides a way for a network manager to maintain consistency among selected system administrative files on all the systems in a YP domain.

The YP service maintains network-wide databases, such as `/etc/hosts`. The servers throughout the network contain copies of the YP maps.

When any system on the network wants to look up something in `/etc/hosts`, it makes a remote procedure call (RPC) to one of the servers to get the information. One server is the master — the only server whose database may be modified. The other servers are slaves, and they are periodically updated so that their information is synchronized with that of the master server.

YP can serve any number of files, including some that reside in the `/etc` directory, such as `/etc/passwd` and `/etc/networks`. In addition to these, users can add their own files to YP.

The `/etc/svc.conf` file defines the order in which to query the name services running on your system. It is a mandatory system file that is created when you install the ULTRIX software.

The following sections describe various aspects of how YP accomplishes its services.

1.2.1 Naming Domains

A domain is a collection of systems that shares a set of YP maps and shares the same YP master server. The `domainname` command tells you the name of the system's domain. The `getdomainname` system call returns the name of the domain to the program that called it.

Data is stored in the `/var/yp/domainname` directory. A system can contain data for several different domains. Recall that the `/etc/yp` directory is symbolically linked to `/var/yp`.

1.2.2 Storing Data

YP maps store data in dbm files. For example, the YP map for `/etc/hosts` in the domain `market` might be stored in these files:

```
/var/yp/market/hosts.byaddr.dir
/var/yp/market/hosts.byaddr.pag
/var/yp/market/hosts.byname.dir
/var/yp/market/hosts.byname.pag
```

The `makedbm` command takes an ASCII file such as `/etc/hosts` and converts it into dbm files suitable for use by YP. However, system administrators should use the Makefile script in the `/var/yp` directory to create YP map files. The Makefile script then calls `makedbm`. See `ypmake(8yp)` in the ULTRIX Reference Pages for further information on rebuilding YP databases.

1.2.3 Default YP Files

YP serves the following default database files:

- `/etc/hosts`
- `/etc/passwd`
- `/etc/group`
- `/etc/networks`
- `/etc/rpc`
- `/etc/services`
- `/etc/protocols`
- `/etc/netgroup`

Library routines such as `getpwent`, `getgrent`, and `gethostent` work with YP according to how the `/etc/svc.conf` file has been set up. C programs that call these library routines will need to be relinked to function correctly.

Note

If YP is running, the library routines such as `getpwent`, `getgrent`, and `gethostent`, cause entries served by YP to be returned in the order the data appears in the YP map. This returned order is not necessarily the same as the original ASCII files. See the `yp_next` function described in `ypclnt(3yp)` in the ULTRIX Reference Pages for further information.

The following sections discuss each of the default files.

1.2.3.1 The `/etc/hosts` File – The `/etc/hosts` file is stored as two different files in YP. The first, `hosts.byname`, is indexed by host name. The second, `hosts.byaddr`, is indexed by Internet address. The `hosts` YP map expands into the four YP map files, with the suffixes `.pag` and `.dir`.

When a user program calls the library routine `gethostbyname`, a single RPC call to a server retrieves the entry from the `hosts.byname` file. Similarly, `gethostbyaddr` retrieves the entry from the `hosts.byaddr` file. If YP is not running (which you can cause by commenting out the `ypbind` entry in the `/etc/rc.local` file), then `gethostbyname` reads the `/etc/hosts` file.

Maps sometimes have more than one name. Although the `ypcat` command is a general YP map print program, it knows about the standard files in YP. Thus, the

command `yycat hosts` is translated into `yycat hosts.byaddr`, because there is no file called `hosts` in YP. Type the following command for a list of expanded names:

```
# yycat -x
```

1.2.3.2 The /etc/passwd File – The `/etc/passwd` file is similar to the `/etc/hosts` file. It exists as two separate files, `passwd.byname` and `passwd.byuid`. The `yycat` program prints it, and `yymake` updates it. Unlike the `gethostbyaddr` and the `gethostbyname` library functions, however, the `getpwent` function reads the local `/etc/passwd` file and interprets the YP special characters: plus (+), minus (-), and at (@).

A plus (+) entry is used to include the entries from the YP `passwd` map. A minus (-) entry, on the other hand, is used to prevent this user from logging in to the system regardless of the YP `passwd` map. Use the (@) character in conjunction with plus and minus entries to either include or exclude members of the network group specified. See `netgroups(5yp)` in the ULTRIX Reference Pages for further information.

If you wrote a program using `getpwent` to print all the entries from your password file, it would print a virtual password file. Rather than printing + and -, it would print whatever entries the local password file included from the YP map.

If you are running YP and need to change a password, you must change the password in the YP map using the `yypasswd` command, unless you need to modify an entry in the local `/etc/passwd` file. The `yypasswd` command has the same user interface as the `passwd` command, but works only if the `yypasswd` daemon is running on the YP master server.

Note

The `passwd` command does not change the password YP map. It changes only the local password file `/etc/passwd` and not the YP master password file on the YP master server that is usually stored as `/var/yp/src/passwd`. See Chapter 3 for further information.

1.2.3.3 Other YP Files – Of the other files used by YP in the `/etc` directory, `/etc/group` is treated like `/etc/passwd`, in that the `getgrent` library routine only consults the YP `group` map if explicitly told to do so by plus (+) or minus (-) entries in the `/etc/group` file. The files `/etc/networks`, `/etc/rpc`, `/etc/services`, `/etc/protocols`, and `/etc/netgroup` are treated like `/etc/hosts`: for these files, the library routines go directly to YP, without consulting the local files.

Any plus or minus (+,-) entries have no effect in the `/etc/networks`, `/etc/hosts`, `/etc/rpc`, `/etc/protocols`, `/etc/services`, and `/etc/netgroup` files.

1.2.4 The /etc/svc.conf File

The `/etc/svc.conf` file defines the order in which to query the name services running on your system. It is a mandatory system file that is created when you install the ULTRIX software. If you want to use YP, you must edit the `/etc/svc.conf` file with the necessary database and service order information.

The following is a typical entry in the `/etc/svc.conf` file:

```
passwd=local,bind
```

This entry tells the system to search first locally for password information. If it cannot find the information locally, the system then queries a YP server.

Note

It is recommended that you list `local` as the first service for all databases to increase speed and efficiency.

You can specify any of the following databases in the `/etc/svc.conf` file:

- `aliases`
- `auth`
- `group`
- `hosts`
- `netgroup`
- `networks`
- `passwd`
- `protocols`
- `rpc`
- `services`

See Chapter 2 for information on editing `/etc/svc.conf` and see `svc.conf(5)` in the ULTRIX Reference Pages.

This chapter explains how to set up the Yellow Pages (YP) service on your system YP manually and how to modify an existing YP environment.

This chapter discusses the following topics:

- Overview of setting up a YP server and client
- Prerequisite information for setting up a YP server or client
- Setting up YP manually
- Altering YP client files
- Creating and modifying YP maps
- Propagating YP maps

For information on setting up the Yellow Pages service automatically by using the `ypsetup` command, see the *Guide to System and Network Setup*.

2.1 Overview of Setting Up a YP Server and Client

To set up a system as a server, the system must contain the YP maps and must also run the YP daemons `/etc/portmap` and `/usr/etc/ypserv`. The YP master server also must run the `/usr/etc/rpc.yppasswdd` daemon. The `ypsetup` command places an entry in the `/etc/rc.local` file to start these daemons automatically at boot time.

Any system, including a YP server, can act as a YP client by running the `/etc/ypbind` daemon. A YP client gets its information from a YP server if the information is not in the client's local files. If a YP client cannot find the information in its local files, it makes an RPC call to a YP server and gets the information from a YP map.

The `ypbind` daemon remembers the name of the YP server. When a client boots, `ypbind` broadcasts a request to the Ethernet wire asking for the name of a YP server.

Similarly, `ypbind` broadcasts a request for the name of a new YP server if the old server has gone off the network for any reason. If `ypbind` is run with the `-S` security option, no requests are broadcast. Only directed requests to the specified list of servers are issued. The `ypwhich` command gives the name of the server that the `ypbind` daemon currently points to.

Users on a YP client can use the `ypcat` and `ypmatch` commands to print data from a YP map. The following command prints the information in the YP `hosts` map:

```
# ypcat hosts
```

Similarly, the following command prints the information in the YP `passwd` map:

```
# ypcat passwd
```


To look for someone's password entry, you need to use either the `ypcat` or the `ypmatch` command. For example, to obtain the password entry for a user named `jane`, use one of the following command lines:

```
# ypcat passwd | grep jane
# ypmatch jane passwd
```

2.2 Prerequisite Information

Before you can set up YP on your system, your system must be in multiuser mode with the `/usr` file system mounted, and your system must be established on a local area network. In addition, you must know the answers to the following questions:

- What is the default YP domain name for your system?
- Will your system be the YP master server on the domain?

If your system will be the master server:

- You must be sure there is no other master already existing on the domain.
- You must know the names of the YP slave servers on your domain. To keep the YP maps consistent across all YP servers, the YP master server maintains a list of slave servers to send the updated copies of the maps using the `yppush` command. The `ypsetup` command adds the names you enter to the master server's list.

- Will your system be a YP slave server?

If your system will be a YP slave server, be sure there is a YP master server already on the domain. Otherwise, you will not be able to initialize the YP maps for your system.

- Will your system be a YP client?

You must be sure there is at least one other system on the network configured as either a YP master or slave server. Otherwise, you will not be able to access the YP maps.

Once you have the required information, you are ready to set up the YP service.

2.3 Setting Up YP Manually

For an interactive setup with default answers, follow the automatic YP setup procedure.

This section describes how to set up your system manually as a YP master server, slave server, or client and helps you to understand how YP works.

Both the client and the server must be connected to an Internet network for YP to be able to run.

2.3.1 Setting Up a YP Master Server

The following files must contain the data that will be served to the YP clients on the domain:

- /etc/group
- /etc/hosts
- /etc/networks
- /etc/passwd
- /etc/protocols
- /etc/rpc
- /etc/services
- /etc/netgroup

If any of these files are not up to date, edit them and add the correct entries. For example, the /etc/passwd file must contain an entry for each user on the domain that the YP master server will serve.

In addition, be sure the /etc/netgroup file is complete. If you do not have an /etc/netgroup file, create an empty one by typing the following command:

```
# cp /dev/null /etc/netgroup
```

See netgroup(5yp) in the ULTRIX Reference Pages for further information.

By default, the YP maps for the YP master server are constructed from the files residing in the /etc directory. If you want to modify the /etc files to contain only the local entries for the master server, create a directory such as /var/yp/src. Then copy the master copies of the files to it. You should make all future modifications there.

If you plan to run make, ensure that the netgroup file is in /etc. If it is not in /etc, the make command will not find the netgroup file.

For further information, see make(1) in the ULTRIX Reference Pages.

To set up your system manually as a YP master server, follow these steps:

1. Establish the YP domain
2. Build the default YP maps
3. Start the YP server daemons
4. Modify the default YP files
5. Start the YP password server
6. Edit the /etc/rc.local file
7. Modify the /etc/svc.conf file
8. Create the YP servers map

The following sections describe these steps.

2.3.1.1 Establish the YP Domain – Set the domain name and create the domain directory. In the following example, the domain name is set to market:

```
ypmaster# /bin/domainname market
ypmaster# mkdir /var/yp/market
```

2.3.1.2 Build the Default YP Maps – Build the default YP maps:

```
ypmaster# cd /var/yp
ypmaster# make NOPUSH="Y"
```

2.3.1.3 Start the YP Server Daemons – If the RPC port mapper is not running, start it:

```
ypmaster# /etc/portmap
```

After you have started the RPC port mapper, or if the port mapper is already running, start the ypserv daemon:

```
ypmaster# /usr/etc/ypserv
```

2.3.1.4 Modify the Default YP Files – If your system will be acting as a YP client in addition to being a YP master server, create a directory such as /var/yp/src. Then, copy the default files to that directory (/etc/group), /etc/hosts, /etc/netgroup, /etc/networks, /etc/passwd, /etc/protocols, /etc/rpc, and /etc/services). Edit the original default files in /etc as described in Section 2.5.

If you change the directory for all the default files, edit the Makefile script and modify the DIR argument so that it specifies the new directory. For example, if the original directory was /etc and the new directory is /var/yp/src, here is the new argument to the Makefile script:

```
DIR=/var/yp/src
```

If you change the directory for some of the default files, but not all of them, edit only the Makefile DIR arguments for those particular files.

If you change the directory for the password file, in addition to editing the Makefile script, be sure to edit the /etc/rc.local file to reflect the new directory. For example, if the new directory for /etc/passwd is /var/yp/src, be sure the following entry is in the /etc/rc.local file:

```
/usr/etc/rpc.yppassdd /var/yp/src/passwd \  
-m passwd DIR=/var/yp/src
```

Note

If you want to modify the YP master files at any time after changing a directory for a default file, modify the files in the directory you created to store YP files, such as /var/yp/src. Then, run the makedbm command. The files in /etc are the YP master server's local files and do not contain entries for the YP clients on the domain.

Start the ypbind daemon:

```
ypmaster# /etc/ypbind [-S domainname,server1,...,server4]
```

Note that you can specify a maximum of four servers with this command and that the servers must be listed in the /etc/hosts file.

For more information on using the -S option, see Chapter 3.

- 2.3.1.5 Start the YP Password Server** – To allow YP clients to change their YP password entries, start the YP password server daemon. For example, if the master version of the `passwd` file is stored as `/var/yp/src/passwd`, type the following command:

```
ypmaster# /usr/etc/rpc.yppasswdd /var/yp/src/passwd \  
-m passwd DIR=/var/yp/src &
```

- 2.3.1.6 Edit the `/etc/rc.local` File** – Add an entry to the `/etc/rc.local` file to set up the default YP domain name, using this format:

```
/bin/domainname domainname
```

You also need to add entries for the `/etc/portmap`, `/usr/etc/ypserv`, `/usr/etc/rpc.yppasswdd`, and `/etc/yplib` daemons. For example, the entry for `ypserv` should look like this:

```
if [ -f /etc/portmap -a -f /usr/etc/ypserv ]; then  
    /usr/etc/ypserv; echo -n ' ypserv' >/dev/console  
fi
```

On subsequent reboots, the YP service automatically starts from the `/etc/rc.local` file.

Note

The order in which the entries appear in the `/etc/rc.local` file determines the order in which the services are started when the system is brought to multiuser mode. Be sure that all YP entries precede any NFS daemons or other service daemons, such as `lpd` for the printer service in this file. Also, be sure that the entry for the domain name precedes the entries for the YP daemons.

- 2.3.1.7 Create the YP Servers Map** – Use the `makedbm` command to create the YP servers map. For example, if the domain name is `market` and the YP slave servers for the domain are `osprey` and `nuthatch`, type the following commands:

```
ypmaster# cd /var/yp  
ypmaster# makedbm - market/ypservers  
osprey  
nuthatch  
<CTRL/D>
```

2.3.2 Modify the `/etc/svc.conf` File

You need to modify the `/etc/svc.conf` file. The `/etc/svc.conf` file controls the order in which database services, such as YP, are used.

- 2.3.2.1 Edit the `/etc/svc.conf` File with `svcsetup`** – The `svcsetup` command allows you to print and modify the database selections in the `/etc/svc.conf` file on the current system. You must modify this file when you are adding or removing a naming service, such as Yellow Pages or BIND/Hesiod.

Run the `secsetup` command if you want to change the security parameters. Changes take effect immediately.

To run the `svcsetup` command, you must be logged on as superuser. Type the following:

```
# svcsetup
```

The `svcsetup` command then steps through the setup procedure in the following manner:

1. Displays a menu asking whether you want to modify the existing `svc.conf` file, print the default settings to the screen, or exit the `svcsetup` command.

Select the modify option.

2. Lists the database services that you can modify. Each database is assigned a corresponding number by the system. Select the databases that you want to modify by listing the number of each database, followed by a space.

For example, to modify the services that access the `aliases` and `group` databases, respond to the prompt as follows:

```
Change Menu for the /etc/svc.conf file
```

```
aliases          => 0
auth             => 1
group           => 2
hosts           => 3
netgroup        => 4
networks        => 5
passwd          => 6
protocols       => 7
rpc             => 8
services        => 9
```

```
all of the above => 10
none of the above => 11
```

```
Enter your choice(s): 0 2
```

Press the Return key when you are through listing the databases.

If you opt not to edit any databases, the `svcsetup` command exits.

3. Displays a menu of the possible combinations of naming services that you can run on your system, with a number corresponding to each combination.

```
local           => 1
yp              => 2
bind            => 3
local,yp        => 4
local,bind      => 5
yp,local        => 6
bind,local      => 7
```

After the `svcsetup` command lists the recommended naming service combinations, it prompts you to specify the naming service order you want for each of the databases you are changing. Enter the number that corresponds to the new naming service order that you want to run.

The current service order setting is displayed in brackets ([]).

For example, if you chose to modify the service order for the `aliases` and `group` databases, the `svcsetup` command prompts you in the following manner:

```
Enter the naming service order for the "aliases" database [5]: 1
```

```

local          => 1
yp             => 2
bind          => 3
local,yp      => 4
local,bind    => 5
yp,local      => 6
bind,local    => 7

```

Enter the naming service order for the "group" database [5]:

```

local          => 1
yp             => 2
bind          => 3
local,yp      => 4
local,bind    => 5
yp,local      => 6
bind,local    => 7

```

The [5] indicates that the default entry in the `svc.conf` file for the aliases and group databases is `local,bind`.

4. Prints an informational message to the screen that it is updating the `svc.conf` file, and exits.

Note

Resolving a name locally is always faster than using a name service. Therefore, as you add name services, such as YP and BIND, to your system, place the local service first in the `svc.conf` file.

Always have the local service selected for the `passwd` and `hosts` database.

Only `local` and `bind` are valid for the `auth` database and only `yp` is valid for the `netgroup` database.

2.3.2.2 Editing the `/etc/svc.conf` File Manually – You must be logged onto your system as superuser to edit the `svc.conf` file. Invoke an editor, and add or modify the database entries. Each entry should be on a new line and of the following form:

```
database=service,service
```

Although white spaces are permitted after commas, and new lines, they are not required, however.

See `svc.conf(5)` in the ULTRIX Reference Pages for further information.

2.3.3 Setting Up a YP Slave Server

The local area network must be established before you can set up your system as a YP slave server. In particular, you must be able to copy files from the YP master server to the slave server, using the `rcp` command. There must also be a YP master server on the network running the `ypserv` daemon for the domain.

To set up your system manually as a YP slave server, follow these steps:

1. Establish the YP domain
2. Obtain copies of the YP maps

3. Start the YP server daemons
4. Modify the default YP database files
5. Edit the `/etc/rc.local` file
6. Modify the `/etc/svc.conf` file
7. Edit the `/usr/lib/crontab` file
8. Add the new slave server to the domain

The following sections describe these steps.

- 2.3.3.1 Establish the YP Domain** – Set the domain name and create the domain directory. For example, if your domain name is `market`, type the following:

```
ypslave# /bin/domainname market
ypslave# mkdir /var/yp/market
```

- 2.3.3.2 Obtain Copies of the YP Maps** – Run the YP transfer command, `ypxfr`, for each YP map your system will serve. For example, to run `ypxfr` on a system with a YP master called `orville` on a domain called `market` for the `/etc/passwd` map, type the following:

```
ypslave# ypxfr -h orville -c -d market passwd.byname
ypslave# ypxfr -h orville -c -d market passwd.byuid
```

To find a list of the YP maps that your system can serve, look on the YP master in the `/var/yp/domainname` directory. For example: `/var/yp/market`.

- 2.3.3.3 Start the YP Server Daemons** – If the RPC port mapper is not running, start it:

```
ypslave# /etc/portmap
```

After the RPC port mapper is running, start the `ypserv` daemon:

```
ypslave# /usr/etc/ypserv
```

- 2.3.3.4 Modify the Default YP Database Files** – If your system will act as a client, in addition to being a YP slave server, edit the default database files as described in Section 2.5. Then, start the `ypbind` daemon:

```
ypslave# /etc/ypbind [-S domainname, server1, ..., server4]
```

Whenever you start `ypbind`, you can use the `-S` option for added security.

- 2.3.3.5 Edit the `/etc/rc.local` File** – Add an entry to the `/etc/rc.local` file to set up the default YP domain name. For example, if the domain name is `market`, the entry should look like this:

```
/bin/domainname market
```

You also need to add entries for the `/etc/portmap`, `/usr/etc/ypserv`, and `/etc/ypbind` daemons. For example, the `/etc/portmap` daemon has the following entry:

```
if [ -f /etc/portmap ]; then
    /etc/portmap; echo -n ' portmap' >/dev/console
fi
```

The entry for the ypbind daemon should look like this:

```
if [ -f /etc/portmap -a -f /etc/ypbind ]; then
    /etc/ypbind [-S domainname, server1,...,server4]; \
    echo -n ' ypbind' >/dev/console
fi
```

On subsequent reboots, the YP service automatically starts from the /etc/rc.local file.

Note

Be sure that the entry for the domain name precedes the entries for the YP daemons in the /etc/rc.local file.

If your system is also running the Network File System (NFS), be sure the YP entries precede the NFS entries in the /etc/rc.local file.

2.3.3.6 Modify the /etc/svc.conf File – See Section 2.3.2.1 for information about the /etc/svc.conf file.

2.3.3.7 Edit the /usr/lib/crontab File – To allow your YP slave server to receive updated copies of the YP master server's YP maps, place ypxfr command entries in the /usr/lib/crontab file. For examples of crontab entries, look at these files:

```
/etc/yp/ypxfr_1perday
/etc/yp/ypxfr_2perday
/etc/yp/ypxfr_1perhour
```

See cron(8) and ypxfr(8yp) in the ULTRIX Reference Pages for further information.

2.3.3.8 Add the New Slave Server to the Domain – To add the new slave server to the domain, follow the directions in Section 2.7.1.

2.3.4 Setting Up a YP Client

The local area network must be established before you can set up your system as a YP client. In addition, there must be a YP server on the network running the ypserv daemon for the domain.

To set up your system manually as a YP client, follow these steps:

1. Establish the YP domain
2. Start the YP port mapper daemon
3. Modify the default YP database files
4. Edit the /etc/rc.local file
5. Modify the /etc/svc.conf file

The following sections describe these steps.

2.3.4.1 Establish the YP Domain – Set the domain name with the `domainname` command, using this format:

```
/bin/domainname domainname
```

For example, to set the domain name for the domain `market`, type the following:

```
ypclient# /bin/domainname market
```

2.3.4.2 Start the YP Port Mapper Daemon – If the RPC port mapper is not running, start it:

```
ypclient# /etc/portmap
```

2.3.4.3 Modify the Default YP Database Files – Edit the default database files as described in Section 2.3.3.4. Then, start the `ypbind` daemon:

```
ypclient# /etc/ypbind
```

As an alternative, you can issue the following command:

```
/etc/ypbind [-S domainname, server1, ..., server4]
```

Recall that the `-S` option is for security. For more information, see Chapter 3.

2.3.4.4 Edit the `/etc/rc.local` File – Add an entry to the `/etc/rc.local` file to set up the default YP domain name, using this format:

```
/bin/domainname domainname
```

You also need to add entries for the `/etc/portmap` and `/etc/ypbind` daemons. For example, the entry for `ypbind` should look like this:

```
if [ -f /etc/portmap -a -f /etc/ypbind ]; then
    /etc/ypbind [-S domainname, server1, ..., server4]; \
    echo -n ' ypbind' >/dev/console
fi
```

On subsequent reboots, the YP service automatically starts from the `/etc/rc.local` file.

Note

Be sure that the entry for the domain name precedes the entries for the YP daemons in the `/etc/rc.local` file.

If your system is also running the Network File System (NFS), be sure the YP entries precede the NFS entries in the `/etc/rc.local` file.

2.3.4.5 Modify the `/etc/svc.conf` File – See Section 2.3.2.1 for information about the `/etc/svc.conf` file.

2.4 Altering YP Client Local Files

All YP clients on the network should be updated to use the YP master's versions of the YP maps, rather than their potentially out-of-date local files. This policy is enforced by running a `ypbind` process on the client system (including systems that might be running YP servers), and by modifying or eliminating the following files:

`/etc/group`, `/etc/hosts`, `/etc/hosts.equiv`, `/etc/netgroup`, `/etc/networks`, `/etc/passwd`, `/etc/protocols`, `/etc/rpc`, `/etc/services`, and `/.rhosts`. The following sections discuss how to treat each of these files.

2.4.1 The networks, protocols, rpc, services, and netgroup Files

The following files are not needed on any YP client:

- `/etc/networks`
- `/etc/protocols`
- `/etc/rpc`
- `/etc/services`
- `/etc/netgroup`

If you would prefer to keep them, you can leave them where they are or you can move them to backup files. The following example shows how to move the `/etc/networks` file to `/etc/networks.old`:

```
# mv /etc/networks /etc/networks.old
```

2.4.2 The `/etc/hosts.equiv` File

The YP service does not serve the `/etc/hosts.equiv` file. However, you can add escape sequences to activate YP. This reduces problems with `rlogin` and `rsh` that are sometimes caused by different `/etc/hosts.equiv` files on two systems.

To let anyone log in to a system, you could edit `/etc/hosts.equiv` so that it contains a single line with only the plus character (+) on it, which matches any host name.

However, you can exercise more control over logins by using lines of the following form:

```
+@trusted_group1  
+@trusted_group2  
-@untrusted_group
```

Each of the names to the right of the at character (@) is assumed to be a network group name, defined in the global network group YP map that YP serves. For example, if two trusted groups are `staff` and `users`, and an untrusted group is `guest`, these are the appropriate `/etc/hosts.equiv` entries:

```
+@staff  
+@users  
-@guest
```

If no escape sequence is used, only the entries in `/etc/hosts.equiv` are used; YP is not used.

2.4.3 The `/.rhosts` File

The YP service does not serve `/.rhosts` files. The format of the `/.rhosts` file is identical to that of `/etc/hosts.equiv`. However, because the `/.rhosts` file controls remote root access to the local system, you should restrict access to it. Make the list of trusted hosts explicit or use the network group names.

2.4.4 The /etc/hosts File

The `/etc/hosts` file must contain entries for the local host's name and the local loopback name. Otherwise, the system could hang while coming up to multiuser mode. The entries in the `/etc/hosts` file are accessed at boot time when the YP service is not yet available. After the system is running, and after the `yplib` process is up, the `/etc/hosts` file is not accessed.

The following example shows the `hosts` file for YP client `orville`:

```
127.0.0.1      localhost
192.9.1.87    orville      # John Q. Random
```

2.4.5 The /etc/passwd File

The `/etc/passwd` file should contain entries for root and the primary users of the system and an escape entry to force the use of YP.

A sample YP client's `/etc/passwd` file looks like:

```
root:6H2/WWVZnlFgM:0:1:Bossman with a C shell:/:/bin/csh
operator::0:28:Operator:/opr:/opr/opser
daemon:*:1:1:Mr. Background:/:
sys:xzuEOV1LjYpJM:2:3:Mr. Kernel:/usr/sys:
bin:xcvjW4alfaUn:3:4:Mr. Binary:/bin:
uucp:Nologin:8:8:USENET New System:/usr/spool/netnews:
+:
```

The last line is the escape entry that informs the library routines to use the YP service rather than give up the search. Entries that exist in the local files, such as `/etc/passwd`, mask analogous entries in the YP maps. In addition, earlier entries in the file mask later entries with the same user name or the same user identification (UID).

Note

It is important that the `+` entry always be last. Any entries in the `/etc/passwd` file placed after the `+` entry are ignored because the library routines go directly to the YP map.

If you run the `netsetup` or `uucpsetup` commands after YP is running, check the `/etc/passwd` file to be sure the `+` entry is last.

If you want to run `uucp` throughout your YP domain, you must run `uucpsetup` on the YP master server and then remake the password maps:

1. Run `uucpsetup` on the YP master server.
2. Edit the `/etc/passwd` file. If `/etc/passwd` is the YP password map, place the `+` entry at the end of the file. Otherwise, move the entries that `uucpsetup` appended after the `+` entry to the YP password map (usually `/var/yp/src/passwd`).
3. Make the YP password map:

```
# cd /var/yp
# make passwd
```

See Sections 2.6 and 2.7 for further information about modifying and propagating YP maps.

2.4.6 Reducing the /etc/group File to a Single Line

You can reduce the `/etc/group` file to a single line by using the following format:

+:

This line forces all translation of group names and group identifications to be made by the YP service.

2.5 Modifying and Creating YP Maps

You should modify the YP maps that YP serves on the YP master server, which then propagates copies of its modified databases to the YP slave servers. You can modify the maps you expect to change most frequently, such as `passwd`, by first editing the ASCII file and then running `make` on `/var/yp/Makefile`.

For example, to add a YP user, follow these steps:

1. Edit the YP master server's `passwd` file and add an entry for the new user. If you have chosen to move the default files, as noted in Section 2.3.1.4, edit the `/var/yp/src/passwd` file. Otherwise, edit the file `/etc/passwd`.
2. Type the following commands:

```
# cd /var/yp
# make passwd
```

Whether you use the `Makefile` command in `/var/yp` or some other procedure, the goal is the same; a new pair of `dbm` files must be created in the domain directory on the YP master server. For further information, see `ypmake(8yp)` in the *ULTRIX Reference Pages*.

You can manually edit the following nonstandard YP maps:

- Maps that are specific to the applications of a particular vendor or site
- Maps that you expect to change rarely
- Maps for which no ASCII form exists, such as maps that did not exist before YP was set up.

Use the `makedbm` command with the `-u` option to disassemble the YP maps into a form that can be modified using standard tools, such as `awk`, `sed`, or `vi`. Then, build a new version of the YP maps using the `makedbm` command. You can do this manually in two ways:

- You can redirect `makedbm` output to a temporary file that can be modified and then piped back into `makedbm`.
- You can operate on the `makedbm` output within a pipeline that feeds directly into `makedbm` again. This is appropriate if you can update the disassembled map by modifying it with `awk` or `sed`, or by appending to it with `cat`.

For example, suppose that you want to create a nonstandard YP map, called `mymap`, and that you want it to consist of key-value pairs in which the keys are strings such as `a1`, `b1`, `c1`, and so forth, and the values are `ar`, `br`, `cr` and so forth. There are two procedures that you can follow when creating maps. In one, you use an existing ASCII file as input; in the other, you use standard input.

- If there is an existing ASCII file, you can create the YP map for it using the `makedbm` command. For example, if the file is named `/var/yp/src/mymap.asc` and resides on the YP master server called

ypmaster for a domain called `market`, you can create the YP map by typing the following commands:

```
# cd /var/yp/src
# /var/yp/makedbm mymap.asc ../market/mymap
```

To update the YP map, remember to modify the ASCII file first. Modifications made to the map, but not also made to the ASCII file would become lost. Make the modification like this:

```
# cd /var/yp/src
# vi mymap.asc
# /var/yp/makedbm mymap.asc ../market/mymap
```

- If there is no original ASCII file, you can create the YP map by typing input like the following. In this example, the default domain is `market`:

```
# cd /var/yp/market
# /var/yp/makedbm - mymap
al ar
bl br
cl cr
<CTRL/D>
```

If you need to modify that map, you can use `makedbm` to create a temporary ASCII intermediate file, which you can edit using standard tools. For example:

```
# cd /var/yp/market
# /var/yp/makedbm -u mymap > mymap.tem
```

You can now edit `mymap.tem` so that it contains the correct information. To create a new version of the YP map, type the following commands:

```
# /var/yp/makedbm mymap.tem mymap
# rm mymap.tem
```

Support on the YP slave servers for propagation of the new maps consists of appropriate entries either in `/usr/lib/crontab` or in one of the `ypxfr` shell scripts mentioned in Section 2.3.3.7. To get an initial copy of the map, you can run `ypxfr` manually on each of the slave servers. The map must be globally available before clients begin to access it. If the map is available from some YP servers, but not all, you get unpredictable behavior from client programs.

After you have modified the YP maps, you need to propagate them to the other servers on the domain. See Section 2.6 for a description of how to propagate YP maps.

2.6 Propagating YP Maps

To propagate a YP map is to move it from place to place, usually from the master YP server to a slave. You can propagate YP maps from the master server to the slave in three different ways, as described in the following sections.

2.6.1 Using `make` to Propagate YP Maps

You can propagate the default YP maps from the master YP server using the `make` command, as described in Section 2.6.1. For example, to propagate the `hosts` file, type the following:

```
ypmaster# cd /var/yp
ypmaster# make hosts
```

The `Makefile` script automatically runs the `yppush` command to push the YP map from the YP master server to the slave servers on the domain. See `yppush(8yp)` in the ULTRIX Reference Pages for further information about how `yppush` propagates the YP maps.

The `yppush` command uses the `ypservers` YP map to obtain a list of YP servers in your domain. To each of the named YP servers, it sends a Transfer Map request. The `ypserv` command forks a copy of `ypxfr`, by invoking it with the `-C` option, passing the information it needs to identify the map, and calling back the initiating `yppush` process with a summary status.

2.6.2 Using `makedbm` to Propagate YP Maps

You can propagate nondefault YP maps from the YP master server using `makedbm` and then `yppush`, as described in Section 2.6.1 For example, to propagate a nondefault YP map called `sales.asc` that resides in `/var/yp/src` on the domain `market`, type the following:

```
ypmaster# cd /var/yp/src
ypmaster# /var/yp/makedbm sales.asc ../market/sales
ypmaster# yppush sales
```

The `yppush` command uses the `ypservers` YP map to obtain a list of YP servers in your domain. To each of the named YP servers, it sends a Transfer Map request. The `ypserv` command forks a copy of `ypxfr`, by invoking it with the `-C` option, passing the information it needs to identify the map, and calling back the initiating `yppush` process with a summary status.

2.6.3 Using `ypxfr` to Propagate YP Maps

After you have initialized a YP slave server, you can propagate the YP maps from the YP master server by running the `ypxfr` command from `cron` or by running it manually.

2.6.3.1 Running `ypxfr` from `cron` – Maps have differing rates of change; for instance, the `protocols` YP map might not change for months at a time, but the `passwd` YP map could change several times a day in a large organization. You can set up the `/usr/lib/crontab` file entries to run `ypxfr` periodically at a rate appropriate for any map in your YP database. The `ypxfr` command contacts the master server and transfers the map only if the master's copy is more recent than the local one.

To avoid needing a `crontab` entry for each map, you can group several maps with approximately the same change characteristics together in a shell script, which can be run from `/usr/lib/crontab`. Suggested groupings, mnemonically named, are in the `/var/yp` directory. The following files do daily YP map checks and/or updates: `ypxfr_1perhour`, `ypxfr_1perday`, and `ypxfr_2perday`. (If you set up a YP slave server with `ypsetup`, these entries are placed in `/usr/lib/crontab` automatically.) If the rates of change are inappropriate for your environment, you can modify or replace these shell scripts.

Run these shell scripts on each YP slave server in the domain. Also, alter the exact time of execution from one server to another to prevent the master from overloading. If you want the map transferred from a particular server other than the master, you can specify this using the `-h` option with the `ypxfr` command within the shell script.

Finally, you can check and transfer maps that have unique change characteristics by explicitly invoking `ypxfr` from within `/usr/lib/crontab`. For example:

```
15 2 * * * /var/yp/ypxfr_1perday
```

In this example, `/var/yp/ypxfr_1perday` is a script.

See `ypxfr(8yp)` and `cron(8)` in the ULTRIX Reference Pages for further information.

- 2.6.3.2 Running ypxfr Manually** – You can run `ypxfr` manually as a command. Typically, you do this only in exceptional situations, such as when setting up a temporary YP server to create a test environment or when quickly trying to update an out-of-date YP server's YP maps. For example, to propagate the `group` YP map, type the following:

```
ypslave# /usr/etc/ypxfr group.byname
ypslave# /usr/etc/ypxfr group.bygid
```

Be sure you run `ypxfr` for each file making up the YP map.

Each of the `ypxfr` transfer attempts and results can be captured in a log file. If the file `/var/yp/ypxfr.log` exists, then the results are appended to it. There is no attempt to limit the log file length. To stop the information from accumulating in the log file, remove `/var/yp/ypxfr.log`. See `ypxfr(8yp)` in the ULTRIX Reference Pages for further information.

2.7 Modifying the YP Environment

This section describes how to modify the YP environment. The following topics are discussed:

- Adding YP servers to the domain
- Removing YP slave servers from the domain
- Changing a YP map's master server
- Adding users to a YP client

2.7.1 Adding YP Servers to the Domain

To add a YP slave server to the domain, begin by modifying the maps on the YP master server. If the new server is a host that has not been a YP server before, you must add the host name to the `ypservers` map in the master YP server's default domain. For example, the following sequence adds a server named `osprey` to domain `market`:

```
ypmaster# cd /var/yp
ypmaster# (/var/yp/makedbm -u market/ypservers ; echo osprey)\
|/var/yp/makedbm - tmpmap
ypmaster# mv tmpmap.dir market/ypservers.dir
ypmaster# mv tmpmap.pag market/ypservers.pag
ypmaster# yppush ypservers
```

Note

The second command in this example is on two lines. You can type these lines as one long command even if the line wraps on your screen, or you can use the backslash escape character (`\`), as shown. You cannot type half the command (without the backslash), press the Return key, and type the second half.

The new host address should also be in the `hosts` YP map. If it is not, add an entry for this host to the YP master server's master `hosts` file and then run `make`. For example, if the `hosts` file is stored as `/etc/hosts`, these are the commands:

```
ypmaster# vi /etc/hosts
.
.
ypmaster# cd /var/yp
ypmaster# make hosts
```

Set up the new YP slave server's databases by copying the files from the YP master server. To do this, log in to the new YP slave and set up the YP environment as described in the *Guide to System and Network Setup* for the automatic procedure and Section 2.3.3 for the manual procedure.

After you have added a server to the domain, you need to propagate the YP maps from the YP master server to the new slave. See Section 2.6 for a description of how to propagate YP maps.

2.7.2 Removing YP Slave Servers from the Domain

To remove a YP slave server from the domain, begin by modifying the maps on the YP master server. You need to remove the server's host name from the `ypservers` map in the master YP server's default domain. For example, the following sequence removes a server named `osprey` from domain `market`:

```
ypmaster# cd /var/yp
ypmaster# /var/yp/makedbm -u market/ypservers | \
grep -v osprey | /var/yp/makedbm - tmpmap
ypmaster# mv tmpmap.dir market/ypservers.dir
ypmaster# mv tmpmap.pag market/ypservers.pag
ypmaster# yppush ypservers
```

Note

The second command in this example is on two lines. You can type these lines as one long command even if the line wraps on your screen, or you can use the backslash escape character (`\`), as shown. However, you cannot simply type half the command (without the backslash), press the Return key, and type the second half.

2.7.3 Changing a YP Map's Master Server

To change a YP map's master server to a different system, first build the map at the new master. Because the old YP master's name occurs as a key-value pair in the existing map, it is not sufficient to use an existing copy at the new master server or to send a copy there with `ypxfr`. The key must be reassociated with the new master's name. If the map has an ASCII source file, the current version should be present at the new master. Remake the YP map locally with the following sequence:


```
newmaster# cd /var/yp
newmaster# make salary.byhour
```

In this example, `salary.byhour` is the name of the YP map. The `/var/yp/Makefile` file must be set up correctly for the `make` command to work. If it is not, you should do it before doing anything else. In addition, go back to the old master (if it is to remain a YP server) and edit `/var/yp/Makefile` so that the `salary.byhour` map is no longer made there. To do this, comment out the section that made the map `salary.byhour` in the old master server's `/var/yp/Makefile`.

If the map only exists as a dbm file, you can re-create it on the new master by disassembling an existing copy from any YP server and running the disassembled version back through `makedbm`. For example:

```
newmaster# cd /var/yp
newmaster# ypcat -k salary.byhour |\
/var/yp/makedbm - market/salary.byhour
```

After making the map on the new master, you need to send a new copy of the map to the other YP slave servers. Do not use `yppush`, because the other slaves would try to get new copies from the old master, rather than the new one.

A typical method is to become superuser on the old master server and type the following:

```
oldmaster# /var/yp/ypxfr -h newmaster salary.byhour
```

Now that you have a new copy on the old master server, you can run `yppush`. The remaining slave servers will attempt to get the current version of the map from the old master server. When they do, they will get the new map, which names the new master as the current master.

2.7.4 Adding Users to a YP Client

To add a user to a YP client on the network, add an entry to the YP master server's password file and create a home directory on the new user's system as described in the following steps:

1. Edit the YP master server's `/etc/passwd` file
2. Update the YP map
3. Make a home directory
4. Set up the new user's environment
5. Propagate the updated YP map

The following sections describe these steps.

2.7.4.1 Edit the YP Master Server's `/etc/passwd` File – On the YP master server, add a new line to the master copy of the password file. If you are using the file `/etc/passwd` as the master copy, use the `vipw` command. The `vipw` command brings the password file into the `vi` editor and prevents anyone else from editing it until you are done:

```
ypmaster# /etc/vipw
```

Otherwise, edit the master copy. For example:

```
# vi /var/yp/src/passwd
```

The `passwd` file is a readable ASCII file with a one-line entry for each valid user on the system. Here is a sample `passwd` entry for a user named Jane Doe:

```
doe:fnuTqqab.6yec:444:10:Jane Doe:/usr/staff/does/bin/csh
```

See the *Guide to System Environment Setup* for a description of how to edit the `passwd` file to add a new user.

Note

The remote systems on the network recognize a user by the user identification (UID) number. Therefore, it is important that each user have the same UID on each of the systems on the network.

2.7.4.2 Update the YP Map – After you have updated the YP master server's password file and created a password for the new user, be sure to update the YP map by running `/var/yp/make` for `/etc/passwd`:

```
ypmaster# cd /var/yp
ypmaster# make passwd
```

You need to adjust the `make` command if the master copy of the `passwd` file is kept somewhere other than `/etc`. For example, if the `passwd` file is in `/var/yp/src`, type the following:

```
ypmaster# cd /var/yp
ypmaster# make DIR=/var/yp/src passwd
```

2.7.4.3 Make a Home Directory – On the new user's system, create a home directory for the new user. Use the same directory name that you specified in the YP master server's `/etc/passwd` file. For example, if you are setting up a new user `doe` in `/usr/staff`, use this sequence of commands:

```
ypclient# cd /usr/staff
ypclient# mkdir doe
ypclient# chown doe doe
ypclient# chgrp 10 doe
```

A common group identification number is 10. See `group(5yp)` in the *ULTRIX Reference Pages* for further information.

If the YP map for the password file has not yet been updated on the system's YP server, you get an error message when you attempt to run the `chown` command. The message has the following format:

```
unknown user id: username
```

In that case, you can use the new user's UID number (from the `/etc/passwd` file entry) instead of the login name to change the owner of the home directory. Here is the format of the command:

```
chown userid# username
```

See the *Guide to System Environment Setup* for further information about setting up new user accounts.

2.7.4.4 Set Up the New User's Environment – You can define new users' login environments in several ways. For example, you might give new users a copy of such files as `.login` and `.cshrc` if they use the C shell (`/bin/csh`), or just `.profile` if they use the Bourne shell (`/bin/sh`). Copies of the default environment files are stored in the directory `/usr/skel`. See the *Guide to System Environment Setup* and `csh(1)` and `sh(1)` in the ULTRIX Reference Pages for further information about setting up a new user's environment.

If the new user is a member of any groups at your site, add the user's login name to the `/etc/group` file as necessary. Be sure to make the changes to the `/etc/group` and `/etc/netgroup` files on the YP master server if you are running YP. See `group(5yp)` and `groups(1)` in the ULTRIX Reference Pages for more information about user groups.

2.7.4.5 Propagate the Updated YP Map – After you have modified the YP maps to include the new user, you need to propagate them to the other servers on the domain. See Section 2.6 for a description of how to propagate YP maps.

This chapter explains how to maintain and manage the Yellow Pages (YP) service. There are many ways you can use the YP service on your system, and some are more efficient than others for your particular YP domain. The information included in this chapter will help you understand the implications of setting up YP in various manners.

This chapter also discusses system security with YP and offers ways to increase this security.

3.1 System Security with YP

This section describes the various aspects of system security while YP is running. For further information about security, see `yppasswd(1yp)`, `hosts.equiv(5yp)`, `exports(5yp)`, `passwd(5yp)`, `group(5yp)`, `netgroup(5yp)`, and `yppasswdd(8yp)` in the ULTRIX Reference Pages.

3.1.1 Using the `-S` Option

To increase system security, you can use the `-S` option with the `yppbind` command and with the entry for `/etc/yppbind` in the `/etc/rc.local` file.

This option locks the domain and servers list. You can specify a single domain and up to four servers when you use this option. Once you use the `-S` option, the machine on which you entered the command will not switch domains and will use only the servers specified in the command. All specified servers must be listed in the `/etc/hosts` file.

The following example shows use of the `-S` option:

```
/etc/yppbind -S domainname, server1, server2, server3, server4
```

3.1.2 Global and Local YP Files

Of the YP maps, the following are originally in the `/etc` directory before YP is set up: `/etc/group`, `/etc/hosts`, `/etc/networks`, `/etc/passwd`, `/etc/protocols`, `/etc/rpc`, and `/etc/services`.

In addition, YP uses the `/etc/netgroup` file to create the netgroup YP map.

The YP maps are divided into local and global file types. The `/etc/passwd` and `/etc/group` files are local files. They are first checked for on the local system, and then any entries beginning with the YP special characters (`+`, `-`, `@`) are interpreted as appropriate.

The remaining YP maps (`hosts`, `netgroup`, `networks`, `protocols`, `rpc`, and `services`) are treated as global files only. The information in these maps is network-wide data and is accessed only from YP. However, when booting, each

system needs an entry in `/etc/hosts` for itself.

In summary, if YP is running, local files are consulted first; global files are only checked in the YP maps.

3.1.3 Local System Files with Pointers to YP Maps

The files `/etc/hosts.equiv` and `.rhosts` are not in the YP database. Each system has its own unique copy. However, you can place entries in your `/etc/hosts.equiv` file that refer to YP. Consider the following sample line:

```
+@engineering
```

Because this entry begins with `+@`, it includes all members of `engineering` as it is defined in the YP map `netgroup`. (The `@` refers to members of the `/etc/netgroup` file.) A line consisting only of `+` includes everyone in the `/etc/hosts.equiv` file.

Conversely, an entry starting with `-@` excludes everyone listed in that network group. For example, the following entry excludes everyone listed within the network group `sales`:

```
-@sales
```

To be able to log in to a remote system without having a password, you need to have an entry for your local system name in the `/etc/hosts.equiv` file and an entry for your login name in the `/etc/passwd` file (on the remote system). By having a plus (`+`) entry in `/etc/hosts.equiv`, you effectively bypass this check, and anyone with a login entry in the `/etc/passwd` file is allowed to log in to the system over the network without restriction.

The `/etc/passwd` and `/etc/group` files can also have plus and minus (`+, -`) entries. A line such as the following in the `/etc/passwd` file pulls an entry for `doe` from YP:

```
+doe:::John H. Doe:/usr2/doe:/bin/csh
```

The user and group identifications and the password are obtained from YP. The description field, home directory, and default shell are obtained from the plus (`+`) entry itself.

On the other hand, an `/etc/passwd` entry such as the following gets all of its information from YP:

```
+doe:
```

Notice the differences in the following two entries:

```
+doe::1189:10:John H. Doe:/usr2/doe:/bin/csh
```

```
doe::1189:10:John H. Doe:/usr2/doe:/bin/csh
```

In the first of the two entries, the password field is obtained from YP. In the second entry, user `doe` has no password. Also, if there is no entry for `doe` in YP, then the effect of the first entry is as if no entry for `doe` were present at all.

Note

Do not put the following entry in the `/etc/passwd` file:

```
+++0:0:0::
```

This entry would make every YP client on the network insecure. Each user whose password data is obtained from the YP service rather than the local `/etc/passwd` file would have root identification and permissions.

Finally, an entry such as the following excludes the user `doe` from being allowed to log in to the system:

```
-doe:
```

See Chapter 1 for further information about the plus and minus entries.

3.2 YP Map Access Policies

This section summarizes the policies used by the C-library routines when they access the following files on a system running YP:

`/etc/group`

Always consulted. If there are plus or minus (+,-) entries, the YP group map is consulted; otherwise, YP is unused.

`/etc/hosts`

Consulted only when booting (by the `ifconfig` command in the `/etc/rc.local` file). After that, the YP hosts map is used.

`/etc/hosts.equiv`

Always consulted, but not kept in the YP maps. If there are plus or minus (+,-) entries whose arguments are network groups, the YP netgroup map is consulted; otherwise, YP is unused.

`/etc/netgroup`

Never consulted. The `/etc/netgroup` file is used only for the construction of the YP netgroup map. All data is taken from YP.

`/etc/networks`

Never consulted. The data that was formerly read from this file now comes from the YP networks map.

`/etc/passwd`

Always consulted. If there are plus or minus (+,-) entries, the YP password map is consulted; otherwise, YP is unused.

`/etc/protocols`

Never consulted. The data that was formerly read from this file now comes from the YP protocols map.

`/.rhosts`

Always consulted, but not kept in the YP maps. If there are plus or minus (+,-) entries whose arguments are network groups, the YP netgroup map is consulted; otherwise, YP is unused.

`/etc/services`

Never consulted. The data that was formerly read from this file now comes from the YP services map.

`/etc/svc.conf`

Always consulted. This file specifies the order in which database lookup services are to be queried.

3.3 Special YP Password Change

When you change your password with the `passwd` command, you change the entry explicitly given in the local `/etc/passwd` file. If your password is not given explicitly, but is pulled in from YP with a plus (+) entry, then the `passwd` command prints this error message:

```
Not in passwd file.
```

If you are running YP on your system, the special account password entries are stored in `/etc/passwd`, but general user accounts are typically stored in `/var/yp/passwd`. Therefore, to change the superuser `root` password you must use the `passwd` command. To change a general user's password in YP, you must use the `yppasswd` command.

To enable the `yppasswd` command, the system administrator must start the `yppasswdd` daemon on the system serving as the master for the YP password file. The following entry in the `/etc/rc.local` file causes the `yppasswdd` daemon to start automatically each time the system is booted:

```
/usr/etc/rpc.yppasswdd /etc/passwd -m passwd DIR=/etc
```

See `yppasswdd(8yp)` in the ULTRIX Reference Pages for further information.

3.4 Using Netgroups

Netgroups are network-wide groups of systems and users defined in the `/etc/netgroup` file on the master YP server. These groups can be used for permission checking during remote mount, login, remote login, and remote shell processes.

The master YP server can use `/etc/netgroup` to generate three YP maps in the `/var/yp/domainname` directory: `netgroup`, `netgroup.byuser` and `netgroup.byhost`. The `netgroup` YP map contains the basic information in `/etc/netgroup`. The two other YP maps contain a more specific form of the information to speed the lookup process of network groups.

Some programs that consult the YP maps are `mountd`, `rlogin`, and `rsh`. The `mountd` program consults them for system classifications, if it encounters `netgroup` names in the `exports` file. The `rlogin` and `rsh` programs consult the `netgroup` map for both system and user classifications if they encounter `netgroup` names in the `hosts.equiv` or `.rhosts` file.

If you place your `/etc/netgroup` file in a source directory (such as `/var/yp/src`), when you execute the `make` command in the `/var/yp` directory, the `make` command will not find the `netgroup` file. To correct this, update the `netgroup` file in the source directory. Then copy it to `/etc/netgroup` before executing the `make` command. For more information, see `make(1)` in the ULTRIX Reference Pages.

For information on the `/etc/netgroup` file format, see `netgroup(5yp)` in the ULTRIX Reference Pages. See the *Guide to the Network File System* for information about the Network File System (NFS).

Here is a sample /etc/netgroup file for the domain market:

```
# Engineering: Everyone, but eric, has a system; he has no system.
# The system 'testing' is used by all of the hardware group.
#
engineering    hardware software
hardware      (mercury,alan,market) (venus,beth,market) (testing,-,market)
software      (earth,chris,market) (mars,deborah,market) (-,eric,market)
#
# Marketing: Time-sharing on star
#
marketing      (star,fran,market) (jupiter,greg,market)\
(jupiter,dan,market)
#
# Others
#
allusers      (-,market)
allhosts      (,-,market)
```

Based on this sample, the users would be classified into groups for the domain market as follows:

Group	Users
hardware	alan, beth
software	chris, deborah, eric
engineering	alan, beth, chris, deborah, eric
marketing	fran, greg, dan
allusers	every user in the passwd YP map
allhosts	no users

Here is how the systems would be classified:

Group	Hosts
hardware	mercury, venus, testing
software	earth, mars
engineering	mercury, venus, testing, earth, mars
marketing	star, jupiter
allusers	no hosts
allhosts	all hosts in the hosts YP map

This chapter describes the most common causes of YP malfunctions and provides some methods for solving the problems. The following topics are discussed:

- How to solve problems on a YP client
- How to solve problems on a YP server

The source of a YP problem usually lies in one of the following areas:

- There are no YP servers on the domain running the `ypserv` daemon
- The network or the YP server is overloaded
- The YP client has not set the domain name properly for the system
- The `ybind` process is not running
- The network is down

Before you can solve YP problems, you must be familiar with how YP operates and you should be familiar with the following YP commands and daemons:

`domainname`, `portmap`, `ybind`, `yocat`, `yomake`, `yomatch`, `yopasswdd`, `yopoll`, `yopush`, `yoserv`, `yosetup`, `yowhich`, and `yoxfr`. For additional information, see Chapter 2.

When solving YP problems, keep in mind that there are three main points of failure: the server, the client, or the network.

Note

The client and the server must be connected by a network for YP to be able to run and serve databases properly.

4.1 How to Solve Problems on a YP Client

This section provides a description of common errors on a YP client and offers solutions for these problems. The problems are:

- Commands hang
- The YP service is unavailable
- The `ybind` process exits
- The `ywhich` command is inconsistent

4.1.1 Solving the Problem of Commands That Hang

The most common problem on a YP client is for a command to hang and generate console messages of this form:

```
yp: server not responding for domain <domainname>. Still trying
```

This message indicates that `ypbind` on the local system is unable to communicate with `ypserv` in the specified domain.

Commands may hang if systems that run `ypserv` are taken off the network for any reason. This may also occur if the network or the YP server is so overloaded that `ypserv` cannot get a response back to the local system's `ypbind` within the timeout period.

Under these circumstances, the other YP clients on the network show the same or similar problems. The condition is temporary in most cases. The messages usually disappear when a YP server reboots and `ypserv` is running again, or when the load decreases on the YP servers or the Ethernet.

However, in the following circumstances, the situation does not improve:

- The YP client has not set, or has incorrectly set, the domain name on the system. Clients must use a domain name that the YP servers know. Use the `domainname` command to see the client domain name. Compare that with the domain name set on the YP servers. The domain name should be set in the `/etc/rc.local` file. For example, if the domain name is `market`, there should be an entry in the `/etc/rc.local` file similar to this:

```
/bin/domainname market
```

If `/etc/rc.local` fails to set, or incorrectly sets, the `domainname`, do the following:

1. Become superuser on the system in question.
2. Edit `/etc/rc.local` to fix the domain name line with a proper name. This assures the domain name will be correct every time the system boots.
3. Set domain name manually, so it is fixed immediately. For example, if the domain name is `market`, type the following command:

```
# domainname market
```

- If your domain name is correct, make sure your local network has at least one YP server. You can bind to a `ypserv` process only on your local network, not on another accessible network. There must be at least one YP server for your system's domain running on your local network. Two or more YP servers improve availability and response characteristics for the YP service.
- If your local network has a YP server, make sure it is running. Check other systems on your local network. If several client systems have problems simultaneously, suspect a server problem.

Find a client system that is operating normally and run the `ypwhich` command. If `ypwhich` does not return an answer, terminate it and go to a terminal on the YP server and type the following command:

```
# ps ax | grep yp
```

Look for `ypserv` and `ypbind` processes. Depending upon the results, take one of the following actions:

- If the `ps` command shows no `ypserv` process running, start one:

```
# /usr/etc/ypserv
```

- If the `ypserv` daemon was running but the `ypbind` daemon is not, start it by typing:

```
# /etc/ypbind [-S domainname, server1,...,server4]
```

Then execute `ypwhich` on the YP server. If `ypwhich` still returns no answer, `ypserv` has probably hung and should be restarted. Terminate the existing `ypserv`, `ypbind`, and `portmap` processes and start them again. For example, if the process IDs are 102, 121, and 156 type the following commands:

```
# kill -9 102 121 156
# /etc/portmap
# /usr/etc/ypserv
# /etc/ypbind
```

Find the process ID numbers by using the `ps` command.

4.1.2 Problems Caused by YP Service Unavailability

If other systems on the network appear to be running properly, but the YP service becomes unavailable on your system, many different symptoms can appear, such as:

- Some commands appear to operate correctly, while others terminate, printing an error message about the unavailability of YP.
- Some commands run inefficiently in a backup strategy particular to the program involved.
- Some commands or daemons exit with obscure messages or no message at all. Messages such as the following may appear (in this example, the domain name is `market`):

```
# ypcat passwd
ypcat: can't bind to YP server for domain <market>
Reason: can't communicate with ypbind.
```

```
# /var/yp/yppoll passwd.byname
RPC_TIMEDOUT
```

If symptoms such as these occur, type the following while in a directory containing files owned by many users, including users not in your system's `/etc/passwd` file (such as `/usr`):

```
# ls -l
```

If the `ls` command reports file owners who are not in your system's `/etc/passwd` file as numbers, rather than names, this is another indication that YP is not working.

These symptoms usually indicate that the `ypbind` process is not running. Run the `ps` command with the `a` and `x` options to check whether a `ypbind` process is running; if it is, kill `ypbind` and then restart it. If you do not find the `ypbind` process, type the following to start it:

```
# /etc/ypbind [-S domainname server1, ..., server4]
```

Another possibility is that the `/etc/svc.conf` file is incorrect. Be sure this file has an entry for YP.

4.1.3 Determining why the ypbind Process Exits

If the ypbind process exits almost immediately each time it is started, you should look for a problem in some other part of the system. Check for the presence of the portmap daemon by typing the following commands:

```
# ps ax | grep portmap
```

If you do not find it running, reboot the system.

If the portmap daemon does not stay up or acts unusual, look for more fundamental problems.

You may be able to talk to the portmap daemon on your system from another system on your network that is operating normally. From such a system, use the rpcinfo command. For example, if your system is named spice and the system that is operating normally is named sugar, type the following from sugar:

```
sugar# rpcinfo -p spice
```

If your portmap daemon is running properly, the output should look like:

program	vers	proto	port	
100003	2	udp	2049	nfs
100005	1	udp	1025	mountd
100004	2	udp	1033	ypserv
100004	2	tcp	1024	ypserv
100004	1	udp	1033	ypserv
100004	1	tcp	1024	ypserv
100007	2	tcp	1025	ypbind
100007	2	udp	1045	ypbind
100007	1	tcp	1025	ypbind
100007	1	udp	1045	ypbind

The port numbers on your system may be different from those shown.

If the ypbind processes are not there, ypbind has been unable to register its services. Reboot your system. If the ypbind processes are there and they change each time you try to restart /etc/ypbind, then reboot the system, even if the portmap daemon is running.

4.1.4 Why the ypwhich Command Might Be Inconsistent

If you use the ypwhich command several times at the same client system, the answer you get back may vary because the YP server can change. The binding of a YP client to a YP server can change over time on a busy network, or when the YP servers are busy.

Whenever possible, the system stabilizes at a point where all clients get acceptable response time from the YP servers. As long as your client system gets the YP service, it does not matter where the service comes from. A YP server often gets its own YP service from another YP server on the network.

If ypwhich returns "Domain *domainname* not bound", this is not necessarily a problem. Try entering ypcat passwd and then type ypwhich.

4.2 How to Solve Problems on a YP Server

This section provides a description of common errors on a YP server and offers solutions to these problems.

Because YP works by propagating maps among servers, you can sometimes find different versions of a map on different servers on the network. If transient, this version skew is normal. Otherwise, it is abnormal.

Most commonly, a normal update is prevented when a YP server or a network gateway system between YP servers is down during a map transfer attempt. When the YP servers and the network gateways between them are running, `ypxfr` should succeed.

If a particular slave server has update problems, log in to that server and run `ypxfr` interactively. If `ypxfr` fails, it prints an error message that will help you solve the problem. If `ypxfr` succeeds, but you believe that it is failing at times, create a log file to enable the logging of messages by typing the following commands:

```
# cd /var/yp
# touch ypxfr.log
```

This saves all output from `ypxfr`. The output looks much like what `ypxfr` creates when run interactively, but each line in the log file is timestamped.

You might see unexpected orderings in the timestamps. The timestamp tells you when `ypxfr` began its work. If copies of `ypxfr` ran simultaneously, but their work took different amounts of time, they might write their summary status line to the log files in a different order.

Any pattern of intermittent failure shows in the log file. After you have fixed the problem, turn off logging by removing the log file. If you forget to remove it, it grows without limit.

While still logged in to the problem YP slave server, inspect `/usr/lib/crontab` and the `ypxfr` shell scripts it invokes. Typing mistakes in these files can cause propagation problems. Failures to refer to a shell script within `crontab` or failures to refer to a map within any shell script can also cause propagation problems.

Make sure that the YP slave server is in the `ypservers` map within the domain. If it is not, it still works as a server, but `yppush` will not tell it when a new copy of a map exists.

If the problem is not obvious, you can work around it while you debug by using the `rcp` or `tftp` command to copy the current version from any stable YP server. You might not be able to do this as superuser, but you probably will be able to do it as daemon. For example, type the following to transfer a map called `buster`:

```
# chmod go+w /var/yp/market
# su daemon
$ rcp ypmaster:/var/yp/market/buster.\* /var/yp/market
$ <CTRL/D>
# chown root /var/yp/market/buster.*
# chmod go-w /var/yp/market
```

Notice that the asterisk (*) has been escaped with a backslash in the command line so that it will be expanded on the YP master server, instead of locally. In addition, notice that the map files should be owned by root, so you must change ownership of them after the transfer. It is easiest if you can do the `rcp` command as superuser.

Note

Because of architectural differences between VAX processors and other types of processors, you may not be able to copy files from one processor to another using the `rcp` command. The `ypxfr` command, however, does resolve the byte ordering differences found in a heterogeneous networking environment.

4.2.1 Updating a YP Database

If you change a database and then execute a `make` command, the database may not get updated. If this happens, remove the file `database.time` from the directories `/var/yp` and `/var/yp/domainname`.

For example, if the `netgroup` file of the domain `market` is changed and successfully updated, the `make` command should respond with:

```
netgroup updated
```

If the `make` command instead states that the `netgroup` is up to date, enter these commands:

```
# cd /var/yp
# rm netgroup.time
# cd cadnetwork
# rm netgroup.time
# cd ..
# make netgroup
```

4.2.2 Determining Why the `ypserv` Process Exits

If the `ypserv` process exits almost immediately and will not stay up even when repeatedly activated, the process of finding the problem is virtually identical to that described in Section 4.1.3. Check for the `portmap` daemon:

```
# ps ax | grep portmap
```

Reboot the server if you do not find it. However, if it is there, run the `rpcinfo` command:

```
# /usr/etc/rpcinfo -p

program vers proto  port
 100003   2   udp   2049  nfs
 100005   1   udp   1025  mountd
 100004   2   udp   1033  ypserv
 100004   2   tcp   1024  ypserv
 100004   1   udp   1033  ypserv
 100004   1   tcp   1024  ypserv
 100007   2   tcp   1025  ypbind
 100007   2   udp   1045  ypbind
 100007   1   tcp   1025  ypbind
 100007   1   udp   1045  ypbind
```

The port numbers on your system may be different from those shown. If `ypserv` processes are not there, `ypserv` has been unable to register its services. Reboot the system. If the `ypserv` processes are there but they change each time you try to restart `/usr/etc/ypserv`, reboot the system.

Appendix **A**

This appendix provides a quick reference for common YP commands.

makedbm (8yp)

Builds a dbm file, which is a valid YP map. Databases not built from `/var/yp/Makefile` can be built using `makedbm`. The `makedbm` command also disassembles a map so that you can see the key-value pairs. You can modify the disassembled form with standard tools (such as editors, `awk`, `grep`, and `cat`). The disassembled map is in the form required for input back into `makedbm`.

ypbind (1yp)

Activates the Yellow Pages binder daemon. The process is activated at system startup time from `/etc/rc.local`. It runs on all machines using the YP services, both servers and clients. A `ypbind` process associates the domain name with the Internet address of the YP server and the port on that host at which the `yplib` process is listening for server requests. The `-S` option allows the system administrator to lock `ypbind` to a particular domain and set of servers.

ypcat (1yp)

Displays the contents of a YP map. You can use `ypcat` when it does not matter which server's version you are seeing. If you need to see a particular server's map, log in to that server (using `rlogin`, or the `rsh` command) and use the `makedbm` command.

ypmake (8yp)

Rebuilds the YP database by using the `make` command. The `make` command uses the `/var/yp/Makefile` file to build any YP maps that are outdated, and then executes the `yppush` command to notify slave databases that a change has occurred. The `ypmake` command should run only on a YP master server machine.

ypmatch (1yp)

Prints the value for one or more specified keys in a YP map. You have no control over which server's version of the map you are seeing.

yppoll (8yp)

Determines the order number and master server for a specified map.

yppush (8yp)

Copies a new version of a YP map from the master YP server to the slave YP servers. This command is run on the master YP server by the `ypmake` command.

yplib (8yp)

Activates the YP database lookup server. The `yplib` process must run on each YP server.

ypsetup (8yp)

Sets up your system YP environment for the first time. The `ypsetup` command initializes the default maps for a master YP server, transfers copies of the master YP server maps for a slave YP server, and sets up the `/etc/rc.local` file for the master, slaves, and clients on the domain.

ypwhich (1yp)

Tells you which YP server a system is using at the moment.

ypxfr (8yp)

Moves a YP map from one YP server to another, using YP itself as the transport medium. It can be run interactively or periodically from `/etc/crontab`. In addition, `ypserv` uses `ypxfr` as its transfer agent when it is asked to transfer a map.

C

crontab file
editing, 2–9

D

dbm file
storing data, 1–2

domain
See YP domain

domain name
setting, 4–2

G

group file
editing for YP, 2–13
pulling entries from YP, 3–2
YP and, 1–4

H

home directory
creating, 2–19

hosts file
See also hosts.byaddr file
See also hosts.name file
editing for YP, 2–12
YP and, 1–3

hosts.byaddr file, 1–3

hosts.byname file, 1–3

hosts.equiv file
editing for YP, 2–11
referring to YP, 3–2
YP and, 3–2

M

makedbm command
defined, A–1

master server
See YP master server

N

netgroup file
format, 3–5
YP and, 1–4, 3–4
YP client and, 2–11

netgroup.byhost, 3–4

netgroup.byuser, 3–4

netgroups
defined, 3–4

networks file
YP and, 1–4
YP client and, 2–11

P

passwd file
See also passwd.byname file
See also passwd.byuid file
editing for YP, 2–12
editing master server's, 2–18
last entry, 2–12n
pulling entries from YP, 3–2
user ID and, 2–19n
YP and, 1–4
YP special characters and, 1–4

passwd.byname file, 1–4

passwd.byuid file, 1–4

password
changing in YP, 3-4
logging into remote system without, 3-2
YP and, 1-4

port mapper
starting, 2-4, 2-8, 2-10

protocols file
YP and, 1-4
YP client and, 2-11

R

rc.local file
entry order, 2-5n, 2-9n, 2-10n

rcp command, 4-6

.rhosts file, 3-2
restricting access, 2-11

rpc file
YP and, 1-4
YP client and, 2-11

RPC port mapper
See port mapper

S

security
and the -S option, 3-1

service order file
See svc.conf files

services file
YP and, 1-4
YP client and, 2-11

services order file
See /etc/svc.conf file
setting up with svcsetup, 2-5

slave server
See YP slave server

svc.conf file
if incorrect, 4-3
modifying, 2-5
recommended service order, 2-7
service order restrictions, 2-7

svcsetup command
defined, 2-5

system security
passwd file and, 3-3n
YP and, 3-3

U

user environment
setting up, 2-20

uucp utility
YP domain and, 2-12n

Y

Yellow Pages
See YP service

Yellow Pages database lookup service
See YP service

YP and, 3-1

YP client
adding users, 2-18
altering local files, 2-10 to 2-13
commands hang, 4-1 to 4-3
defined, 1-2
setting up manually, 2-9 to 2-10
troubleshooting, 4-1 to 4-4

YP database
changes to, 1-2
updating, 4-6

YP domain
defined, 1-1, 1-2
establishing, 2-3, 2-8, 2-10
getting name, 1-2
setting default, 2-8
setting default name, 2-5, 2-10

YP environment
modifying, 2-16 to 2-20

YP map
See also YP server map
access policies, 3-3
building default, 2-4
copying, 2-8
creating nonstandard, 2-13 to 2-14
editing nonstandard, 2-13
global file types, 3-1

YP map (cont.)

- library routines and, 1–3n
- local file pointers, 3–2
- local file types, 3–1
- modifying, 2–13 to 2–14
- propagating, 2–14
- propagating with make command, 2–14
- propagating with makedbm command, 2–15
- propagating with ypxfr command, 2–15
- updating, 2–19
- YP master server and, 2–13

YP master server

- acting as YP client, 2–4
- changing, 2–17 to 2–18
- defined, 1–2
- modifying database files, 2–4n
- required database files, 2–2
- setting up, 2–2

YP password server daemon

- starting, 2–5

YP server

See also YP master server

See also YP slave server

- acting as YP client, 2–1
- adding to domain, 2–16
- commands hang, 4–2
- setting up, 2–1 to 2–2
- troubleshooting, 4–4 to 4–6

YP server map

- creating, 2–5

YP service

- command reference list, A–1 to A–2
- default database files, 1–3 to 1–4
- description, 1–2 to 1–4
- introduction, 1–1 to 1–2
- managing, 3–1 to 3–5
- network and, 4–1n
- prerequisites for setup, 2–2
- setting up, 2–1 to 2–20
- setting up manually, 2–2 to 2–10
- troubleshooting, 4–1 to 4–6
- unavailable to client, 4–3

YP slave server

- defined, 1–2

YP slave server (cont.)

- removing from domain, 2–17
- setting up, 2–7 to 2–9
- updating YP maps, 2–9

ybind command

- defined, A–1
- and the `-S` option, 3–1

ybind daemon

- starting, 2–4

ybind process

- exits on client, 4–4

yccat command

- defined, A–1

yptest command

- defined, A–1

yptest command

- defined, A–1

yppoll command

- defined, A–1

yppush command

- defined, A–1

yptest command

- running, 2–4, 2–8

yptest daemon

- defined, A–1

yptest process

- exiting, 4–6

yptest command

- defined, A–2

yptest command

- defined, A–2

- inconsistent responses, 4–4

ypxfr command

- defined, A–2
- running, 2–8
- running from cron file, 2–15
- running manually, 2–16
- troubleshooting YP server and, 4–5

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	_____	Local Digital subsidiary or approved distributor
Internal*	_____	SSB Order Processing - WMO/E15 <i>or</i> Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to the Yellow Pages Service
AA-ME00B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What would you like to see more/less of? _____

What do you like best about this manual? _____

What do you like least about this manual? _____

Please list errors you have found in this manual:

Page	Description
------	-------------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

_____	_____
-------	-------

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear - Fold Here and Tape

digital™

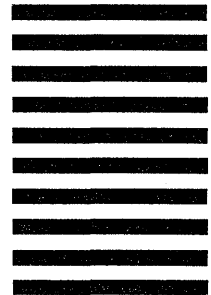


NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062-9987



Do Not Tear - Fold Here

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Guide to the Yellow Pages Service
AA-ME00B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What would you like to see more/less of? _____

What do you like best about this manual? _____

What do you like least about this manual? _____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

----- Do Not Tear - Fold Here and Tape -----

digital™



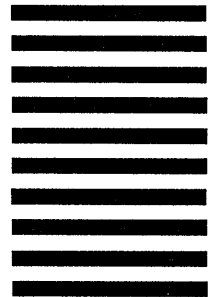
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3-2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062-9987



----- Do Not Tear - Fold Here -----

Cut
Along
Dotted
Line

