

**ULTRIX-32  
Guide to the  
Yellow Pages Service**

Order No. AA-ME00A-TE

---

ULTRIX-32 Operating System, Version 3.0

Digital Equipment Corporation


Copyright © 1987, 1988 Digital Equipment Corporation  
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:

DEC	Q-bus	VAX
DECnet	RT	VAXstation
DECUS	ULTRIX	VMS
MASSBUS	ULTRIX-11	VT
MicroVAX	ULTRIX-32	ULTRIX Worksystem Software
PDP	UNIBUS	

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

IBM is a registered trademark of International Business Machines Corporation.

MICOM is a registered trademark of Micom System, Inc.

This manual was written and produced by the ULTRIX Documentation Group in Nashua, New Hampshire.

## Contents

### About This Guide

Audience .....	vii
Organization .....	vii
Related Documents .....	viii
Conventions .....	viii

### 1 Introduction to the Yellow Pages Service

1.1 The YP Service .....	1-3
1.1.1 Naming Domains .....	1-3
1.1.2 Storing Data .....	1-3
1.1.3 Default YP Files .....	1-4
1.1.3.1 The /etc/hosts File .....	1-4
1.1.3.2 The /etc/passwd File .....	1-5
1.1.3.3 Other YP Files .....	1-5
1.2 YP Command Quick Reference .....	1-6

### 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 Automatically .....	2-3

2.3.1 Run the ypsetup Command .....	2-3
2.3.2 Modify the Default Data Base Files .....	2-4
2.4 Setting Up YP Manually .....	2-4
2.4.1 Setting Up a YP Master Server .....	2-4
2.4.1.1 Establish the YP Domain .....	2-5
2.4.1.2 Build the Default YP Maps .....	2-5
2.4.1.3 Start the YP Server Daemons .....	2-5
2.4.1.4 Modify the Default YP Files .....	2-6
2.4.1.5 Start the YP Password Server .....	2-6
2.4.1.6 Edit the /etc/rc.local File .....	2-7
2.4.1.7 Modify the /etc/svcorder File .....	2-7
2.4.1.8 Create the YP Servers Map .....	2-8
2.4.2 Setting Up a YP Slave Server .....	2-8
2.4.2.1 Establish the YP Domain .....	2-8
2.4.2.2 Obtain Copies of the YP Maps .....	2-8
2.4.2.3 Start the YP Server Daemons .....	2-9
2.4.2.4 Modify the Default YP Data Base Files .....	2-9
2.4.2.5 Edit the /etc/rc.local File .....	2-9
2.4.2.6 Modify the /etc/svcorder File .....	2-10
2.4.2.7 Edit the /usr/lib/crontab File .....	2-10
2.4.2.8 Add the New Slave Server to the Domain .....	2-10
2.4.3 Setting Up a YP Client .....	2-10
2.4.3.1 Establish the YP Domain .....	2-11
2.4.3.2 Start the YP Port Mapper Daemon .....	2-11
2.4.3.3 Modify the Default YP Data Base Files .....	2-11
2.4.3.4 Edit the /etc/rc.local File .....	2-11
2.4.3.5 Modify the /etc/svcorder File .....	2-11
2.5 Altering YP Client Local Files .....	2-12
2.5.1 The networks, protocols, rpc, services, and netgroup Files .....	2-12
2.5.2 The /etc/hosts.equiv File .....	2-12
2.5.3 The .rhosts File .....	2-13
2.5.4 The /etc/hosts File .....	2-13
2.5.5 The /etc/passwd File .....	2-13

2.5.6 The /etc/group File .....	2-14
2.6 Modifying and Creating YP Maps .....	2-14
2.7 Propagating YP Maps .....	2-16
2.7.1 Using make to Propagate YP Maps .....	2-17
2.7.2 Using makedbm to Propagate YP Maps .....	2-17
2.7.3 Using ypxfr to Propagate YP Maps .....	2-17
2.8 Modifying the YP Environment .....	2-19
2.8.1 Adding YP Servers to the Domain .....	2-19
2.8.2 Removing YP Slave Servers from the Domain .....	2-20
2.8.3 Changing a YP Map's Master Server .....	2-20
2.8.4 Adding Users to a YP Client .....	2-21
2.8.4.1 Edit the YP Master Server's /etc/passwd File .....	2-22
2.8.4.2 Update the YP Map .....	2-22
2.8.4.3 Make a Home Directory .....	2-22
2.8.4.4 Set Up the New User's Environment .....	2-23
2.8.4.5 Propagate the Updated YP Map .....	2-23

### **3 Maintaining and Managing the Yellow Pages Service**

3.1 System Security with YP .....	3-1
3.1.1 Global and Local YP Files .....	3-1
3.1.2 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 Netgroups: Network-Wide Groups of Systems and Users .....	3-4

### **4 Troubleshooting the Yellow Pages Service**

4.1 How to Solve Problems on a YP Client .....	4-1
4.1.1 Commands Hang .....	4-2

4.1.2 YP Service Unavailable .....	4-3
4.1.3 The ypbind Process Exits .....	4-4
4.1.4 The ypwhich Command Is Inconsistent .....	4-5
4.2 How to Solve Problems on a YP Server .....	4-5
4.2.1 YP Does Not Update Data Base .....	4-7
4.2.2 The ypserv Process Exits .....	4-7

## About This Guide

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 and an index. The chapters are:

#### Chapter 1: Introduction

This chapter introduces YP and provides the background information needed before you can set up and run YP on your system. Also, basic YP concepts are described.

#### Chapter 2: Setting Up the Yellow Pages Service

This chapter describes how to set up YP on your system automatically using the ypsetup command. It also describes how to set up YP manually. The manual description is included for those who want to understand how YP operates and which files are affected by YP.

#### Chapter 3: Maintaining and Managing the Yellow Pages Service

This chapter describes how to maintain and manage the YP service. System security and YP map access policies are discussed.

#### Chapter 4: Troubleshooting the Yellow Pages Service

This chapter 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.

## 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:

<b>special</b>	In text, each mention of a specific command, option, partition, pathname, directory, or file is presented in this type.
<b>command(x)</b>	In text, cross-references to the command documentation include the section number in the reference manual where the commands are documented. For example: See the <code>cat(1)</code> command. This indicates that you can find the material on the <code>cat</code> command in Section 1 of the reference pages.
<b>literal</b>	In syntax descriptions, this type indicates terms that are constant and must be typed just as they are presented.
<i>italics</i>	In syntax descriptions, this type indicates terms that are variable.
[ ]	In syntax descriptions, square brackets indicate terms that are optional.
...	In syntax descriptions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
<b>function</b>	In function definitions, the function itself is shown in this type. The function arguments are shown in italics.
<b>UPPERCASE</b>	The ULTRIX system differentiates between lowercase and uppercase characters. Enter uppercase characters only where specifically indicated by an example or a syntax line.
<b>example</b>	In examples, computer output text is printed in this type.
<b>example</b>	In examples, user input is printed in this bold type.
<b>%</b>	This is the default user prompt in multiuser mode.



*system#* This is the superuser prompt on the system indicated by the system name specified.

*#* This is the default superuser prompt.

*>>>* This is the console subsystem prompt.

*.* In examples, a vertical ellipsis indicates that not all of the lines of the example are shown.

*.*

**KEYNAME** In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal keyboard.

**CTRL/x** In examples, symbols like this indicate that you must hold down the CTRL key while you type the key that follows the slash. Use of this combination of keys may appear on your terminal screen as the letter preceded by the circumflex character. In some instances, it may not appear at all.



# Introduction to the Yellow Pages Service 1

This chapter introduces the network data base service called Yellow Pages (YP).

YP is a distributed data base lookup service that provides these services:

- It maintains a set of data bases. Programs can ask for the value associated with a particular key or any group of keys in a data base.
- It propagates updated data bases among the systems on the network (YP domain), ensuring consistency. The YP data bases are fully duplicated on several systems known as YP servers. Multiple servers give the YP service a high degree of availability and reliability because the query response is the same everywhere.

A YP domain is a directory in `/var/yp` containing a named set of YP maps. You can determine your YP domain by executing the `domainname` command. Note that YP domains are different from both Internet domains and sendmail domains.

A domain name is required for retrieving data from a YP data base. Each system on the network belongs to a default domain set in `/etc/rc.local` at boot time with the `domainname` command. If your YP domain is `market` and you want to find the Internet address of host `sugar`, you must ask YP for the value associated with the key `sugar` in the map `hosts.byname` within the YP domain `market`. For example:

```
% ypcat hosts | grep sugar
```

A YP server holds all the maps of a YP domain in a subdirectory of `/var/yp`, named after the domain. For example, maps for the `market` domain would be held in `/var/yp/market`. YP uses this information internally.

A YP map is a set of data base files residing in the directory `/var/yp/domainname`. 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 information about the dbm file format.

Servers provide resources and clients use them. There is not necessarily a one-to-one correspondence between servers, clients, and systems. In fact, a system that acts as a server can also act as a client. To illustrate, consider two different services: the Network File System (NFS) and YP.

NFS allows client systems to mount remote file systems and directories and to access files in place, provided a server system has exported the file system or directory. A server that exports file systems and directories may also mount remote file systems and directories exported by other systems, thus becoming a client. A given system can be both server and client, a client only, or a server only.

The YP server is a process rather than a system. A process can request information out of the YP data base, obviating the need to have such information on every system. All processes that make use of the YP service are YP clients.

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 data bases. YP makes the data base set available over the network. A YP client system runs YP processes and requests data from data bases on other systems. Two kinds of systems have data bases: 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.

#### **Note**

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

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 data base server that knows those details.

The following topics are discussed in this chapter:

- The YP service
- YP command quick reference

## 1.1 The YP Service

The YP service provides a way or method for the network manager to maintain consistency among selected system administrative files on all the systems in a YP domain. There are two parts of YP to consider: how it operates, and what system files from /etc are affected by YP.

The YP service maintains network-wide data bases, 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 data base 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 of those 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 following sections describe various aspects of how YP accomplishes its services.

### 1.1.1 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 a user 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 directory /var/yp/domainname. A system can contain data for several different domains.

### 1.1.2 Storing Data

YP maps store data in dbm files. For example, the YP map for /etc/hosts in the domain market is 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 in /var/yp to create YP map files. This Makefile in turn calls makedbm. See ypmake(8yp) in the ULTRIX Reference Pages for further information on rebuilding YP data bases.

### 1.1.3 Default YP Files

These are the default data base files that YP serves:

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

Library routines such as `getpwent`, `getgrent`, and `gethostent` use YP. 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.

**1.1.3.1 The /etc/hosts File** - The 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 `ypcat hosts` is translated into `ypcat hosts.byaddr`, because there is no file called `hosts` in YP. Type the following command for a list of expanded names:

```
# ypcat -x
```

**1.1.3.2 The /etc/passwd File** – The passwd file is similar to the hosts file. It exists as two separate files, passwd.byname and passwd.byuid. The ypcat program prints it, and ypmake 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. The at (@) character can be used 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 yppasswd command unless you need to modify an entry in the local /etc/passwd file. The yppasswd command has the same user interface as the passwd command, but works only if the yppasswd daemon is running on the YP master server.

#### Note

The passwd command does not change the password YP map. It only changes 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.

**1.1.3.3 Other YP Files** – Of the other files in the /etc directory, /etc/group is treated like /etc/passwd, in that getgrnt 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 affect in the /etc/networks, /etc/hosts, /etc/rpc, /etc/protocols, /etc/services, and /etc/netgroup files.

## 1.2 YP Command Quick Reference

Most of the information describing the structure of YP and the commands available for it is contained in the ULTRIX Reference Pages. For a quick reference, the related commands and abstracts of their functions are listed here.

### `makedbm(8yp)`

Describes a low-level tool for building a dbm file, which is a valid YP map. Data bases 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`.

### `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)`

Describes how to use `/var/yp/Makefile`, which builds several commonly changed components of the YP maps. These components are the maps built from several ASCII files normally found in the `/etc` directory: `group`, `hosts`, `netgroup`, `networks`, `passwd`, `protocols`, `rpc`, and `services`.

### `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)`

Asks any `ypserv` process for the information it holds internally about a single map.

### `yppush(8yp)`

Describes a tool to administer a running YP system. It is run on the master YP server. It requests each of the `ypserv` processes within a domain to transfer a particular map, waits for a summary response from the transfer agent, and prints the results for each server.

### `ypserv(8yp)`

Describes the processes that make up YP. These processes are `ypserv`, the YP map server daemon, and `ypbind`, the YP binder daemon. The `ypserv` process must run on each YP server. The



ypbind daemon must run on all systems that act as YP clients.

**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 or which YP server is the master of some named map.

**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.



## Setting Up and Modifying Yellow Pages 2

This chapter explains how to set up the Yellow Pages (YP) service on your system using the `ypsetup` command. It also describes how to set up YP manually and how to modify an existing YP environment. See Chapter 1 for a description of YP.

The topics discussed in this chapter are:

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

### 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`. In addition, the YP master server needs to 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 the 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. 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 will add 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 server?

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 Automatically**

To set up YP automatically, run the `ypsetup` command. Then, modify the YP data base files as outlined in Sections 2.3.1 and 2.3.2.

### **2.3.1 Run the `ypsetup` Command**

With the system in multiuser mode, run the `ypsetup` command:

```
# /etc/ypsetup
```

Once `ypsetup` is running, it prints an informational message.

Answer each of the `ypsetup` prompts. The `ypsetup` command asks if you want the YP daemons started automatically. If you say yes, then the YP service will run when `ypsetup` terminates. If you say no, then you will need to start the YP daemons by hand after `ypsetup` terminates, in order for YP to run. Either way the YP daemons will start automatically upon subsequent system reboots.

When `ypsetup` has made the required modifications, it prints this informational message:

```
***** YELLOW PAGES SETUP COMPLETE *****
```

The `ypsetup` command checks to see if the file `/etc/svcorder` exists. The `svcorder` file designates the order in which database lookup services such as YP and BIND are to be searched to resolve host names and addresses. This file is required for the YP service to run.

If the `svcorder` file does not exist, then `ypsetup` creates it with an entry for the YP service. If the `svcorder` file already exists but does not contain an entry for YP, `ypsetup` gives you this reminder:

```
Be sure to update the svcorder file to reflect the proper
order in which to access the Yellow Pages service for host
name/address resolution.
```

If you are running more than one service, you should consider the order in which the services are to be queried. You can edit the `/etc/svcorder` file to reflect this order after `ypsetup` completes.

See `svcorder(5)` in the ULTRIX Reference Pages for further information about this file.

### **2.3.2 Modify the Default Data Base Files**

If your system will be a YP client, edit the original versions of the default data base files to include the YP escape characters. This is described in Section 2.5. These files are: `/etc/group`, `/etc/hosts`, `/etc/netgroup`, `/etc/networks`, `/etc/passwd`, `/etc/protocols`, and `/etc/services`.

## **2.4 Setting Up YP Manually**

For an interactive setup with default answers, follow the automatic YP setup procedure (see Section 2.3).

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.

The following topics are discussed in this section:

- Setting up a YP master server
- Setting up a YP slave server
- Setting up a YP client

### **2.4.1 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 `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 about the `netgroup` file.

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/svcorder` file
8. Create the YP servers map

**2.4.1.1 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
```

**2.4.1.2 Build the Default YP Maps** – Build the default YP maps:

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

**2.4.1.3 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
```

**2.4.1.4 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 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 Makefile:

```
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, 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.yppasswdd /var/yp/src/passwd \  
-m passwd DIR=/var/yp/src
```

### Note

If you want to modify the YP master files at any time thereafter, you should modify the files in the holding area, 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
```

**2.4.1.5 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:



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

**2.4.1.6 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/ypbind 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 determine 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.

**2.4.1.7 Modify the /etc/svcorder File** – You need to modify the /etc/svcorder file. If it does not exist, then you should create it. This file designates the order and selection of database lookup services. It consists of the service names, one service per line, in the order they are to be queried. For example, if you are only running YP, the /etc/svcorder file should consist of this one line:

```
yp
```

If you are running both the YP service and the BIND service and you want the BIND service queried in the event that YP fails to find the information, here are the proper entries:

```
yp  
bind
```

Note that the order in which the service names appear in the file is significant. See svcorder(5) in the ULTRIX Reference Pages for further information.

**2.4.1.8 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:

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

## **2.4.2 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 data base files
5. Edit the `/etc/rc.local` file
6. Modify the `/etc/svcorder` file
7. Edit the `/usr/lib/crontab` file
8. Add the new slave server to the domain

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

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

**2.4.2.2 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:

```
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*.

**2.4.2.3 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
```

**2.4.2.4 Modify the Default YP Data Base Files** – If your system will act as a client, in addition to being a YP slave server, edit the default data base files as described in Section 2.5. Then, start the ypbind daemon:

```
ypslave# /etc/ypbind
```

**2.4.2.5 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 entry for the */etc/portmap* daemon is:

```
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; 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 NFS, be sure the YP entries precede the NFS entries in the /etc/rc.local file.

**2.4.2.6 Modify the /etc/svcorder File** - You need to modify the /etc/svcorder file. See Section 2.4.1.7 for information about the /etc/svcorder file.

**2.4.2.7 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.

**2.4.2.8 Add the New Slave Server to the Domain** - To add the new slave server to the domain, follow the directions in Section 2.8.1.

### **2.4.3 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 data base files
4. Edit the /etc/rc.local file
5. Modify the /etc/svcorder file

**2.4.3.1 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:

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

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

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

**2.4.3.3 Modify the Default YP Data Base Files** – Edit the default data base files as described in Section 2.5. Then, start the `ypbind` daemon:

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

**2.4.3.4 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; 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 NFS, be sure the YP entries precede the NFS entries in the `/etc/rc.local` file.

**2.4.3.5 Modify the /etc/svcorder File** – You need to modify the `/etc/svcorder` file. See Section 2.4.1.7 for information about the `/etc/svcorder` file.

## 2.5 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 six sections discuss how to treat each of these files.

### 2.5.1 The `networks`, `protocols`, `rpc`, `services`, and `netgroup` Files

The `/etc/networks`, `/etc/protocols`, `/etc/rpc`, `/etc/services`, and `/etc/netgroup` files are not needed on any YP client. 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
```

### 2.5.2 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 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, and YP is not used.

### 2.5.3 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, since the `.rhosts` file controls remote root access to the local system, you should restrict access to it. You should make the list of trusted hosts explicit or use the network group names.

### 2.5.4 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 `ybind` process is up, the `/etc/hosts` file is not accessed. An example of the hosts file for YP client orville is:

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

### 2.5.5 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/WWVZnIFgM:0:1:Bossman with a C shell:/:/bin/csh
operator::0:28:Operator:/opr/opser
daemon:*:1:1:Mr. Background:/:
sys:xzuEOVlLjYpJM: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.

### 2.5.6 The `/etc/group` File

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

```
+:
```

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

## 2.6 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 data bases 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 2.4.1, edit the directory. For example, edit `/var/yp/src/passwd`. Otherwise, edit the local file `/etc/passwd`.
2. Type:

```
# 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 nonstandard YP maps that are specific to the applications of a particular vendor or site, YP maps that you expect to change rarely, or YP 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 `al`, `bl`, `cl`, 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:

```
# 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
a| ar
b| br
c| 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:

```
# /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 2.7. To get an initial copy of the map, you can run `ypxfr` by hand 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 2.7 for a description of how to propagate YP maps.

## 2.7 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. The YP maps can be propagated from

following sections.

### 2.7.1 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. For example, to propagate the `hosts` file, type:

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

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

### 2.7.2 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. For example, to propagate a nondefault YP map called `sales.asc` that resides in `/var/yp/src` on the domain `market`, type:

```
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`, 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.

### 2.7.3 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 using the `ypxfr` command in two ways:

- 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

can set up the `/usr/lib/crontab` file entries to run `ypxfr` periodically at a rate appropriate for any map in your YP data base. 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: `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 easily modify or replace these shell scripts.

These shell scripts should be run on each YP slave server in the domain. You should alter the exact time of execution from one server to another to prevent the master from overloading. If you want the map transferred from some particular server other than the master, you can specify this using `ypxfr`'s `-h` option 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.

- **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:

```
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.

## 2.8 Modifying the YP Environment

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

- 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

### 2.8.1 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 sequence for adding a server named `osprey` to domain `market` is:

```
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. However, you cannot simply 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`, the commands are:

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

The new YP slave server's data bases should be set up 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 Section 2.3 for the automatic procedure and Section 2.4.2 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 2.7 for a description of how to propagate YP maps.

### 2.8.2 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 sequence for removing a server named `osprey` from domain `market` is:

```
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.

### 2.8.3 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 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 recreate 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:

```
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.

#### **2.8.4 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

**2.8.4.1 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/doe:/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.

**2.8.4.2 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:

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

**2.8.4.3 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 format of the message is:

```
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.

**2.8.4.4 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 his or her 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.

**2.8.4.5 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 2.7 for a description of how to propagate YP maps.



## Maintaining and Managing the Yellow Pages Service 3

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 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.

### 3.1.2 Local System Files with Pointers to YP Maps

The files `/etc/hosts.equiv` and `/rhosts` are not in the YP data base. 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:::
```

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

.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/svcorder

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 Netgroups: Network-Wide Groups of Systems and Users

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

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), then 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**

hardware  
software  
engineering  
marketing  
allusers  
allhosts

## **Hosts**

mercury, venus, testing  
earth, mars  
mercury, venus, testing, earth, mars  
star, jupiter  
no hosts  
all hosts in the hosts YP map



## Troubleshooting the Yellow Pages Service 4

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

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

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`, `ypmake`, `ypmatch`, `yppasswdd`, `yppoll`, `yppush`, `ypserv`, `ypsetup`, `ypwhich`, and `ypxfr`. 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 data bases 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
- YP service unavailable
- The `ybind` process exits

- The ypwhich command is inconsistent

#### 4.1.1 Commands 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, you should 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:

```
# 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

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:

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

Look for `ypserv` and `ypbind` processes. Depending upon the results:

- 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
```

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` and `ypbind` processes and start them again. For example, if the process IDs are 102 and 121, type:

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

The process ID numbers are found by using the `ps` command.

#### 4.1.2 YP Service Unavailable

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. You can run the ps command with the a and x options to check for one. If you do not find the ypbind process, type the following to start it:

```
# /etc/ypbind
```

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

#### **4.1.3 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:

```
# 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 The ypwhich Command Is 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.

## 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:

```
# 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 non-DIGITAL 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 YP Does Not Update Data Base

If you change a data base and then execute a make command, the data base may not get updated. If this happens, remove the file *data base.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 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 6.5.1.3. 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.





# Index

## Special Characters

**.rhosts file**  
YP and, 3-2

## C

**client**  
server and, 1-2  
**crontab file**  
editing, 2-10

## D

**dbm file**  
storing data, 1-3  
**domain**  
*See* YP domain  
**domain name**  
setting, 4-2

## G

**group file**  
editing for YP, 2-14  
pulling entries from YP, 3-2  
YP and, 1-5

## H

**home directory**  
creating  
**hosts file**  
*See also* hosts.byaddr file  
*See also* hosts.name file  
editing for YP, 2-13  
YP and, 1-4  
**hosts.byaddr file**, 1-4  
**hosts.byname file**, 1-4  
**hosts.equiv file**  
editing for YP, 2-12  
referring to YP, 3-2  
YP and, 3-2

## M

**makedbm command**  
defined, 1-6  
**master server**  
*See* YP master server

## N

**netgroup file**  
format, 3-5e  
YP and, 1-5, 3-4  
YP client and, 2-12

**netgroup.byhost**, 3-4

**netgroup.byuser**, 3-4

**netgroups**

defined, 3-4

**networks file**

YP and, 1-5

YP client and, 2-12

## P

**passwd file**

*See also* passwd.byname file

*See also* passwd.byuid file

editing for YP, 2-13

editing master server's, 2-22

last entry, 2-14n

pulling entries from YP, 3-2

user ID and, 2-22n

YP and, 1-5

YP special characters and, 1-5

**passwd.byname file**, 1-5

**passwd.byuid file**, 1-5

**password**

changing in YP, 3-4, 3-4

logging into remote system without,  
3-2

YP and, 1-5

**port mapper**

starting, 2-5, 2-9, 2-11

**protocols file**

YP and, 1-5

YP client and, 2-12

## R

**rc.local file**

entry order, 2-7n, 2-10n, 2-11n

**rpc command**

non-DIGITAL processors and, 4-6n

**.rhosts file**

restricting access, 2-13

**rpc file**

YP and, 1-5

YP client and, 2-12

**RPC port mapper**

*See* port mapper

## S

**server**

client and, 1-2

**services file**

YP and, 1-5

YP client and, 2-12

**slave server**

*See* YP slave server

**svccorder file**

defined, 2-3

if incorrect, 4-4

if missing, 2-3

modifying, 2-7

YP and, 2-3

**system security**

passwd file and, 3-3n

YP and, 3-1 to 3-3

## U

**user environment**

setting up, 2-23

**uucp utility**

YP domain and, 2-14n

## Y

**Yellow Pages data base lookup service**

*See* YP service

**YP client**

adding users, 2-21

### **YP client (cont.)**

- altering local files, 2-12 to 2-14
- commands hang, 4-2 to 4-3
- defined, 1-2
- setting up manually, 2-10 to 2-11
- troubleshooting, 4-1 to 4-5

### **YP data base**

- changes to, 1-2
- updating, 4-7

### **YP domain**

- defined, 1-1, 1-3
- establishing, 2-5, 2-8, 2-11
- getting name, 1-3
- setting default, 2-9
- setting default name, 2-7, 2-11

### **YP environment**

- modifying, 2-19 to 2-23

### **YP map**

- See also* YP server map
  - access policies, 3-3
  - building default, 2-5
  - copying, 2-8
  - creating nonstandard, 2-15 to 2-16
  - defined, 1-1
  - editing nonstandard, 2-15
  - global file types, 3-1
  - library routines and, 1-4n
  - local file pointers, 3-2
  - local file types, 3-1
  - modifying, 2-14 to 2-16
  - propagating, 2-16
  - propagating with make command, 2-17
  - propagating with makedbm command, 2-17
  - propagating with ypxfr command, 2-17
  - updating, 2-22
  - YP master server and, 2-14
- ### **YP master server**
- acting as YP client, 2-6

### **YP master server (cont.)**

- changing, 2-20 to 2-21
- defined, 1-2
- modifying data base files, 2-6n
- required data base files, 2-4
- setting up, 2-4 to 2-8

### **YP password server daemon**

- starting, 2-6

### **YP server**

- See also* YP master server
- See also* YP slave server
- acting as YP client, 2-1
- adding to domain, 2-19
- commands hang, 4-3
- defined, 1-2
- setting up, 2-1 to 2-2
- troubleshooting, 4-5 to 4-7

### **YP server map**

- creating, 2-8

### **YP service**

- command reference list, 1-6 to 1-7
- default data base files, 1-4 to 1-5
- description, 1-3 to 1-5
- introduction, 1-1 to 1-2
- managing, 3-1 to 3-6
- network and, 4-1n
- NFS and, 1-2
- prerequisites for setup, 2-2
- setting up, 2-1 to 2-23
- setting up automatically, 2-3
- setting up manually, 2-4 to 2-11
- troubleshooting, 4-1 to 4-7
- unavailable to client, 4-3 to 4-4

### **YP slave server**

- defined, 1-2
- removing from domain, 2-20
- setting up, 2-8 to 2-10
- updating YP maps, 2-10

### **yplib daemon**

- starting, 2-6

**ypbind process**

exits on client, 4-4

**ypcat command**

defined, 1-6

**ypmake command**

defined, 1-6

**ypmatch command**

defined, 1-6

**yppoll command**

defined, 1-6

**yppush command**

defined, 1-6

**ypserv command**

running, 2-5, 2-9

**ypserv daemon**

defined, 1-6

**ypserv process**

exiting, 4-7

**ypsetup command**

defined, 1-7

running, 2-3

**ypwhich command**

defined, 1-7

inconsistent responses, 4-5

**ypxfr command**

defined, 1-7

running, 2-8

running from cron file, 2-17

running manually

troubleshooting YP server and, 4-5

## HOW TO ORDER ADDITIONAL DOCUMENTATION

### DIRECT TELEPHONE ORDERS

In Continental USA  
and New Hampshire,  
Alaska or Hawaii  
call **800-DIGITAL**

In Canada  
call **800-267-6215**

### DIRECT MAIL ORDERS (U.S. and Puerto Rico\*)

DIGITAL EQUIPMENT CORPORATION  
P.O. Box CS2008  
Nashua, New Hampshire 03061

### DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.  
100 Herzberg Road  
Kanata, Ontario K2K 2A6  
Attn: Direct Order Desk

### INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION  
PSG Business Manager  
c/o Digital's local subsidiary  
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminister, Massachusetts 01473

\*Any prepaid order from Puerto Rico must be placed  
with the Local Digital Subsidiary:  
809-754-7575



## Reader's Comments

**Note:** This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Did you find errors in this manual? If so, specify the error and the page number.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) \_\_\_\_\_

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_

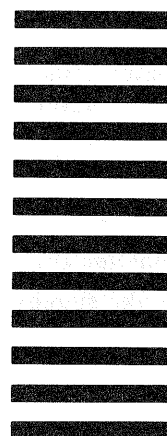
Street \_\_\_\_\_

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

**digital**



No Postage  
Necessary  
if Mailed in the  
United States



**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation  
Documentation Manager  
ULTRIX Documentation Group  
ZKO3-3/X18  
Spit Brook Road  
Nashua, N.H.  
03063

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

Cut Along Dotted Line



## Reader's Comments

**Note:** This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Did you find errors in this manual? If so, specify the error and the page number.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) \_\_\_\_\_

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_

Street \_\_\_\_\_

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

**digital**



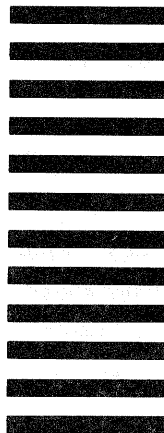
No Postage  
Necessary  
if Mailed in the  
United States

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation  
Documentation Manager  
ULTRIX Documentation Group  
ZKO3-3/X18  
Spit Brook Road  
Nashua, N.H.  
03063



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

Cut Along Dotted Line











