Network Services Guide

Network Services Guide

1. Introduction

This document is intended for users who have a general interest in network services. Although this is not a manual for system administrators, the material is heavily slanted in that direction.

Sun provides several network services, such as Network Disk (ND), and the Network File System (NFS), discussed in the document *Sun's Network File System*. The yellow pages are another network service offered for the first time on the 2.0 release. They permit password information and host addresses for an entire network to be held in a single database. This greatly eases the task of system and network administration. Sun will provide more network services in the future.

2. What Are The Yellow Pages?

The yellow pages (YP) constitute a distributed network lookup service:

- YP is a lookup service: it maintains a set of databases for querying. Programs can ask for the value associated with a particular key, or all the keys, in a database.
- YP is a network service: programs need not know the location of data, or how it is stored. Instead, they use a network protocol to communicate with a database server that knows those details.
- YP is distributed: databases are fully replicated on several machines, known as YP servers. Servers propagate updated databases among themselves, ensuring consistency. At steady state, it doesn't matter which server answers a request; the answer is the same everywhere.

2.1. The YP Map

The yellow pages serve information stored in YP *maps*. Each map contains a set of keys and associated values. For example, the *hosts* map contains (as keys) all host names on a network, and (as values) the corresponding Internet addresses. Each YP map has a *mapname*, used by programs to access data in the map. Programs must know the format of the data in the map. Currently, most maps are derived from ASCII files formerly found in */etc*: *passwd*, *group*, *hosts*, *networks*, and others. The format of data in the YP map is in most cases identical to the format of the ASCII file. Maps are implemented by *dbm*(3) files located in subdirectories of */etc/yp* on YP server machines.

2.2. The YP Domain

A YP *domain* is a named set of YP maps. You can determine your YP domain by executing the *domainname*(1) command. Note that YP domains are different from both Internet domains and *sendmail* domains. A YP domain is simply a directory in */etc/yp* containing a set of maps.

A domain name is required for retrieving data from a YP database. For instance, if your YP domain is *sun* and you want to find the Internet address of host *dbserver*, you must ask YP for the value associated with the key *dbserver* in the map *hosts.byname* within the YP domain *sun*. Each machine on the network belongs to a default domain, set in */etc/rc.local* at boot time with the *domainname*(8) command.

A YP server holds all the maps of a YP domain in a subdirectory of */etc/yp*, named after the domain. In the example above, maps for the *sun* domain would be held in */etc/yp/sun*. Every YP server must have the directory */etc/yp/yp_private*, which contains information about servers, domains, and maps. This information is used internally by the YP. For completeness, the YP server machine is its own client.

2.3. Servers And Clients

Servers provide resources, while clients consume them. A server or a client is not necessarily the same thing as a machine. To illustrate, let's consider two different services: the NFS (network file system) and the YP.

NFS The NFS allows client machines to mount remote filesystems and access files in place, provided a server machine has exported the filesystem. However, a server that exports filesystems may also mount remote filesystems exported by other machines, thus becoming a client. So a given machine may be both server and client, or client only, or server only. Furthermore, NFS servers and clients need not coincide with ND servers and clients.

YP The YP server, by contrast, is a process rather than a machine, running on a machine that may be neither ND server nor NFS server. All processes that make use of YP services are YP clients. Sometimes clients are served by YP servers on the same machine, but other times by YP servers running on another machine. To further muddy the waters, processes on master YP server machines (discussed below) don't use YP services at all, and aren't YP clients. But processes using YP services on slave YP servers are YP clients.

2.4. Masters and Slaves

YP servers are either master or slave. For any map, one YP server is designated the master, and all changes to the map should be made on that machine. The changes propagate from master to slaves. A newly built map is timestamped internally when *makedbm* creates it. If you build a YP map on a slave server, you will break the YP update algorithm (temporarily), and you will have to get all versions in synch manually. Moral: after you decide which server is the master, do all database updates and builds there, not on slaves.

It is possible for different maps to have different servers as master. Therefore, a given server may be a master with regard to one map, and a slave with regard to another. This can get confusing quickly. It is suggested that a single server be master for all the maps created by *ypinit* in a single domain. This document assumes the simple case, in which one server is the master for all maps in the database.

3. Overview of the Yellow Pages

In releases before 2.0, each machine on the network had its own copy of */etc/hosts*, a file containing the Internet address of each machine on the network. Every time a machine was added to the network, each */etc/hosts* file had to be updated.

The YP is a network service containing network-wide databases such as */etc/hosts*. There are servers spread throughout the network containing copies of the databases. When an arbitrary machine on the network wants to look up something in */etc/hosts*, it makes an RPC call to one of the servers to get the information. One server is the master — the only one whose database may be modified. The other servers are slaves, and they are periodically updated so that their information is in synch with that of the master.

The YP can serve up any number of databases. Normally that will include files that previously lived in */etc*, such as */etc/hosts* and */etc/networks*. However, users can add their own databases to the YP.

The YP itself simply serves up information, and has no idea what it means. Thus there are two parts of YP we need to consider: how it operates, and what files formerly in */etc* now live in the YP. This has serious ramifications for users.

3.1. The YP Network Service

3.1.1. Naming

Imagine a company with two different networks, each of which has its own separate list of hosts and passwords. Within each network, user names, numerical user IDs, and host names are unique. However, there is duplication between the two networks. If these two networks are ever connected, chaos could result. The host name, returned by the *hostname*(1) command and the *gethostname*() system call, may no longer uniquely identify a machine. Thus a new command and system call, *domainname*(1) and *getdomainname(2)* have been added. In the example above, each of the two networks could be given a different domain name. However, it is always simpler to use a single domain whenever possible.

The relevance of domains to YP is that data is stored in */etc/yp/*domainname. In particular, a machine can contain data for several different domains.

3.1.2. Data Storage

The data is stored in *dbm*(3) format. Thus the database *hosts.byname* for the domain *sun* is stored as */etc/yp/sun/hosts.byname.pag* and */etc/yp/sun/hosts.byname.dir*. The command *makedbm*(8) takes an ASCII file such as */etc/hosts* and converts it into a *dbm* file suitable for use by the YP. However, system administrators normally use the makefile in */etc/yp* to create new *dbm* files (read on for details). This makefile in turn calls *makedbm*.

3.1.3. Servers

To become a server, a machine must contain the YP databases, and must also be running the YP daemon *ypserv*. The *ypinit* (8) command invokes this daemon automatically. It also takes a flag saying whether you are creating a master or a slave. When updating the master copy of a database, you can force the change to be propagated to all the slaves with the *yppush* (8) command. This pushes the information out to all the slaves. Conversely, from a slave, the *yppull* (8) command gets the latest information from the master. The makefile in */etc/yp* first executes *makedbm* to make a new database, and then calls *yppush* to propagate the change throughout the network.

3.1.4. Clients

Remember that a client machine (which is not a server) does not contain any data itself, but rather makes an RPC call to a YP server each time it needs information from a YP database. The *ypbind* (8) daemon caches the name of a server. When a client boots, *ypbind* broadcasts asking for the name of the YP server. Similarly, if the cached server crashes, *ypbind* broadcasts asking for the name of a new server. The *ypwhich* (1) command gives the name of the server that *ypbind* currently points at.

Since client machines no longer have entire copies of files in the YP, a new command *ypcat* (1) has been provided. The command `ypcat hosts` is equivalent to `cat /etc/hosts` in a pre 2.0 system; as you might guess, `ypcat passwd` is equivalent to `cat /etc/passwd`. To look for someone's password entry, searching through the password file no longer suffices; you have to issue the following command

```
% ypcat passwd | grep userid
```

where you replace `userid` with the login name you're searching for.

3.2. Default YP Files

By default, Sun workstations have six files from */etc* in the YP: */etc/passwd*, */etc/groups*, */etc/networks*, */etc/hosts*, */etc/services*, and */etc/protocols*. In addition, there is a new file *netgroup*, which many sites ought to create and put in the YP database.

Library routines such as *,getpwent* (3) *,getgrent* (3) and *gethostent* (3) have been rewritten to take advantage of the YP. Thus, C programs that call these library routines will have to be relinked in order to function correctly.

3.2.1. Hosts

The hosts file is stored as two different files in the YP. The first, *hosts.byname*, is indexed by hostname. The second, *hosts.byaddr*, is indexed by Internet address. Remember that this actually expands into four files, with suffixes *.pag*, and *.dir*. When a user program calls the library routine *gethostbyname* (3), a single RPC call to a server retrieves the entry from the *hosts.byname* file. Similarly, *gethostbyaddr* (3) retrieves the entry from the *hosts.byaddr* file. Of course if the YP is not running (which is caused by commenting *ypbind* out of the */etc/rc* file), then *gethostbyname* will read the */etc/hosts* files, just as it always has.

Although the *ypcat* command is a general YP database print program, it knows about the standard files in the YP. Thus `ypcat hosts` is translated into `ypcat hosts.byaddr`, since there is no file called *hosts* in the YP.

Normally, the hosts file for the YP will be the same as the */etc/hosts* file on the machine serving as a YP master. In this case, the makefile in */etc/yp* will check to see if */etc/hosts* is newer than the *dbm* file. If it is, it will use a simple *sed* script to recreate *hosts.byname* and *hosts.byaddr*, run them through *makedbm* (8) and then call *yppush* (8). See *ypmake* (8) for details.

3.2.2. Passwd

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. However, if *getpwent* (3) always went directly to the YP as does *gethostent* (3), then everyone would be forced to have an identical password file! Consequently, *getpwent* reads the local */etc/passwd* file, just as it always did. But now it interprets ‘‘+’’ entries in the password file to mean, interpolate entries from the YP database. If you wrote a simple program using *getpwent* to print out all the entries from your password file, it would print out a virtual password file: rather than printing out + signs, it would print out whatever entries the local password file included from the YP database. The difference between */etc/hosts* and */etc/passwd* is discussed in more detail in the section ‘‘How Security is Changed with the Yellow Pages,’’ part of the *System Administrator’s Manual*.

3.2.3. Others

Of the other four files in */etc*, */etc/group* is treated like */etc/passwd*, in that *getgrent()* will only consult the YP if explicitly told to do so by the */etc/group* file. The files */etc/networks*, */etc/protocols*, */etc/services*, and */etc/networks* are treated like */etc/hosts*: for these files, the library routines go directly to the YP, without consulting the local files.

3.2.4. Changing your passwd

To change data in the YP, you must log onto the master machine, and edit databases there; *ypwhich* (1) tells where the master server is. However, since changing a password is so commonly done, the *yppasswd* (1) command has been provided to change your YP password. It has the same user interface as the *passwd* (1) command. This command will only work if the *yppasswdd* (8c) server has been started up on the YP master server machine.