Working With Oracle® Solaris 11.2
Directory and Naming Services: DNS and NIS
Ce logiciel et la documentation qui l’accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d’utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d’interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu’elles soient exemptes d’erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l’accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l’utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s’applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d’applications de gestion des informations. Ce logiciel ou matériel n’est pas conçu ni n’est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d’applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l’utilisation de ce logiciel ou matériel pour ce type d’applications.

Oracle et Java sont des marques déposées d’Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d’autres propriétaires qu’Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d’Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d’Advanced Micro Devices. UNIX est une marque déposée d’The Open Group.

Ce logiciel ou matériel et la documentation qui l’accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l’accès à des contenus, produits ou services tiers, ou à leur utilisation.
Contents

Using This Documentation ................................................................. 7

1 About Naming and Directory Services ........................................... 9
   What Is a Naming Service? ............................................................ 9
   Oracle Solaris Naming Services ..................................................... 15
      Naming Services and the Service Management Facility ............... 15
      Description of the DNS Naming Service .................................. 15
      Description of Multicast DNS and Service Discovery ............... 16
      Description of the /etc Files Naming Service ......................... 16
      Description of the NIS Naming Service .................................. 17
      Description of the LDAP Naming Services ............................... 17
   Naming Services: A Quick Comparison ........................................ 18
   IPv6 Extensions to Oracle Solaris Name Services ......................... 18
   DNS Extensions for IPv6 ............................................................ 19

2 About the Name Service Switch ...................................................... 21
   Overview of the Name Service Switch ......................................... 21
   Databases and Sources for the Name Service Switch .................... 22
      keyserv and publickey Entries in the Name Service Switch ........ 26
   Configuring the Name Service Switch ......................................... 27
      ▼ How to Change the Source for a Database ............................... 27
      ▼ How to Configure a Search Criterion for a Database ............... 28
      ▼ How to Change the Source for All Naming Databases ............... 29
      ▼ How to Use a Legacy nsswitch.conf File ................................. 29
   Name Service Switch and Password Information .......................... 30

3 Managing Domain Name System .................................................... 31
   DNS Overview ............................................................................. 31
   Multicast DNS ........................................................................... 31
Contents

Multicast DNS Service Discovery ................................................................. 32
Related Materials About DNS .................................................................. 32
DNS and the Service Management Facility ................................................... 32
Administering DNS (Tasks) ........................................................................ 34
 ▼ How to Install the DNS Package ........................................................... 34
 ▼ How to Configure a DNS Server ........................................................... 34
 ▼ How to Create an rndc.conf File ............................................................ 35
 ▼ How to Configure DNS Server Options ............................................... 36
 ▼ How to Run the DNS Service as an Alternative User ............................. 37
 ▼ How to Enable a DNS Client ................................................................. 38
 ▼ How to Troubleshoot DNS Server Startup Issues ................................ 38
 ▼ How to Verify the DNS Configuration .................................................. 39
Administering Multicast DNS ................................................................. 40
 ▼ How to Enable mDNS and DNS Service Discovery ............................... 40
Advertising Resources for DNS .............................................................. 41
DNS Reference .......................................................................................... 42
 DNS Files .................................................................................................. 42
 DNS Commands and Daemons ................................................................. 42
Compilation Flags Used When BIND Was Built ..................................... 43

4 Setting Up Oracle Solaris Active Directory Clients ................................. 45
 Overview of the nss_ad Naming Service Module .................................... 45
 ▼ How to Configure the nss_ad Module ................................................... 46
Password Updates ..................................................................................... 48
How the nss_ad Naming Service Module Retrieves Data From AD ............ 48
 Retrieving passwd Information ............................................................... 48
 Retrieving shadow Information ............................................................... 49
 Retrieving group Information ................................................................. 49

5 About the Network Information Service ................................................. 51
 NIS Introduction ........................................................................................ 51
 NIS Architecture ..................................................................................... 52
 NIS Machine Types .................................................................................. 53
 NIS Servers .............................................................................................. 53
 NIS Clients ............................................................................................... 53
 NIS Elements ........................................................................................... 54
 The NIS Domain ...................................................................................... 54
 NIS Daemons ........................................................................................... 54
Contents

NIS Commands ............................................................................................... 55
NIS Maps ........................................................................................................ 56
NIS Binding ..................................................................................................... 60
Server-List Mode .......................................................................................... 61
Broadcast Mode .............................................................................................. 61

6 Setting Up and Configuring Network Information Service ......................... 63
Configuring NIS Task Map ............................................................................ 63
Before You Begin Configuring NIS ................................................................. 64
NIS and the Service Management Facility ..................................................... 64
Planning Your NIS Domain ............................................................................ 66
Identify Your NIS Servers and Clients ........................................................... 66
Preparing the Master Server .......................................................................... 66
Preparing the Master Server (Task Map) ....................................................... 66
Source Files Directory .................................................................................... 67
passwd Files and Namespace Security .......................................................... 67
▼ How to Prepare Source Files for Conversion ............................................. 68
Preparing /var/yp/Makefile .......................................................................... 70
▼ How to Install the NIS Master Server Package ......................................... 71
▼ How to Set Up the Master Server ............................................................. 72
▼ How to Support Multiple NIS Domains on One Master Server ................. 73
Starting and Stopping NIS Services on an NIS Server .................................... 74
Starting and Stopping NIS Services on an NIS Server (Task Map) ................. 74
Starting the NIS Service Automatically .......................................................... 75
▼ How to Enable the NIS Server Services Manually .................................... 75
▼ How to Disable the NIS Server Services ................................................. 76
▼ How to Refresh the NIS Server Service ................................................... 76
Setting Up NIS Slave Servers ........................................................................ 77
Setting Up NIS Slave Servers (Task Map) .................................................... 77
Preparing a Slave Server ............................................................................... 77
▼ How to Set Up a Slave Server ................................................................... 77
▼ How to Start NIS on a Slave Server ......................................................... 79
▼ How to Add a New Slave Server .............................................................. 80
Administering NIS Clients ............................................................................ 82
Administering NIS Clients (Task Map) .......................................................... 82
▼ How to Configure an NIS Client in Broadcast Mode .............................. 83
▼ How to Configure an NIS Client Using Specific NIS Servers ................. 83
▼ Disabling the NIS Client Services ........................................................... 84
Contents

7 Administering Network Information Service ..................................................... 85
   Password Files and Namespace Security ....................................................... 85
   Administering NIS Users ............................................................................. 86
      ▼ How to Add a New NIS User to an NIS Domain .................................. 86
   Setting User Passwords ............................................................................. 87
   NIS Netgroups ......................................................................................... 88
   Working With NIS Maps ........................................................................... 89
      Obtaining Map Information .................................................................... 89
      Changing a Map's Master Server ............................................................. 90
      Modifying Configuration Files ............................................................... 92
      Modifying and Using /var/yp/Makefile .................................................... 92
      Modifying Makefile Entries ................................................................... 94
   Updating and Modifying Existing Maps ....................................................... 96
      ▼ How to Update Maps Supplied With the Default Set ......................... 96
   Maintaining Updated Maps ....................................................................... 97
   Modifying Non-Default Maps .................................................................. 99
   Using the makedbm Command to Modify a Non-Default Map ................. 100
   Creating New Maps From Text Files .......................................................... 100
   Adding Entries to a File-Based Map .......................................................... 100
   Creating Maps From Standard Input .......................................................... 100
   Modifying Maps Made From Standard Input .......................................... 101
   Working With NIS Servers ....................................................................... 101
      Binding to a Specific NIS Server ............................................................. 101
      ▼ How to Set a Machine's NIS Domain Name ...................................... 102
      ▼ How to Configure Machine Host Name and Address Lookup Through NIS
         and DNS .......................................................................................... 103
   Turning Off NIS Services ......................................................................... 104

8 Troubleshooting Network Information System .............................................. 105
   NIS Binding Problems ............................................................................... 105
   Symptoms of NIS Binding Problems .......................................................... 105
   NIS Problems Affecting One Client ............................................................ 106
   NIS Problems Affecting Many Clients ......................................................... 109

Glossary ............................................................................................................. 115

Index .................................................................................................................. 123
Using This Documentation

- **Overview** – Describes the DNS and NIS naming services, methods for planning their use, and steps to implement DNS and NIS.
- **Audience** – System administrators.
- **Required knowledge** – Familiarity with naming service concepts and terminologies that refer to DNS and NIS.

Product Documentation Library

Late-breaking information and known issues for this product are included in the documentation library at http://www.oracle.com/pls/topic/lookup?ctx=E36784.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.
Chapter 1 • About Naming and Directory Services

This chapter provides an overview of naming and directory services included in the Oracle Solaris release. It also briefly describes DNS, NIS, and LDAP naming services.

The following topics are covered in this chapter:

■ “What Is a Naming Service?” on page 9
■ “Oracle Solaris Naming Services” on page 15
■ “Naming Services: A Quick Comparison” on page 18

What Is a Naming Service?

A Naming service performs lookups of stored information, such as:

■ Host names and addresses
■ User names
■ Passwords
■ Access permissions
■ Group membership, automount maps, and so on

This information is made available so that users can log in to their host, access resources, and be granted permissions. The name service information can be stored in files, maps, or various forms of database files. These information repositories can be local to the system or located in a central network-based repository or database.

Without a central naming service, each host would have to maintain its own copy of this information. If you centralize all data, administration becomes easier.

Naming services are fundamental to any computing network. Among other features, naming services provide functionality that does the following.

■ Associates (binds) names with objects
■ Resolves names to objects
■ Removes bindings
What Is a Naming Service?

- Lists names
- Renames information

A network information service enables systems to be identified by common names instead of numerical addresses. This makes communication simpler because users do not have to remember and try to enter cumbersome numerical addresses like 192.168.0.0.

For example, take a network of three systems that are named, pine, elm, and oak. Before pine can send a message to either elm or oak, pine must know their numerical network addresses. For this reason, pine keeps a file, /etc/inet/hosts, that stores the network address of every system in the network, including itself.

```
pine  elm  oak
/etc/inet/hosts
10.0.3.1 pine
10.0.3.2 elm
10.0.3.3 oak
```

Likewise, in order for elm and oak to communicate with pine or with each other, the systems must keep similar files.

```
pine  elm  oak
/etc/inet/hosts  /etc/inet/hosts  /etc/inet/hosts
10.0.3.1 pine  10.0.3.1 pine  10.0.3.1 pine
10.0.3.2 elm  10.0.3.2 elm  10.0.3.2 elm
10.0.3.3 oak  10.0.3.3 oak  10.0.3.3 oak
```

In addition to storing addresses, systems store security information, mail data, network services information and so on. As networks offer more services, the stored list of information grows. As a result, each system might keep an entire set of files that are similar to /etc/inet/hosts.
A network information service stores network information on a server, which can be queried by any system.

The systems are known as clients of the server. The following figure illustrates the client-server arrangement. Whenever information about the network changes, instead of updating each client’s local file, an administrator updates only the information stored by the network information service. Doing so reduces errors, inconsistencies between clients, and the sheer size of the task.

This arrangement, of a server providing centralized services to clients across a network, is known as client-server computing.

Although the main purpose of a network information service is to centralize information, the network information service can also simplify network names. For example, assume your company has set up a network which is connected to the Internet. The Internet has assigned your network the network address 192.168.0.0 and the domain name example.com. Your company has two divisions, Sales and Manufacturing (Manf), so its network is divided into a main network and one subnet for each division. Each net has its own address.
What Is a Naming Service?

Each division could be identified by its network address, as shown above, but descriptive names made possible by naming services would be preferable.

Instead of addressing mail or other network communications to 192.168.0.0, mail could be addressed to example.com. Instead of addressing mail to 192.168.2.0 or 192.168.3.0, mail could be addressed to sales.example.com or manf.example.com.

Names are also more flexible than physical addresses. Physical networks tend to remain stable, but company organization tends to change.

For example, assume that the example.com network is supported by three servers, S1, S2, and S3. Assume that two of those servers, S2 and S3, support clients.
Clients C1, C2, and C3 would obtain their network information from server S2. Clients C4, C5, and C6 would obtain information from server S3. The resulting network is summarized in the following table. The table is a generalized representation of that network but does not resemble an actual network information map.

<table>
<thead>
<tr>
<th>Network Address</th>
<th>Network Name</th>
<th>Server</th>
<th>Clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.0</td>
<td>example.com</td>
<td>S1</td>
<td></td>
</tr>
<tr>
<td>192.168.2.0</td>
<td>sales.example.com</td>
<td>S2</td>
<td>C1, C2, C3</td>
</tr>
<tr>
<td>192.168.3.0</td>
<td>manf.example.com</td>
<td>S3</td>
<td>C4, C5, C6</td>
</tr>
</tbody>
</table>

Now, assume that you create a third division, Testing, which borrowed some resources from the other two divisions, but did not create a third subnet. The physical network would then no longer parallel the corporate structure.
Traffic for the Test Division would not have its own subnet, but would instead be split between 192.168.2.0 and 192.168.3.0. However, with a network information service, the Test Division traffic could have its own dedicated network.

Thus, when an organization changes, its network information service can change its mapping as shown here.

Now, clients C1 and C2 would obtain their information from server S2. C3, C4, and C5 obtain information from server S3.

Subsequent changes in your organization would be accommodated by changes to the network information structure without reorganizing the network structure.
Oracle Solaris Naming Services

The Oracle Solaris platform provides the following naming services:

- Domain Name System (DNS) (see “Description of the DNS Naming Service” on page 15)
- /etc files, the original UNIX naming system (see “Description of the /etc Files Naming Service” on page 16)
- Network Information Service (NIS) (see “Description of the NIS Naming Service” on page 17)

Most modern networks use two or more of these services in combination. The name service switch coordinates the naming service that is used for a particular lookup. See Chapter 2, “About the Name Service Switch” for more information.

Naming Services and the Service Management Facility

In Oracle Solaris, all naming services are now managed by the Service Management Facility (SMF). Configuration information is no longer stored in configuration files but in the SMF repository. Refer to the individual chapters in this guide for more information about how SMF works with a specific naming service.

Legacy configuration files are retained in this Oracle Solaris release only for purposes of compatibility with previous Oracle Solaris releases. Their contents are generated by the SMF service that is relevant to the specific naming service. You should no longer use these files for naming service configuration. Instead, you must use the general SMF commands such as svcs, svcadm, and svccfg.

When you upgrade from Oracle Solaris 10 to Oracle Solaris 11 and its update releases, the system’s name service configuration is automatically migrated to SMF. However, if necessary, manual migration can be performed by using the nscfg command. For more information, refer to the nscfg(1M) man page.

Description of the DNS Naming Service

The Domain Name System (DNS) is a hierarchical, distributed database, implemented on a TCP/IP network. It is primarily used to look up IP addresses for Internet host names and
host names for IP addresses. The data is distributed across the network and is located by using period-separated names that are read from right to left. DNS is also used to store other Internet-related host information, such as mail exchange routing information, location data, and available services. The hierarchical nature of the service enables the local administration of local domains, while providing international coverage of other domains that are connected to the Internet, an intranet, or both.

DNS clients request information about a host name from one or more name servers and wait for a response. DNS servers respond to requests from an information cache that was loaded from any of the following sources:

- A file or a third-party database on a DNS master server
- A file or a third-party database from a cooperating DNS slave server in the network
- Information stored from previous queries

If no response is found and the server is not responsible for the domain in question, the service, if appropriately configured, will recursively request the host name from other servers and cache that response.

**Description of Multicast DNS and Service Discovery**

Two extensions to the DNS protocol are managed by the `svc:network/dns/multicast` service. Multicast DNS (mDNS) implements DNS in a small network where no conventional DNS server has been installed. DNS Service Discovery (DNS-SD) extends Multicast DNS to also provide simple service discovery (network browsing). For more information, see “Multicast DNS” on page 31 and “Multicast DNS Service Discovery” on page 32.

---

**Caution** - The mDNS service uses the `.local` domain name, so that name should not be also used in DNS to avoid possible conflicts.

---

**Description of the `/etc` Files Naming Service**

The original host-based UNIX naming system was developed for stand-alone UNIX machines and then adapted for network use. Many old UNIX operating systems and machines still manage all naming data by using only local files in `/etc`. However, managing hosts, users, and other naming data by using local files is not well suited for large complex networks. For a description of each file, refer to their associated man pages. For example, the `/etc/inet/hosts` file is described in the `hosts(4)` man page.
Description of the NIS Naming Service

The *Network Information Service* (NIS) was developed independently of DNS. DNS makes communication simpler by using machine names instead of numerical IP addresses. NIS focuses on making network administration more manageable by providing centralized control over a variety of network information. NIS stores information about the network, machine names and addresses, users, and network services. This collection of network information is referred to as the *NIS namespace*.

NIS namespace information is stored in NIS maps. NIS maps were designed to replace UNIX `/etc` files, as well as other configuration files. NIS maps store much more than names and addresses. As a result, the NIS namespace has a large set of maps. See “Working With NIS Maps” on page 89 for more information.

NIS uses a client-server arrangement which is similar to DNS. Replicated NIS servers provide services to NIS clients. The principal servers are called *master* servers, and for reliability, the servers have backup, or *slave* servers. Both master and slave servers use the NIS retrieval software and both store NIS maps. For more information on NIS Architecture and NIS Administration, see Chapter 6, “Setting Up and Configuring Network Information Service” and Chapter 7, “Administering Network Information Service”.

Description of the LDAP Naming Services

The Lightweight Directory Access Protocol (LDAP) is the secure network protocol used to access directory servers for distributed naming and other directory services. This standard based protocol supports a hierarchal database structure. The same protocol can be used to provide naming services in both UNIX and multi-platform environments.

The Oracle Solaris OS supports LDAP in conjunction with the Oracle Directory Server Enterprise Edition (formerly Sun Java System Directory Server), as well as other LDAP directory servers.

LDAP is described in “Working With Oracle Solaris 11.2 Directory and Naming Services: LDAP”, including instructions to transition from NIS to LDAP.

For information about single sign-on, as well as the setup and maintenance of Kerberos authentication services, see Chapter 2, “About the Kerberos Service,” in “Managing Kerberos and Other Authentication Services in Oracle Solaris 11.2”.

Chapter 1 • About Naming and Directory Services
Description of the Name Service Switch

The name service switch is a mechanism to allow clients to search through the DNS, LDAP, NIS or local files data sources for naming information. The switch is managed through the `svc:/system/name-service/switch` service. For more information, see Chapter 2, “About the Name Service Switch”.

Naming Services: A Quick Comparison

<table>
<thead>
<tr>
<th></th>
<th>DNS</th>
<th>NIS</th>
<th>LDAP</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Namespace</strong></td>
<td>Hierarchical</td>
<td>Flat</td>
<td>Hierarchical</td>
<td>Files</td>
</tr>
<tr>
<td><strong>Data Storage</strong></td>
<td>Files/resource records</td>
<td>Two column maps</td>
<td>Directories (varied)</td>
<td>Text-based files</td>
</tr>
<tr>
<td><strong>Servers</strong></td>
<td>Master/slave</td>
<td>Master/slave</td>
<td>Master/replica</td>
<td>None</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>DNSSEC, varied</td>
<td>None (root or nothing)</td>
<td>Kerberos, TLS, SSL, varied</td>
<td>None</td>
</tr>
<tr>
<td><strong>Transport</strong></td>
<td>TCP/IP</td>
<td>RPC</td>
<td>TCP/IP</td>
<td>File I/O</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Global</td>
<td>LAN</td>
<td>Global</td>
<td>Local host only</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>Host</td>
<td>All</td>
<td>All</td>
<td>All</td>
</tr>
</tbody>
</table>

**Note** - DNS is the recommended service for host or network address lookups for LDAP and files-based naming.

IPv6 Extensions to Oracle Solaris Name Services

This section describes naming changes that were introduced by the implementation of IPv6. You can store IPv6 addresses in any of the Oracle Solaris naming services, NIS, LDAP, DNS, and files. You can also use NIS over IPv6 RPC transports to retrieve any NIS data.
DNS Extensions for IPv6

An IPv6-specific resource record, the AAAA resource record, has been specified by in RFC 1886 DNS Extensions to Support IP Version 6. This AAAA record maps a host name into a 128 bit IPv6 address. The PTR record is still used with IPv6 to map IP addresses into host names. The 32 four bit nibbles of the 128 bit address are reversed for an IPv6 address. Each nibble is converted to its corresponding hexadecimal ASCII value. Then, ip6.arpa is appended.
About the Name Service Switch

This chapter describes the name service switch. You use the name service switch to coordinate usage of different naming services. The following topics are covered in this chapter:

- “Overview of the Name Service Switch” on page 21
- “Configuring the Name Service Switch” on page 27
- “Name Service Switch and Password Information” on page 30

Overview of the Name Service Switch

The name service switch is a configurable selection service that enables an administrator to specify which name information service or source to use for each type of network information. The services are called a database. The name service switch is used by client applications that call any of the `getXbyY` interfaces, such as the following.

- `gethostbyname`
- `getpwuid`
- `getpwnam`
- `getaddrinfo`

Each system has its own configuration in an SMF repository. Each property defined in the name service switch identifies a particular database, such as a host, password, or group. The value assigned to each property lists one or more sources from which to request the information. Sometimes, these values include guidance or options. The guidance might include how many retries to a service should be attempted, what timeout to apply, or what to do if the service fails.

The name service switch also controls DNS forwarding for clients as described in Chapter 3, “Managing Domain Name System”. DNS forwarding grants Internet access to clients.
Databases and Sources for the Name Service Switch

The databases that are supported by the name service switch are configured by using SMF services. To obtain a listing of these databases, use the `svccfg` command as shown in the following example.

```bash
# svccfg -s name-service/switch listprop config
```

<table>
<thead>
<tr>
<th>Database</th>
<th>Type of Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias</td>
<td>Email addresses and aliases</td>
</tr>
<tr>
<td>auth_attr</td>
<td>Authorization names and descriptions</td>
</tr>
<tr>
<td>automount</td>
<td>Information about remote file systems that could be mounted locally</td>
</tr>
<tr>
<td>bootparam</td>
<td>Boot information for diskless clients</td>
</tr>
<tr>
<td>ether</td>
<td>Ethernet addresses and matching host names</td>
</tr>
<tr>
<td>group</td>
<td>Information about groups that can be used to share access to files</td>
</tr>
<tr>
<td>host</td>
<td>IP address and matching host names</td>
</tr>
<tr>
<td>netgroup</td>
<td>Information for shared NFS file systems</td>
</tr>
<tr>
<td>netmask</td>
<td>Network masks used to implement IP subnets</td>
</tr>
</tbody>
</table>
### Information Database

<table>
<thead>
<tr>
<th>Information Database</th>
<th>Type of Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Name and number for each network</td>
</tr>
<tr>
<td>password</td>
<td>User account information</td>
</tr>
<tr>
<td>prof_attr</td>
<td>Execution profile names, descriptions, and other attributes</td>
</tr>
<tr>
<td>project</td>
<td>Project names, unique identifiers, and associated resource allocations</td>
</tr>
<tr>
<td>protocol</td>
<td>Internet protocol names, numbers and any aliases</td>
</tr>
<tr>
<td>publickey</td>
<td>Public key information</td>
</tr>
<tr>
<td>rpc</td>
<td>Names and numbers of RPC programs</td>
</tr>
<tr>
<td>service</td>
<td>Name, port, and protocol for Internet services</td>
</tr>
<tr>
<td>tnrhdb</td>
<td>Security attributes for a host using the Trusted Extensions feature of Oracle Solaris</td>
</tr>
<tr>
<td>tnrhtp</td>
<td>Templates used by Trusted Extensions</td>
</tr>
</tbody>
</table>

In addition, a **default** property in the name service switch defines the source string for any database that is not otherwise defined. The value for this property is set to `files` to indicate that all the databases and their information are found locally in the `/etc` directory. You can set up a different configuration for the `default` property based on the sources listed in Table 2-2. See “How to Change the Source for All Naming Databases” on page 29 for the procedure.

The `default` property enables you to configure a source that universally applies to the databases, instead of configuring each database's source.

The following table describes the kind of sources that can be listed in the name service switch for the databases listed above.

### TABLE 2-2 Information Sources for the Name Service Switch

<table>
<thead>
<tr>
<th>Information Sources</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ad</td>
<td>Identifies databases stored on an Active Directory server.</td>
</tr>
<tr>
<td>pam_list</td>
<td>Replaces the obsoleted compat database. It can be used for password and group information to support old-style + or - syntax in the <code>/etc/passwd</code>, <code>/etc/shadow</code>, and <code>/etc/group</code> files.</td>
</tr>
<tr>
<td>files</td>
<td>Specifies that host information be obtained from DNS.</td>
</tr>
<tr>
<td>ldap</td>
<td>Specifies that entries be obtained from the LDAP directory.</td>
</tr>
<tr>
<td>mdns</td>
<td>Specifies hosts information by using Multicast DNS (mDNS).</td>
</tr>
</tbody>
</table>
Source Formats for the Name Service Switch

The following search criteria formats can be used to select one or more information sources, and to specify the order that the sources are used.

- **Single Source** — If an information type has only one source, such as files, a search routine that uses the switch searches for the information in that source only. If the routine finds the information, the routine returns a success status message. If the routine does not find the information, the routine stops searching and returns a different status message. What the routine does with the status message varies from routine to routine.

- **Multiple Sources** — If a database contains multiple sources for a given information type, the switch directs the search routine to search in the first listed source. If the routine finds the information, the routine returns a success status message. If the routine does not find the information in the first source, the routine tries the next source. The routine searches all sources until the routine has found the information, or until the routine is halted by a return specification. If all of the listed sources are searched without finding the information, the routine stops searching and returns a non-success status message.

By default in the Oracle Solaris 11 release, the first source is files. This configuration prevents system hangs if the next source listed is not available.

Status Messages for the Name Service Switch

If a routine finds the information, the routine returns a success status message. If the routine does not find the information, the routine returns one of three error status messages. Possible status messages are listed in the following table.

<table>
<thead>
<tr>
<th>Status Message</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESS</td>
<td>The requested entry was found in the specified source.</td>
</tr>
<tr>
<td>UNAVAIL</td>
<td>The source is either unresponsive or unavailable. In other words, none of the database sources could be found or accessed.</td>
</tr>
<tr>
<td>NOTFOUND</td>
<td>The source responded with “No such entry.” In other words, the database was accessed, but the needed information was not found.</td>
</tr>
</tbody>
</table>
Overview of the Name Service Switch

Chapter 2 • About the Name Service Switch

Status Message | Explanation
--- | ---
TRYAGAIN | The source is busy and might respond next time. In other words, the database was found but could not respond to the query.

Switch Action Options for the Name Service Switch

You can instruct the name service switch to respond to status messages with either of the two actions shown in the following table.

**TABLE 2-4** Responses to Status Messages from the Name Service Switch

<table>
<thead>
<tr>
<th>Action</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>Stop looking for the information.</td>
</tr>
<tr>
<td>continue</td>
<td>Try the next source.</td>
</tr>
</tbody>
</table>

In addition, for the TRYAGAIN status message, the following actions can be defined

- **forever** – Retries the current source indefinitely
- **n** – Retry the current source n more times

Default Search Criteria for the Name Service Switch

The combination of the name service switch status message and action options determine what the search routine does at each step. The combination of the status message and action options make up the search criteria.

The switch’s default search criteria are the same for every source. This list includes a description of several of the search criteria.

- **SUCCESS=return**. Stop looking for the information. Proceed using the information that has been found.
- **UNAVAIL=continue**. Go to the next name service switch source and continue searching. If this source is the last or only source, return with a NOTFOUND status.
- **NOTFOUND=continue**. Go to the next name service switch source and continue searching. If this source is the last or only source, return with a NOTFOUND status.
- **TRYAGAIN=forever**. Searches the current name service switch source indefinitely.
- **TRYAGAIN=3**. Searches the current source three times. After exhausting three retries, the TRYAGAIN action transitions to continue and searches the next name service switch source.
You can change the default search criteria by explicitly specifying some other criteria by using the `STATUS=action` syntax shown in the preceding list. For the procedure, see “How to Configure a Search Criterion for a Database” on page 28.

**Note** - Lookups in the name service switch are performed in the order in which items are listed. However, password updates are performed in reverse order, unless otherwise specified by using the `passwd -r repository` command. See “Name Service Switch and Password Information” on page 30 for more information.

---

**What If the Syntax Is Wrong?**

The client library routines contain compiled-in default entries that are used if no specific SMF property or default SMF property is defined in the name service switch, or if the property is syntactically incorrect. Typically, these compiled-in defaults are “files” only.

**auto_home and auto_master**

The switch search criteria for the `auto_home` and `auto_master` tables and maps is combined into one category, which is called `automount`.

**timezone and the Name Service Switch**

The `timezone` table does not use the name service switch, so the table is not included in the property list for the switch.

**keyserv and publickey Entries in the Name Service Switch**

**Caution** - You must restart the `keyserv` daemon after you make a change to the name service switch in order for the changes to take effect.

The `keyserv` daemon reads the `publickey` properties in the name service switch only when `keyserv` is started. If you change the name service switch properties, `keyserv` does not register
the changes until the keyserv daemon is restarted by using svcadm refresh svc:/network/rpc/keyserv:default. This command must be run after the properties have been changed and the name-service/switch service has been refreshed so that the property changes are loaded into the SMF repository.

## Configuring the Name Service Switch

When you configure the name service switch, you simultaneously perform the following actions:

- Indicate the source of the databases.
- Specify a search sequence of the sources, if the database has multiple sources.
- Define switch actions for corresponding search statuses, also known as switch criteria.

The name service switch databases or properties are configured with default values. The following procedures explain the steps to configure certain properties differently.

### How to Change the Source for a Database

This procedure explains how to specify a different source for the host database. Assume that the original source configuration for the database are files and NIS, which means that for a host lookup, the local files are searched first, and then NIS. You reconfigure the name service switch to also use DNS in host lookups.

You can use this procedure as a template to configure the sources for other name service switch databases.

**Before You Begin**

Make sure that the name service switch configuration reflects the actual setup of naming service on your system. For example, if you want DNS to be a source for host lookups, then DNS must be configure as well.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **(Optional) Display the current configuration of the host database.**

   ```
   # svcconf -s name-service/switch listprop config/host
   config                      application
   config/host                 astring             "files nis"
   ```
How to Configure a Search Criterion for a Database

The name service switch has default search criteria, as explained in “Default Search Criteria for the Name Service Switch” on page 25.

In this procedure, you want the redefine the search mechanism for the host database when the information is not found at the first source. The search mechanism should stop instead of proceeding to search the next source.

1. Become an administrator.
   
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. (Optional) Display the current configuration of the host database.
   
   # svccfg -s name-service/switch listprop config/host
   
   config                      application
   config/network              astring             "files dns nis"

3. Create a new search criterion for the host database lookups when information is not found at the first source.
   
   # svccfg -s system/name-service/switch
   svc:/system/name-service/switch> setprop config/host = \astring: "files [NOTFOUND=return] dns nis"
   svc:/system/name-service/switch> quit
   
   With this configuration, the search mechanism for the network database uses the default search criteria for the SUCCESS status and UNAVAILABLE status. However, if the information is not found, the search stops immediately.

4. Refresh service for the name service switch.
   
   # svcadm refresh name-service/switch
How to Change the Source for All Naming Databases

Use this procedure to define a common source for all the databases that are used by the name service for lookups. By default, the common source is `files`. In this procedure, you are adding another source.

1. **Become an administrator.**
   
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **(Optional) Display the current configuration of the default property.**

   ```bash
   # svccfg -s name-service/switch listprop config/default
   config                   application
   config/default           astring             files
   ```

3. **Add NIS as a default source.**

   ```bash
   # svccfg -s system/name-service/switch
   svc:/system/name-service/switch> setprop config/default = astring: "files nis"
   svc:/system/name-service/switch> quit
   ```

4. **Refresh the service for the name service switch.**

   ```bash
   # svcadm refresh name-service/switch
   ```

How to Use a Legacy `nsswitch.conf` File

Use this procedure if your existing naming switch configuration still uses the `nsswitch.conf` file. This procedure migrates your naming switch configurations from the file to SMF, the method to configure naming switch in Oracle Solaris.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Copy the `nsswitch.conf` file to a new system.**

   Make sure to name the file `/etc/nsswitch.conf`. 
3. Load the information from the file into the SMF repository.
   
   ```
   # nscfg import -f svc:/system/name-service/switch:default
   ```

4. Refresh the service for the name service switch.
   
   ```
   # svcadm refresh name-service/switch
   ```

Name Service Switch and Password Information

It is possible to include and access password information in multiple repositories, such as files and nis. You can use the `config/password` property in the name service switch to establish the lookup order for that information.

---

**Caution** - files should be the first source in the name services switch for `passwd` information to prevent a denial of service (DoS) attack on the system.

---

In an NIS environment, the `config/password` property in the name service switch should list the repositories in the following order;

```
config/password astring "files nis"
```

---

**Tip** - Listing files first allows the root user to log in, under most circumstances, even when the system encounters some network or naming service issues.

---

Do not maintain multiple repositories for the same user. In most cases, the naming service looks up and returns the first definition only. Duplicate entries usually mask security problems.

For example, having the same user in both files and in the network repository will (depending on the `config/password name-service/switch` configuration) use one login ID over the other. The first matched ID for a given machine will become the ID used for the login session. If an ID is in both files and the network repository, and the network repository has been disabled for security reasons, then any machine where the ID resides and is accessed before the network ID is disabled might now be insecure and vulnerable to insecure and unwanted access.
Managing Domain Name System

This chapter provides information about the DNS server and client services. The following topics are covered:

- “DNS Overview” on page 31
- “DNS and the Service Management Facility” on page 32
- “Administering DNS (Tasks)” on page 34
- “Administering Multicast DNS” on page 40
- “DNS Reference” on page 42

DNS Overview

DNS, as with most networking protocols, has two parts: a service providing answers and a client that queries the service. In the Oracle Solaris operating system, the default DNS service is provided by BIND, from the Internet Systems Consortium (ISC), and its associated daemon named. The DNS client consists of a collection of utilities and libraries.

Multicast DNS

Multicast DNS (mDNS) provides a naming service system that is easy to set up and maintain for systems on a local link. All participating network devices on the same local link perform standard DNS functions, using mDNS rather than unicast, and do not need a unicast DNS server. For administrators, the primary advantage of mDNS is that no unicast DNS server needs to be maintained on the local network. There is no need, for example, to update and maintain host names in files to resolve hostname to IP address requests for systems on the local link that are using mDNS.
Multicast DNS Service Discovery

Network services include printing, file transfer, music sharing, servers for photo, document, and other file sharing, and services provided by other local devices. DNS service discovery support in Oracle Solaris includes an open source framework and tools from Apple Inc. to enable applications to advertise and discover network services using DNS in this Oracle Solaris release.

For users, network service discovery makes computing easier by enabling them to browse for services on the network, rather than needing to find the service manually. Existing standards and work performed by other companies and groups ensure that cross-platform support is available.

Related Materials About DNS

For information about DNS and BIND administration, see the following documentation:

- BIND 9 Migration Notes documentation in the /usr/share/doc/bind/migration.txt file
- Listings of BIND features, known bugs and defects, and links to additional material on the ISC web site at http://www.isc.org
- DNS and Bind (5th Edition), by Paul Albitz and Cricket Liu, (O'Reilly, 2006)

DNS and the Service Management Facility

The DNS server daemon, named, is managed by using the Service Management Facility (SMF). For an overview of SMF, refer to Chapter 1, “Introduction to the Service Management Facility,” in “Managing System Services in Oracle Solaris 11.2.” Also refer to the svcadm(1M), svcs(1), and svccfg(1M) man pages for more details.

The following list provides a short overview of some of the important information needed to use the SMF service to administer the DNS service.

- To perform administrative actions on this service, such as enabling, disabling, or restarting, use the svcadm command.
Tip - Temporarily disabling a service by using the -t option provides some protection for the service configuration. If the service is disabled with the -t option, the original settings are restored for the service after a reboot. If the service is disabled without -t, the service remains disabled after a reboot.

- The Fault Managed Resource Identifiers (FMRIs) for the DNS service are svc:/network/dns/server:instance and svc:/network/dns/client:instance.
- You can query the status of the DNS server and client by using the svcs command.
  - The following is an example of the svcs command and its output:
    ```
    # svcs \*dns\*
    STATE STIME FMRI
    disabled Nov_16 svc:/network/dns/multicast:default
    online Nov_16 svc:/network/dns/server:default
    online Nov_16 svc:/network/dns/client:default
    ```
  - The following is an example of svcs -l command and its output.
    ```
    # svcs -l dns/server
    fmri svc:/network/dns/server:default
    name BIND DNS server
    enabled true
    state online
    next_state none
    state_time Tue Jul 26 19:26:12 2011
    logfile /var/svc/log/network-dns-server:default.log
    restarter svc:/system/svc/restarter:default
    contract_id 83
    manifest /lib/svc/manifest/network/dns/server.xml
    dependency require_all/none svc:/system/filesystem/local (online)
    dependency require_any/error svc:/network/loopback (online)
    dependency optional_all/error svc:/network/physical (online)
    ```
- If you need to start the DNS service with different options, change the properties of the svc:/network/dns/server service by using the svccfg command. For an example, see “How to Configure DNS Server Options” on page 36.

Because the DNS server daemon, named, is managed by SMF, the server is automatically restarted when an unexpected event occurs that causes named to exit abnormally. Additionally, you can use the svcadm command to restart the service. The BIND-specific management that is available by using rndc command can be used simultaneously with SMF.
Administering DNS (Tasks)

The following tasks are documented:

- “How to Install the DNS Package” on page 34
- “How to Configure a DNS Server” on page 34
- “How to Create an `rndc.conf` File” on page 35
- “How to Configure DNS Server Options” on page 36
- “How to Run the DNS Service as an Alternative User” on page 37
- “How to Enable a DNS Client” on page 38
- “How to Troubleshoot DNS Server Startup Issues” on page 38
- “How to Verify the DNS Configuration” on page 39

▼ How to Install the DNS Package

Normally, the DNS package is automatically installed with the Oracle Solaris release. If the package was not included when the server was installed, use the following procedure to install the package.

1. **Become an administrator.**
   
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Install the DNS package.**
   
   ```
   # pkg install pkg:/service/network/dns/bind
   ```

▼ How to Configure a DNS Server

**Note** - Configuring `named` to specify a change root directory is not recommended. A more secure option is to create a Solaris Zone and configure `named` to run within that zone.

1. **Become an administrator.**
For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Create and verify a DNS configuration file.**

   Before the named daemon will start, a valid configuration file must exist. The file is called /etc/named.conf by default. The configuration of named might be very simple. An empty file provides sufficient information to configure a caching only server, assuming that DNS root servers are accessible.

   ```
   # touch /etc/named.conf
   # named-checkconf -z /etc/named.conf
   ```

3. **(Optional) Create an rndc configuration file.**

   This file is used to configure remote control access of the DNS server. See “How to Create an rndc.conf File” on page 35.

4. **(Optional) Change configuration information for the dns/server service.**

   See “How to Configure DNS Server Options” on page 36.

5. **Start the DNS service.**

   ```
   # svcadm enable dns/server
   ```

### How to Create an rndc.conf File

The /etc/rndc.conf file is used to configure remote control access of the DNS server daemon, named, by using the rndc command. To create a default file, use the following procedure. Refer to the **rndc.conf**(4) man page for further options.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Create the rndc configuration file.**

   ```
   # rndc-confgen -a
   wrote key file "/etc/rndc.key"
   ```

3. **(Optional) Restart the DNS service.**
If you are creating the `rndc.conf` file as part of the DNS server configuration, you can skip restarting the DNS service until all the DNS server configuration is completed.

```bash
# svcadm restart dns/server:default
```

## How to Configure DNS Server Options

This procedure explains how to select the IPv4 transport protocol for `named` traffic. See the `named(1M)` man page.

1. **Become an administrator.**
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Change the configuration information for the `dns/server` service.**

   ```bash
   # svcadm -s dns/server:default
   svc:/network/dns/server:default> setprop options/ip_interfaces = "IPv4"
   svc:/network/dns/server:default> quit
   
   Note - You can change the configuration information with a single command.
   
   # svcadm -s dns/server:default options/ip_interfaces=IPv4
   
3. **Update the SMF repository and enable the DNS service.**

   ```bash
   # svcadm refresh network/dns/server:default
   # svcadm enable network/dns/server:default
   
4. **(Optional) Verify the change.**

   ```bash
   # svcadm -s dns/server:default listprop options/ip_interfaces
   options/ip_interfaces  astring  IPv4
   ```
How to Run the DNS Service as an Alternative User

This procedure explains how to assign a user the relevant authorizations to manage the named daemon.

1. **Become an administrator.**
   
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Provide the alternative user with the appropriate authorization.**

   ```bash
   # useradd -c "Trusted DNS administrator user" -s /usr/bin/pfbash \ 
   -A solaris.smf.manage.bind user
   ``

3. **Set service properties for the user.**

   ```bash
   # svccfg -s dns/server:default
   svc:/network/dns/server:default> setprop start/user = user
   svc:/network/dns/server:default> setprop start/group = user
   svc:/network/dns/server:default> exit
   ``

4. **Create a directory for a new process ID file.**

   Because only root has write access to create the default process ID file, `/var/run/named/named.pid`, the named daemon must be configured to use an alternative file.

   ```bash
   # mkdir /var/named/tmp
   # chown dnsadmin /var/named/tmp
   ``

5. **Change the configuration to use the new directory.**

   Add the following lines to the `named.conf` file:

   ```bash
   # head /etc/named.conf
   options {
   directory "/var/named";
   pid-file "/var/named/tmp/named.pid";
   }
   ``

6. **Update the SMF repository and restart the DNS service.**

   ```bash
   # svcadm refresh svc:/network/dns/server:default
   # svcadm restart svc:/network/dns/server:default
   ```
How to Enable a DNS Client

1. Become an administrator.
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. Configure the DNS domain.
   First, list the domains to search and the IP addresses for the DNS name servers. Then, update the SMF repository.
   
   ```
   # svccfg -s dns/client
   svc:/network/dns/client> setprop config/search = astring: ("example.com" "sales.example.com")
   svc:/network/dns/client> setprop config/nameserver = net_address: (192.168.1.10 192.168.1.11)
   svc:/network/dns/client> select network/dns/client:default
   svc:/network/dns/client:default> refresh
   svc:/network/dns/client:default> quit
   ```

3. Update name service switch information to use DNS.
   The first command updates the DNS configuration information in the SMF repository.
   
   ```
   # svccfg -s system/name-service/switch
   svc:/system/name-service/switch> setprop config/host = astring: "files dns"
   svc:/system/name-service/switch> select system/name-service/switch:default
   svc:/system/name-service/switch:default> refresh
   svc:/system/name-service/switch:default> quit
   ```

4. Start the services needed to run the DNS client.
   
   ```
   # svcadm enable network/dns/client
   # svcadm enable system/name-service/switch
   ```

How to Troubleshoot DNS Server Startup Issues

All of these steps do not have to be followed. If you think you find the problem in an early step you may proceed to step 6 to get the service running properly.

1. Become an administrator.
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.
2. **Check the DNS service status.**

   ```
   # svc -x dns/server:default
   svc:/network/dns/server:default (BIND DNS server)
   State: online since Tue Oct 18 19:35:00 2011
   See: named(1M)
   See: /var/svc/log/network-dns-server:default.log
   Impact: None.
   ```

3. **Check the DNS service log file.**

   ```
   # tail /var/svc/log/network-dns-server:default.log
   ```

4. **Check syslog messages.**

   ```
   # grep named /var/adm/messages
   ```

5. **Start the `named` daemon manually.**

   Running `named` in the foreground forces all logging to standard error so that it is easier to identify problems.

   ```
   # named -g
   ```

6. **After the issue has been fixed, clear the maintenance required state.**

   ```
   # svcadm clear dns/server:default
   # svc -s dns/server:default
   STATE          STIME    FMRI
   online         17:59:08 svc:/network/dns/server:default
   ```

Next: How to Verify the DNS Configuration

---

**How to Verify the DNS Configuration**

When modifying the DNS configuration, you can verify the syntax of the `/etc/named.conf` file with the `named-checkzone` command.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Change the configuration file, as needed.**

   In this example, the default directory is changed.

   ```
   # echo 'options {directory "/var/named";};' > /etc/named.conf
   ```

3. **Verify the file contents.**
# named-checkconf
/etc/named.conf:1: change directory to '/var/named' failed: file not found

/etc/named.conf:1: parsing failed

In this example, the check failed because the /var/named directory has not yet been created.

4. Correct any errors reported.

# mkdir /var/named

5. Repeat steps 3 and 4 until no errors are reported.

6. (Optional) To reflect the change in the running service use one of the methods below:

  ■ Use the `rndc` command to update the configuration using the `reload` or `reconfig` option, depending on the changes made.

  ■ Restart the `named` service.

    # svcadm restart svc:/network/dns/server:default

---

**Administering Multicast DNS**

The following sections explain how to enable multicast DNS (mDNS) and DNS service discovery. Also provided are examples of how to advertise resources for DNS service discovery.

▼ **How to Enable mDNS and DNS Service Discovery**

For mDNS and DNS Service Discovery to function, mDNS must be deployed on all systems that are to participate in mDNS. The mDNS service is used to advertise the availability of services provided on the system.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **If needed, install the mDNS package.**

   # pkg install pkg:/service/network/dns/mdns
3. **Update name service switch information.**

   To be able to resolve local hosts, change the `config/host` property of the `name-service/switch` service to include `mdns` as a source. For example:

   ```
   # /usr/sbin/svccfg -s svc:/system/name-service/switch
   svc:/system/name-service/switch> setprop config/host = astring: "files dns mdns"
   svc:/system/name-service/switch> select system/name-service/switch:default
   svc:/system/name-service/switch:default> refresh
   svc:/system/name-service/switch> quit
   ```

4. **Enable the mDNS service.**

   ```
   svcadm enable svc:/network/dns/multicast:default
   ```

   Enabling mDNS in this way ensures that your changes persist through upgrades and reboots. For more information, see the `svcadm(1M)` man page.

5. **(Optional) If needed, check the mDNS error log.**

   Check the mDNS service log, `/var/svc/log/network-dns-multicast:default.log`, for errors or messages.

### Advertising Resources for DNS

You can use the `dns-sd` command as a network diagnosis tool, to browse and discover services, similar to how you would use the `ping` or `traceroute` command.

The `dns-sd` command is primarily for interactive use, mainly because its command-line arguments and its output format can change over time, which makes invoking it from a shell script unpredictable and risky. Additionally, the asynchronous nature of DNS service discovery (DNS-SD) does not easily lend itself to script-oriented programming.

For complete information, see the `dns-sd(1M)` man page. To incorporate the DNS service in applications, see the `libdns-sd(3DNS_SD)` man page.

The following are examples of advertising services using DNS service discovery.

**EXAMPLE 3-1  Advertising a Printing Service**

The following command advertises the existence of LPR printing service on port 515 on a system called `My Test`, so that it will be available to DNS-SD compatible printing clients:

```
# dns-sd -R "My Test" _printer._tcp. 515 pdl=application/postscript
```

For this registration to be useful, the LPR service must be available on port 515.
EXAMPLE 3-2 Advertising a Web Page

The following command advertises a web page being served by an HTTP server on port 80 on the My Test system. The web page will appear on the Bonjour list in Safari and other DNS-SD compatible web clients.

```bash
$ dns-sd -R "My Test" _http._tcp . 80 path=/path-to-page.html
```

DNS Reference

This section includes tables of the files, daemons, and commands that are associated with the DNS service. In addition, a table of some of the flags that are used when the ISC version of BIND was built is included.

DNS Files

The following table describes the files that are associated with the DNS service.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/named.conf</td>
<td>Provides configuration information for the named daemon. See the named. conf(4) man page for more information.</td>
</tr>
<tr>
<td>/etc/rndc.conf</td>
<td>Provides configuration information for the rndc command. See the rndc. conf(4) man page for more information.</td>
</tr>
</tbody>
</table>

DNS Commands and Daemons

The following table describes the commands and daemons that are associated with the DNS service.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/bin/dns-sd</td>
<td>Finds or lists resources used by the mDNS service. See the dns-sd(1M) man page for more information.</td>
</tr>
<tr>
<td>/usr/sbin/dig</td>
<td>Requests DNS responses from a DNS server. Often used to troubleshoot. See the dig(1M) man page for more information.</td>
</tr>
</tbody>
</table>
### Compilation Flags Used When BIND Was Built

You can view the flags that were used to compile BIND by using the `named -V` command. This table shows some of the compilation flags that were used when building the ISC version of BIND for the Oracle Solaris 11 release.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/usr/sbin/dnssec-dsfromkey</code></td>
<td>Generates a DS RR from a key file. See the <code>dnssec-dsfromkey(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/dnssec-keyfromlabel</code></td>
<td>Retrieves selected keys from cryptographic device and builds a key file. See the <code>dnssec-keygen(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/dnssec-keygen</code></td>
<td>Creates keys and key files for secure DNS and for transaction signatures (TSIG). See the <code>dnssec-keygen(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/dnssec-signzone</code></td>
<td>Signs a DNS zone. See the <code>dnssec-signzone(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/host</code></td>
<td>Performs simple DNS lookups, often converting host names to IP addresses or IP addresses to host names. See the <code>host(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/named</code></td>
<td>DNS server daemon, which responds to information requests from clients. See the <code>named(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/named-checkconf</code></td>
<td>Checks the syntax of the <code>named.conf</code> file. See the <code>named(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/named-checkzone</code></td>
<td>Checks the syntax and integrity of a DNS zone file. See the <code>named-checkzone(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/named-compilezone</code></td>
<td>Converts a DNS zone file. See the <code>named-compilezone(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/nscfg</code></td>
<td>Legacy name service configuration utility, which imports or exports name service configuration between legacy name service configuration files and the SMF repository. See the <code>nscfg(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/nslookup</code></td>
<td>Deprecated: Queries the DNS server. Instead use the <code>dig</code> command.</td>
</tr>
<tr>
<td><code>/usr/sbin/nsupdate</code></td>
<td>Submits DNS update requests to a DNS server. See the <code>nsupdate(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/rndc</code></td>
<td>Provides remote control of the DNS server daemon. See the <code>rndc(1M)</code> man page for more information.</td>
</tr>
<tr>
<td><code>/usr/sbin/rndc-confgen</code></td>
<td>Generates configuration files for the <code>rndc</code> command. See the <code>rndc-confgen(1M)</code> man page for more information.</td>
</tr>
<tr>
<td>Flag Name</td>
<td>Function</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>with-openssl</td>
<td>Builds BIND with cryptographic and Secure Sockets Layer (SSL) support, which is required for DNSSEC</td>
</tr>
<tr>
<td>enable-threads</td>
<td>Enables multithreading</td>
</tr>
<tr>
<td>enable-devpoll</td>
<td>Uses the /dev/poll driver for fast poll on many file descriptors</td>
</tr>
<tr>
<td>disable-openssl-version-check</td>
<td>Disables the OpenSSL version check because OpenSSL is provided by a separate dynamic library</td>
</tr>
<tr>
<td>enable-fixed-rrset</td>
<td>Enables fixed resource record set ordering, which is needed for backward compatibility</td>
</tr>
<tr>
<td>with-pkcs11</td>
<td>Enables the use of OpenSSL cryptographic hardware support</td>
</tr>
</tbody>
</table>
Setting Up Oracle Solaris Active Directory Clients

The `nss_ad` naming service module provides a back end for the `passwd`, `shadow`, and `group` files. The `nss_ad` module uses Active Directory (AD) and its native schema as the naming service to resolve user and group names and IDs from across an AD forest. The following topics are included:

- “Overview of the `nss_ad` Naming Service Module” on page 45
- “Password Updates” on page 48
- “How the `nss_ad` Naming Service Module Retrieves Data From AD” on page 48

Overview of the `nss_ad` Naming Service Module

The Oracle Solaris client must be joined to an AD domain before any of the AD interoperability functionality, including `nss_ad`, can be used. The `kclient` utility is used to join the client to AD. During the join operation, `kclient` configures Kerberos v5 on the client. Thereafter, `nss_ad` can be used to resolve naming service requests by specifying `ad` as a source in the `nsswitch.conf` file for the supported databases. The `nss_ad` module uses host credentials to look up naming service information in AD.

The `nss_ad` module uses DNS server records to auto-discover AD directory servers, such as domain controllers and global catalog servers. Therefore, DNS must be properly configured on the Oracle Solaris client. The `nss_ad` module also uses the LDAP v3 protocol to access naming information from AD servers. The AD server schema requires no modification because `nss_ad` works with the native AD schema.

The `nss_ad` module does not currently support logins of Windows users onto an Oracle Solaris system. Until such logins are supported, such users should continue to log in by using traditional back ends such as `nis` and `ldap`.

The `idmap` and `svc:/system/name-service/cache` services must be enabled to use `nss_ad`. The `nss_ad` module uses the `idmap` service to map between Windows security identifiers (SIDs), UNIX user identifiers (UIDs), and group identifiers (GIDs).
Ensure that all AD user and group names are qualified with domain names such as user@domain or group@domain. For example, getpwnam(dana) will fail, but getpwnam(dana@domain) will succeed, provided that dana is a valid Windows user in the domain named domain.

The following additional rules also pertain to the nss_ad module:

- Like AD, nss_ad performs case-insensitive matching of user and group names.
- Only use the nss_ad module in UTF-8 locales or in domains where users and groups have only ASCII characters in their names.
- Well-known SIDs are a set of SIDs that identify generic users or generic groups in the Windows world. They are not domain specific and their values remain constant across all Windows operating systems. The names of well-known SIDs are qualified with the string BUILTIN, for example, Remote Desktop Users@BUILTIN.
- The nss_ad module does not support enumeration. Therefore, the getpwent and getgrent interfaces and commands that use them such as getent passwd and getent group cannot retrieve information from AD.
- The nss_ad module currently supports only the passwd and group files. nss_ad does not support other naming service databases that follow the passwd entry, such as audit_user and user_attr. If the ad back end is processed (based on the configuration), it returns NOT FOUND for these databases.

How to Configure the nss_ad Module

The nss_ad module requires that the Oracle Solaris client use DNS for host resolution.

1. **Configure the DNS service.**

   See “How to Enable a DNS Client” on page 38 for instructions.

   ![Note](Note) The AD domain name must be specified either by means of the domain directive or as the first item in the list specified by the search directive.

   If both directives are specified, then whichever is last takes precedence. This is required for the idmap auto-discovery feature to work properly.

In the following example, the dig commands verify that the AD server can be resolved by using its name and IP address.

```
# dig -x 192.168.11.22 +short
myserver.ad.example
# dig myserver.ad.example +short
192.168.11.22
```
2. **Add dns to the list of naming services for hosts.**

   # svcadm enable svc:/system/name-service/switch
   svc:/system/name-service/switch> setprop config/host = astring: "files dns"
   svc:/system/name-service/switch> select system/name-service/switch:default
   svc:/system/name-service/switch:default> refresh
   svc:/system/name-service/switch:default> quit

   **Note** - To include additional naming services such as nis or ldap for host resolution, add them after dns.

3. **Verify that the DNS service is enabled and online.**

   For example:

   # svcscs svc:/network/dns/client

   STATE STIME FMRI
   online Oct_14 svc:/network/dns/client:default

4. **Use the kclient utility to join the system to the AD domain.**

   For example:

   # /usr/sbin/kclient -T ms_ad

5. **Add ad to the list of naming services for password and group.**

   # svcadm enable svc:/system/name-service/switch
   svc:/system/name-service/switch> setprop config/password = astring: "files nis ad"
   svc:/system/name-service/switch> setprop config/group = astring: "files nis ad"
   svc:/system/name-service/switch> select system/name-service/switch:default
   svc:/system/name-service/switch:default> refresh
   svc:/system/name-service/switch:default> quit

6. **Enable the idmap service.**

   # svcadm enable svc:/system/name-service/switch
   svc:/system/name-service/switch> setprop config/password = astring: "files nis ad"
   svc:/system/name-service/switch> setprop config/group = astring: "files nis ad"
   svc:/system/name-service/switch> select system/name-service/switch:default
   svc:/system/name-service/switch:default> refresh
   svc:/system/name-service/switch:default> quit

7. **Update the SMF repository for the name service switch service.**

   # svcadm refresh svc:/system/name-service/switch

   **Note** - The nscd module automatically restarts if necessary, whenever name service switch is refreshed.

8. **Verify that you can access user and group information from AD.**
Password Updates

For example:

```
# getent passwd 'test_user@example'
test_user@example:x:2154266625:2154266626:test_user::
# getent passwd 2154266625
test_user@example:x:2154266625:2154266626:test_user::
```

Password Updates

The **passwd(4)** man page contains a list of valid formats for the config/passwd property in the name service switch. Adding ad to these configurations is supported. However, changing AD user passwords through the passwd command is not supported. If found in the passwd entry during a password update, ad is skipped. Use the kpasswd command to update AD user passwords.

The ad search order can be added to existing valid password and group entries in name service switch. For example:

```
# svcadm -s svc:/system/name-service/switch
svc:/system/name-service/switch> setprop config/password = astring: "files nis ad"
svc:/system/name-service/switch> setprop config/group = astring: "files nis ad"
svc:/system/name-service/switch> select system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch:default> quit
```

How the nss_ad Naming Service Module Retrieves Data From AD

The following section describes how the nss_ad module resolves naming service requests for the passwd, shadow, and group files by retrieving corresponding data from AD.

**Retrieving passwd Information**

The following syntax shows the proper form of a passwd entry:

```
```

See the **passwd(4)** man page for more information.

The nss_ad module retrieves passwd information from AD as follows:
How the nss_ad Naming Service Module Retrieves Data From AD

- **username** – Field uses the value of the `samAccountName` AD attribute and is qualified by the domain name in which the object resides, for example, johnd@example.com.
- **password** – Field uses the value of `x` because the user password is not available in the AD object.
- **uid** – Field uses the Windows user's SID from the `objectSID` AD attribute, which is mapped to the UID by using the `idmap` service.
- **gid** – Field uses the Windows user's primary group SID, which is mapped to the GID by using the `idmap` service. The group SID is obtained by appending the value of the `primaryGroupID` AD attribute to the domain SID. For users in AD, the `primaryGroupID` attribute is an optional attribute, so it might not exist. If the attribute does not exist, `nss_ad` uses the `idmap` diagonal mapping facility to map the user SID from the `objectSID` attribute.
- **gecos** – Value of the `CN` AD attribute.
- **home-directory** – Value of the `homeDirectory` AD attribute, if a value exists. Otherwise, the field is left empty.
- **login-shell** – Field is left empty because there is no login shell attribute in the native AD schema.

Retrieving shadow Information

The following syntax shows the proper form of a shadow entry:

```
```

See the `shadow(4)` man page for more information.

The `nss_ad` module retrieves shadow information from AD as follows:

- **username** – Field uses the value of the `samAccountName` AD attribute and is qualified by the domain name in which the object resides, for example, johnd@example.com.
- **password** – Field uses the value of `*NP*` because the user password is not available in the AD object.

The rest of the shadow fields are left empty because shadow fields are irrelevant with AD and Kerberos v5.

Retrieving group Information

The following syntax shows the proper form of a group entry:

```
-groupname:password:gid:user-list
```
See the `group(4)` for man page for more information.

The `nss_ad` module retrieves information from AD as follows:

- `groupname` – Field uses the value of the `samAccountName` AD attribute and is qualified by the domain name in which the object resides, for example, `admins@example`.
- `password` – Field is left empty because the Windows groups do not have passwords.
- `gid` – Field uses the Windows group's SID from the `objectSID` AD attribute, which is mapped to the GID by using the `idmap` service.
- `user-list` – Field is left empty.
This chapter provides an overview of the Network Information Service (NIS).

NIS is a distributed naming service. It is a mechanism for identifying and locating network objects and resources. It provides a uniform storage and retrieval method for network-wide information in a transport-protocol and media-independent fashion.

This chapter covers the following topics:

- “NIS Introduction” on page 51
- “NIS Machine Types” on page 53
- “NIS Elements” on page 54
- “NIS Binding” on page 60

**NIS Introduction**

By running NIS, the system administrator can distribute administrative databases, called maps, among a variety of servers (master and slaves). The administrator can update those databases from a centralized location in an automatic and reliable fashion to ensure that all clients share the same naming service information in a consistent manner throughout the network.

NIS was developed independently of DNS and has a slightly different focus. Whereas DNS focuses on making communication simpler by using machine names instead of numerical IP addresses, NIS focuses on making network administration more manageable by providing centralized control over a variety of network information. NIS stores information not only about machine names and addresses, but also about users, the network itself, and network services. This collection of network information is referred to as the NIS namespace.

**Note** - In some contexts *machine* names are referred to as *host* names or *machine* names. This discussion uses *machine*, but some screen messages or NIS map names might use *host* or *machine*. 
NIS Architecture

NIS uses a client-server arrangement. NIS servers provide services to NIS clients. The principal server is called a master server, and for reliability, it can have several backup servers or slave servers. Both master and slave servers use the NIS information retrieval software, and both store NIS maps.

NIS uses domains to arrange the machines, users, and networks in its namespace. However, it does not use a domain hierarchy. An NIS namespace is flat.

![Diagram of an example domain](image)

Thus, this physical network would be arranged into one NIS domain.

An NIS domain cannot be connected directly to the Internet using just NIS. However, organizations that want to use NIS and also be connected to the Internet can combine NIS with DNS. You can use NIS to manage all local information and use DNS for Internet host lookup. NIS also provides a forwarding service that forwards host lookups to DNS if the information cannot be found in an NIS map. The Oracle Solaris system also allows you to set up the name service switch service so that hosts lookup requests can be directed in the following ways:

- To access only DNS
- To access DNS, but if a host is not found in DNS, then access NIS
- To access NIS, but if a host is not found by NIS, then access DNS

For maximum interoperability, DNS is the recommended service for host lookups. See Chapter 2, “About the Name Service Switch” for details.
NIS Machine Types

There are three types of NIS machines.

- Master server
- Slave servers
- Clients of NIS servers

Any machine can be an NIS client, but only machines with disks should be NIS servers, either master or slave. Servers are also clients, typically of themselves.

NIS Servers

NIS servers come in two varieties, master and slave. The machine designated as master server contains the set of maps that the system administrator creates and updates as necessary. Each NIS domain must have one, and only one, master server, which can propagate NIS updates with the least performance degradation.

You can designate additional NIS servers in the domain as slave servers. A slave server has a complete copy of the master set of NIS maps. Whenever the master server maps are updated, the updates are propagated among the slave servers. Slave servers can handle any overflow of requests from the master server, minimizing “server unavailable” errors.

Normally, the system administrator designates one master server for all NIS maps. However, because each individual NIS map has the machine name of the master server encoded within it, you could designate different servers to act as master and slave servers for different maps. To minimize confusion, designate a single server as the master for all the maps you create within a single domain. The examples in this chapter assume that one server is the master for all maps in the domain.

NIS Clients

NIS clients run processes that request data from maps on the servers. Clients do not make a distinction between master and slave servers, since all NIS servers should have the same information.

Note - The Oracle Solaris OS does not support a configuration in which an NIS client and a native LDAP client coexist on the same client system.
The NIS naming service is composed of the following elements:

- Domains (see “The NIS Domain” on page 54)
- Daemons (see “NIS Daemons” on page 54)
- Commands (see “NIS Commands” on page 55)
- Maps (see “NIS Maps” on page 56)

The NIS Domain

An NIS domain is a collection of hosts which share a common set of NIS maps. Each domain has a domain name, and each machine sharing the common set of maps belongs to that domain.

NIS domains and DNS domains are not necessarily the same. In some environments, NIS domains are defined based on enterprise-wide network subnet administrative layouts. DNS names and domains are defined by internet DNS naming standards and hierarchies. The two naming domain naming systems might be or might not be configured to match up identically. The domain name for the two services are controlled separately and might be configured differently.

Any host can belong to a given domain, as long as there is a server for that domain's maps in the same network or subnet. NIS domain lookups use remote procedure calls (RPCs). Therefore, NIS requires that all the clients and all the server machines that provide direct services to those clients must exist on the same accessible subnet. It is not uncommon to have each administrative subnet managed as a separate NIS domain (distinct from an enterprise-wide DNS domain) but using common databases managed from a common master machine. The NIS domain name and all the shared NIS configuration information is managed by the `svc:/network/nis/domain` SMF service.

NIS Daemons

The NIS service is provided by the daemons shown in the following table. The NIS service is managed by SMF. Administrative actions on this service, such as enabling, disabling, or restarting, can be performed by using the `svcadm` command. For an overview of SMF, refer to Chapter 1, “Introduction to the Service Management Facility,” in “Managing System Services in Oracle Solaris 11.2”. Also refer to the `svcadm(1M)` and `svcs(1)` man pages for more details.
TABLE 5-1  NIS Daemons

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nscd</td>
<td>A client service that provides a cache for most name service requests, which is managed by the svc:/system/name-service/cache service</td>
</tr>
<tr>
<td>rpc.yppasswdd</td>
<td>The NIS password update daemon managed by the svc:/network/nis/passwd service</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> - The rpc.yppasswdd daemon considers all shells that begin with an r to be restricted. For example, if you are in /bin/rksh, you are not allowed to change from that shell to another shell. If you have a shell that begins with r but is not intended to be restricted as such, refer to Chapter 8, “Troubleshooting Network Information System” for the workaround.</td>
</tr>
<tr>
<td>rpc.ypupdated</td>
<td>A daemon that modifies other maps such as publickey and is managed by the svc:/network/nis/update service</td>
</tr>
<tr>
<td>ypbind</td>
<td>The binding process managed by the svc:/network/nis/client service</td>
</tr>
<tr>
<td>ypserv</td>
<td>The server process managed by the svc:/network/nis/server service</td>
</tr>
<tr>
<td>ypxfrd</td>
<td>A high-speed map transfer daemon managed by the svc:/network/nis/xfr service</td>
</tr>
</tbody>
</table>

NIS Commands

The NIS service is supported by several commands, which are described in the following table.

TABLE 5-2  NIS Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>make</td>
<td>Updates NIS maps by reading /var/yp/Makefile (when the command is run in the /var/yp directory). You can use make to update all maps based on the input files or to update individual maps. The ypmake(1M) man page describes the functionality of make for NIS.</td>
</tr>
<tr>
<td>makedbm</td>
<td>Takes an input file and converts it into dbm.dir and dbm.pag files. NIS uses valid dbm files as maps. You can also use makedbm -u to disassemble a map so that you can see the key-value pairs that comprise it.</td>
</tr>
<tr>
<td>ypcat</td>
<td>Displays the contents of an NIS map.</td>
</tr>
<tr>
<td>yppoll</td>
<td>Shows which version of an NIS map is running on a server that you specify. It also lists the master server for the map.</td>
</tr>
<tr>
<td>ypinit</td>
<td>Automatically creates maps for an NIS server from the input files. It is also used to construct the initial /var/yp/binding/domain/ypservers file on the clients. Use ypinit to set up the master NIS server and the slave NIS servers for the first time.</td>
</tr>
<tr>
<td>ypmatch</td>
<td>Prints the value for one or more specified keys in an NIS map. You cannot specify which version of the NIS server map you are seeing.</td>
</tr>
</tbody>
</table>
### NIS Maps

The information in NIS maps is stored in ndbm format. The `ypfiles(4)` and `ndbm(3C)` man pages explain the format of the map file.

NIS maps extend access to UNIX `/etc` data and other configuration files, such as `passwd`, `shadow` and `group` so that the same data can be shared between a network of systems. Sharing these files simplifies administrative updates and management of those data files. NIS is deployable with minimal effort. However, larger enterprises, especially those with security requirements should consider using LDAP naming services instead. On a network running NIS, the NIS master server for each NIS domain maintains a set of NIS maps for other machines in the domain to query. NIS slave servers also maintain duplicates of the master server's maps. NIS client machines can obtain namespace information from either master or slave servers.

NIS maps are essentially two-column tables. One column is the key and the other column is information related to the key. NIS finds information for a client by searching through the keys. Some information is stored in several maps because each map uses a different key. For example, the names and addresses of machines are stored in two maps: `hosts.byname` and `hosts.byaddr`. When a server has a machine's name and needs to find its address, it looks in the `hosts.byname` map. When it has the address and needs to find the name, it looks in the `hosts.byaddr` map.

An NIS Makefile is stored in the `/var/yp` directory of machines designated as an NIS server at installation time. Running `make` in that directory causes `makedbm` to create or modify the default NIS maps from the input files.

**Note** - Always create maps on the master server, as maps created on a slave will not automatically be pushed to the master server.
Default NIS Maps

A default set of NIS maps are provided in the Oracle Solaris system. You might want to use all these maps or only some of them. NIS can also use whatever maps you create or add when you install other software products.

Default maps for an NIS domain are located in each server’s `/var/yp/domain-name` directory. For example, the maps that belong to the domain `test.com` are located in each server’s `/var/yp/test.com` directory.

The following table describes the default NIS maps and lists the appropriate source file name for each map.

<table>
<thead>
<tr>
<th>Map Name</th>
<th>Corresponding Source File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audit_user</td>
<td>audit_user</td>
<td>Contains user auditing preselection data.</td>
</tr>
<tr>
<td>auth_attr</td>
<td>auth_attr</td>
<td>Contains authorization names and descriptions.</td>
</tr>
<tr>
<td>bootparams</td>
<td>bootparams</td>
<td>Contains path names of files that clients need during boot: root, swap, possibly others.</td>
</tr>
<tr>
<td>ethers.byaddr</td>
<td>ethers</td>
<td>Contains machine names and Ethernet addresses. The Ethernet address is the key in the map.</td>
</tr>
<tr>
<td>ethers.byname</td>
<td>ethers</td>
<td>Same as <code>ethers.byaddr</code>, except the key is machine name instead of the Ethernet address.</td>
</tr>
<tr>
<td>exec_attr</td>
<td>exec_attr</td>
<td>Contains profile execution attributes.</td>
</tr>
<tr>
<td>group.bygid</td>
<td>group</td>
<td>Contains group security information with group ID as key.</td>
</tr>
<tr>
<td>group.byname</td>
<td>group</td>
<td>Contains group security information with group name as key.</td>
</tr>
<tr>
<td>hosts.byaddr</td>
<td>hosts</td>
<td>Contains machine name, and IP address, with IP address as key.</td>
</tr>
<tr>
<td>hosts.byname</td>
<td>hosts</td>
<td>Contains machine name and IP address, with machine (host) name as key.</td>
</tr>
<tr>
<td>mail.aliases</td>
<td>aliases</td>
<td>Contains aliases and mail addresses, with aliases as key.</td>
</tr>
<tr>
<td>mail.byaddr</td>
<td>aliases</td>
<td>Contains mail address and alias, with mail address as key.</td>
</tr>
<tr>
<td>netgroup.byhost</td>
<td>netgroup</td>
<td>Contains group name, user name and machine name.</td>
</tr>
<tr>
<td>netgroup.byuser</td>
<td>netgroup</td>
<td>Same as <code>netgroup.byhost</code>, except that key is user name.</td>
</tr>
<tr>
<td>Map Name</td>
<td>Corresponding Source File</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>netgroup</td>
<td>netgroup</td>
<td>Same as netgroup.byhost, except that key is group name.</td>
</tr>
<tr>
<td>netid.byname</td>
<td>passwd, hosts, group</td>
<td>Used for UNIX-style authentication. Contains machine name and mail address (including domain name). If there is a netid file available it is consulted in addition to the data available through the other files.</td>
</tr>
<tr>
<td>publickey.byname</td>
<td>publickey</td>
<td>Contains the public key database used by secure RPC.</td>
</tr>
<tr>
<td>netmasks.byaddr</td>
<td>netmasks</td>
<td>Contains network mask to be used with IP submitting, with the address as key.</td>
</tr>
<tr>
<td>networks.byaddr</td>
<td>networks</td>
<td>Contains names of networks known to your system and their IP addresses, with the address as key.</td>
</tr>
<tr>
<td>networks.byname</td>
<td>networks</td>
<td>Same as networks.byaddr, except key is name of network.</td>
</tr>
<tr>
<td>passwd.adjunct.byname</td>
<td>passwd and shadow</td>
<td>Contains auditing information and the hidden password information for C2 clients.</td>
</tr>
<tr>
<td>passwd.byname</td>
<td>passwd and shadow</td>
<td>Contains password information with user name as key.</td>
</tr>
<tr>
<td>passwd.byuid</td>
<td>passwd and shadow</td>
<td>Same as passwd.byname, except that key is user ID.</td>
</tr>
<tr>
<td>prof.attr</td>
<td>prof_attr</td>
<td>Contains attributes for execution profiles.</td>
</tr>
<tr>
<td>protocols.byname</td>
<td>protocols</td>
<td>Contains network protocols known to your network.</td>
</tr>
<tr>
<td>protocols.bynumber</td>
<td>protocols</td>
<td>Same as protocols.byname, except that key is protocol number.</td>
</tr>
<tr>
<td>rpc.bynumber</td>
<td>rpc</td>
<td>Contains program number and name of RPCs known to your system. Key is RPC program number.</td>
</tr>
<tr>
<td>services.byname</td>
<td>services</td>
<td>Lists Internet services known to your network. Key is port or protocol.</td>
</tr>
<tr>
<td>services.byservice</td>
<td>services</td>
<td>Lists Internet services known to your network. Key is service name.</td>
</tr>
<tr>
<td>user_attr</td>
<td>user_attr</td>
<td>Contains extended attributes for users and roles.</td>
</tr>
<tr>
<td>ypservers</td>
<td>N/A</td>
<td>Lists NIS servers known to your network.</td>
</tr>
</tbody>
</table>

The ageing.byname mapping contains information that is used by the yppasswdd daemon to read and write password aging information to the directory information tree (DIT) when the NIS-to-LDAP transition is implemented. If password aging is not being used, then it can be commented out of the mapping file. For more information about the NIS-to-LDAP transition,
Using NIS Maps

NIS makes updating network databases much simpler than with the /etc files system. You no longer have to change the administrative /etc files on every machine each time you modify the network environment.

However, NIS provides no additional security than that provided by the /etc files. If additional security is needed, such as restricting access to the network databases, sending the results of searches over the network by using SSL, or using more advanced features such as Kerberos secured searches, then LDAP naming services should be used instead.

For example, when you add a new user to a network running NIS, you only have to update the input file in the master server and run the make command. This command automatically updates the passwd.byname and passwd.byuid maps. These maps are then transferred to the slave servers and are available to all of the domain’s client machines and their programs. When a client machine or application requests information by using the user name or UID, the NIS server refers to the passwd.byname or passwd.byuid map, as appropriate, and sends the requested information to the client.

You can use the ypcat command to display the values in a map. The ypcat basic format is the following.

```
% ypcat mapname
```

where mapname is the name of the map you want to examine or its nickname. If a map is composed only of keys, as in the case of ypservers, use ypcat -k. Otherwise, ypcat prints blank lines. The ypcat(1) man page describes more options for ypcat.

You can use the ypwhich command to determine which server is the master of a particular map. Type the following.

```
% ypwhich -m mapname
```

where mapname is the name or the nickname of the map whose master you want to find. ypwhich responds by displaying the name of the master server. For complete information, refer to the ypwhich(1) man page.

NIS Map Nicknames

Nicknames are aliases for full map names. To obtain a list of available map nicknames, such as passwd for passwd.byname, type ypcat -x or ypwhich -x.
Nicknames are stored in the /var/yp/nicknames file, which contains a map nickname followed by the fully specified name for the map, separated by a space. This list can be added to or modified. Currently, there is a limit of 500 nicknames.

**NIS Binding**

NIS clients are connected to an NIS server through the binding process. This process is supported by the svc:/network/nis/client and svc:/network/nis/domain services. These services must be enabled for any NIS service to operate. The svc:/network/nis/client service can work in one of two modes: server-list or broadcast.

- **Server-list** — In the server-list mode, the ypbind process queries the svc:/network/nis/domain service for the names of all NIS servers in the domain. The ypbind process binds only to servers in this file.

  NIS servers can be added by using the svc:/network/nis/domain service. They are added to the config/ypservers property in the svc:/network/nis/domain service. Each property value represents a specific NIS server.

  Additionally, any server name that is specified in the svc:/network/nis/domain service must contain an entry in the /etc/inet/hosts file for NIS binding to function.

- **Broadcast** — The ypbind process can also use an RPC broadcast to initiate a binding.

  Because broadcasts are only local subnet events that are not routed further, there must be at least one server (master or slave) on the same subnet as the client. The servers themselves might exist throughout different subnetworks because map propagation works across subnet boundaries. In a subnet environment, one common method is to make the subnet router an NIS server. This allows the domain server to serve clients on either subnet interface.

  Broadcast mode is generally the recommended mode of operation. Broadcast mode does not require additional host entries to be specified (or changes to be made to /etc/inet/hosts).

Normally, after a client is bound to a server, it stays bound to that server until something causes the binding to change. For example, if a server goes out of service, the clients it served will then bind to new servers.

To determine which NIS server is currently providing service to a specific client, use the following command.

```
% ypwhich machinename
```

where machinename is the name of the client. If no machine name is mentioned, the ypwhich command defaults to the local machine (that is, the machine on which the command is run).
Server-List Mode

The binding process in server-list mode works as follows:

1. Any program, running on the NIS client machine that needs information provided by an NIS map, asks ypbind for the name of a server.
2. The ypbind daemon looks in the /var/yp/binding/domainname/ypservers file for a list of NIS servers for the domain.
3. The ypbind daemon initiates binding to the first server in the list. If the server does not respond, ypbind tries the second, and so on, until it finds a server or exhausts the list.
4. The ypbind daemon tells the client process which server to talk to. The client then sends the request directly to the server.
5. The ypserv daemon on the NIS server handles the request by consulting the appropriate map.
6. The ypserv daemon sends the requested information back to the client.

Broadcast Mode

The broadcast mode binding process works as follows:

1. The ypbind daemon must be started with the broadcast option set (broadcast).
2. The ypbind daemon issues an RPC broadcast in search of an NIS server.

Note - In order to support such clients, it is necessary to have an NIS server on each subnet requiring NIS service.

3. The ypbind daemon initiates binding to the first server that responds to the broadcast.
4. The ypbind daemon tells the client process which server to talk to. The client then sends the request directly to the server.
5. The ypserv daemon on the NIS server handles the request by consulting the appropriate map.
6. The ypserv daemon sends the requested information back to the client.
Setting Up and Configuring Network Information Service

This chapter describes the initial set up and configuration of the Network Information Service (NIS).

Note - In some contexts, machine names are referred to as host names or machine names. This discussion uses “machine,” but some screen messages or NIS map names might use host or machine.

This chapter covers the following topics:

- “Configuring NIS Task Map” on page 63
- “Before You Begin Configuring NIS” on page 64
- “Planning Your NIS Domain” on page 66
- “Preparing the Master Server” on page 66
- “Starting and Stopping NIS Services on an NIS Server” on page 74
- “Setting Up NIS Slave Servers” on page 77
- “Administering NIS Clients” on page 82

Configuring NIS Task Map

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare source files for conversion.</td>
<td>You clean up local /etc files before building the NIS maps from them.</td>
<td>“How to Prepare Source Files for Conversion” on page 68</td>
</tr>
<tr>
<td>Set up the master server.</td>
<td>Creates a master server, which is the primary source of NIS information.</td>
<td>“How to Set Up the Master Server” on page 72</td>
</tr>
<tr>
<td>Start NIS on the master server.</td>
<td>Starts providing NIS information from an NIS server.</td>
<td>“Starting and Stopping NIS Services on an NIS Server” on page 74</td>
</tr>
</tbody>
</table>
Before You Begin Configuring NIS

Before configuring your NIS namespace, you must do the following.

- Plan your NIS domain. See “Planning Your NIS Domain” on page 66 for details.
- Install properly configured name service switch information on all the machines that will be using NIS. See Chapter 2, “About the Name Service Switch” for details.

NIS and the Service Management Facility

The NIS service is managed by the Service Management Facility. For an overview of SMF, refer to Chapter 1, “Introduction to the Service Management Facility,” in “Managing System Services in Oracle Solaris 11.2”. Also refer to the `svcadm(1M)` and `svcs(1)` man pages for more details.

The following list provides a short overview of some of the important information needed to use the SMF service to administer NIS.

- Administrative actions on this service, such as enabling, disabling, or restarting, can be performed by using the `svcadm` command. However, `ypstart` and `ypstop` can also be used from the command line to start or stop NIS. See the `ypstart(1M)` and `ypstop(1M)` man pages for more information.

**Tip** - Temporarily disabling a service by using the `-t` option provides some protection for the service configuration. If the service is disabled with the `-t` option, the original settings would be restored for the service after a reboot. If the service is disabled without `-t`, the service will remain disabled after reboot.

- The NIS Fault Manager Resource Identifiers (FMRIs) are:
  - `svc:/network/nis/server` for the NIS server
  - `svc:/network/nis/client` for the NIS client
- `svc:/network/nis/domain` for the domain name
- You can query the status of the NIS service by using the `svcs` command.
- The following are examples of the `svcs` command and its output:

```bash
$ svcs network/nis/server
STATE STIME FMRI
online Jan_10 svc:/network/nis/server:default

$ svcs \
is\*
STATE STIME FMRI
online Oct_09 svc:/network/nis/domain:default
online Oct_09 svc:/network/nis/client:default
```

- The following is an example of the `svcs -l` command and its output:

```bash
$ svcs -l /network/nis/client
fmri svc:/network/nis/client:default
name NIS (YP) client
enabled true
state online
next_state none
state_time Tue Aug 23 19:23:28 2011
logfile /var/svc/log/network-nis-client:default.log
restarter svc:/system/svc/restarter:default
contract_id 88
manifest /lib/svc/manifest/network/nis/client.xml
manifest /lib/svc/manifest/network/network-location.xml
manifest /lib/svc/manifest/system/name-service/upgrade.xml
dependency require_all/none svc:/system/filesystem/minimal (online)
dependency require_all/restart svc:/network/rpc/bind (online)
dependency require_all/restart svc:/network/nis/domain (online)
dependency optional_all/none svc:/network/nis/server (absent)
dependency optional_all/none svc:/network/location:default (online)
dependency optional_all/none svc:/system/name-service/upgrade (online)
dependency optional_all/none svc:/milestone/config (online)
dependency require_all/none svc:/milestone/unconfig (online)
```

- You can use the `svccfg` utility to obtain more detailed information about a service. See the `svccfg(1M)` man page.
- You can check a daemon's presence by using the `ps` command.

```bash
$ ps -ef |grep ypbind
daemon 100813   1   0 Aug 23 7   0:00 /usr/lib/netsvc/yp/ypbind -broadcast
```
Planning Your NIS Domain

Before you configure machines as NIS servers or clients, you must plan the NIS domain.

Decide which machines will be in your NIS domain. An NIS domain does not have to mirror your DNS domain. A DNS domain can have more than one NIS domain, and machines can exist in your DNS domain that are outside of your NIS domain.

An NIS domain name can be 256 characters long. A good practice is to limit domain names to no more than 32 characters. NIS domain names are case-sensitive. For convenience, you can choose to use your Internet domain name as the basis for your NIS domain name. Be aware that users might become confused if the NIS domain name includes capitals, but the DNS domain name does not. For example, if your Internet domain name is example.com, you can also name your NIS domain example.com. If you wanted to divide example.com into two NIS domains, for example, one for the sales department and the other for the manufacturing department, you could name one domain sales. example.com and the other domain manf. example.com.

Note - Merging and administering split NIS domains can be very difficult, so ensure that you have a good reason to split an NIS domain.

Before a machine can use NIS services, the correct NIS domain name and machine name must be set. A machine's name is set by the hostname command. The machine's domain name is set by the domainname command. The hostname and domainname commands can be used to display the machine name and the NIS domain name.

Identify Your NIS Servers and Clients

Select one machine to be the master server. Decide which machines will be slave servers.

Decide which machines will be NIS clients. Typically, all machines in your NIS domain are set to be NIS clients, although this is not necessary.

Preparing the Master Server

The following sections describe how to prepare the source files and the passwd files for the master server.

Preparing the Master Server (Task Map)

The following table lists the tasks for preparing the NIS master server.
Preparing the Master Server

Chapter 6 • Setting Up and Configuring Network Information Service

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare the source files for conversion.</td>
<td>Prepares the source files for conversion to NIS maps.</td>
<td>&quot;How to Prepare Source Files for Conversion&quot; on page 68</td>
</tr>
<tr>
<td>Install the NIS master server package.</td>
<td>Installs the NIS master server package.</td>
<td>&quot;How to Install the NIS Master Server Package&quot; on page 71</td>
</tr>
<tr>
<td>Set up the master server.</td>
<td>Configures the NIS master server and creates the NIS maps on the master server.</td>
<td>&quot;How to Set Up the Master Server&quot; on page 72</td>
</tr>
<tr>
<td>Support multiple NIS domains.</td>
<td>Sets up the NIS master server to support multiple domains.</td>
<td>&quot;How to Support Multiple NIS Domains on One Master Server&quot; on page 73</td>
</tr>
</tbody>
</table>

Source Files Directory

The source files are typically located in the /etc directory on the master server. However, leaving them in /etc is undesirable because the contents of the maps are then the same as the contents of the local files on the master server. This is a special problem for passwd and shadow files because all users have access to the master server maps and the root password would be passed to all NIS clients through the passwd map. See “passwd Files and Namespace Security” on page 67 for additional information.

However, if you put the source files in some other directory, you must modify the Makefile in /var/yp by changing the DIR=/etc line to DIR=/your-choice, where your-choice is the name of the directory you will be using to store the source files. This allows you to treat the local files on the server as if they were those of a client. (It is good practice to first save a copy of the original Makefile.)

In addition, the audit_user, auth_attr, exec_attr, and prof_attr NIS maps should be created from a directory other than the default. Amend /var/yp/Makefile by changing RBACDIR =/etc/security to RBACDIR=/your-choice.

passwd Files and Namespace Security

For security reasons, the files used to build the NIS password maps should not contain an entry for root, to prevent unauthorized root access. Therefore, the password maps should not be built from the files located in the master server's /etc directory. The password files used to build the
password maps should have the root entry removed from them and be located in a directory that can be protected from unauthorized access.

For example, the master server password input files should be stored in a directory such as /var/yp, or any directory of your choice, as long as the file itself is not a link to another file and its location is specified in the Makefile. The correct directory option is set automatically according to the configuration specified in your Makefile.

**Caution** - Be sure that the passwd file in the directory specified by PWDDIR does not contain an entry for root.

If your source files are in a directory other than /etc, you must alter the PWDIR password macro in /var/yp/Makefile to refer to the directory where the passwd and shadow files reside. You change the line PWDIR=/etc to PWDIR=/your-choice, where your-choice is the name of the directory you that will use to store the passwd map source files.

**How to Prepare Source Files for Conversion**

This procedure explains how to prepare the source files for conversion to NIS maps.

1. **Become an administrator.**
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Check the source files on the master server to make sure that they reflect your system.**
   Check the following files:
   - audit_user
   - auth_attr
   - auto.home or auto_home
   - auto.master or auto_master
   - bootparams
   - ethers
   - exec_attr
   - group
   - hosts
3. **Copy all of these source files, except for passwd and shadow, to the source directory that you have selected.**

   The source directory is defined in `/var/yp/Makefile` by the `DIR` macro.

4. **Copy the passwd and shadow files to the password source directory that you have selected.**

   The password source directory is defined in the Makefile by the `PWDIR` macro.

5. **Copy the audit_user, auth_attr, exec_attr, and prof_attr files to the rights source directory that you have selected.**

   The rights source directory is defined in `/var/yp/Makefile` by the `RBACDIR` macro. If desired, merge the contents of the files in the `/etc/security/auth_attr.d` directory into a copy of the `auth_attr` file before copying it. Similarly, combine the files in the `exec_attr.d` and `prof_attr.d` directories with `exec_attr` and `prof_attr`, if desired.

   **Caution** - Because these files will need to be remerged any time the system is upgraded, keep the local files separate from the release files in the `/etc/security/* .d` directories.

6. **Check the /etc/mail/aliases file.**

   Unlike other source files, the `/etc/mail/aliases` file cannot be moved to another directory. This file must reside in the `/etc/mail` directory. Refer to the `aliases(4)` man page for more information.
Note - You can add an NIS-specific mail aliases file by pointing the ALIASES = /etc/mail/aliases entry in /var/yp/Makefile to another location. When you then run the make command, the ALIASES entry creates a mail.aliases map. The sendmail service uses this map in addition to the /etc/mail/aliases file when the /etc/nsswitch.conf file properly targets nis in addition to files. Refer to “Modifying and Using /var/yp/Makefile” on page 92.

7. **Clean all comments and other extraneous lines and information from the source files.**

These operations can be done through a sed or awk script or with a text editor. /var/yp/Makefile performs some file cleaning automatically for you, but it is good practice to manually examine and clean these files before running the make command.

8. **Make sure that the data in all the source files is correctly formatted.**

Source file data must be in the correct format for that particular file. Check the man pages for the different files to make sure that each file is in the correct format.

### Preparing /var/yp/Makefile

After checking the source files and copying them into the source file directory, you now need to convert those source files into the ndbm format maps that the NIS service uses. This is done automatically for you by ypinit when called on the master server, as explained in “How to Set Up the Master Server” on page 72.

The ypinit script calls the make program, which uses /var/yp/Makefile. A default copy of the file is provided for you in the /var/yp directory and contains the commands needed to transform the source files into the desired ndbm format maps.

You can use the default Makefile as is, or modify it. If you do modify the default Makefile, be sure to first copy and store the original default Makefile in case you need it for future use. You might need to make one or more of the following modifications to the Makefile:

- **Nondefault maps**
  
  If you have created your own non-default source files and want to convert them to NIS maps, you must add those source files to the Makefile.

- **DIR value**
  
  If you want the Makefile to use source files stored in some directory other than /etc, as explained in “Source Files Directory” on page 67, you must change the value of DIR
in the `Makefile` to the directory that you want to use. When changing this value in the `Makefile`, do not indent the line.

- **PWDIR value**
  
  If you want the `Makefile` to use the `passwd`, `shadow`, and `adjunct` source files that are stored in some directory other than `/etc`, you must change the value of `PWDIR` in the `Makefile` to the directory that you want to use. When changing this value in the `Makefile`, do not indent the line.

- **RBACDIR value**

  If you want the `Makefile` to use the `audit_user`, `auth_attr`, `exec_attr`, and `prof_attr` source files that are stored in some directory other than `/etc`, you must change the value of `RBACDIR` in the `Makefile` to the directory that you want to use. When changing this value in the `Makefile`, do not indent the line.

- **Domain name resolver**

  If you want the NIS server to use the domain name resolver for machines not in the current domain, comment out the `Makefile` line `B=`, and uncomment (activate) the line `B=-b`.

The function of the `Makefile` is to create the appropriate NIS maps for each of the databases listed under `all`. After passing through `makedbm` the data is collected in two files, `mapname.dir` and `mapname.pag`. Both files are in the `/var/yp/domainname` directory on the master server.

The `Makefile` builds `passwd` maps from the `PWDIR/passwd`, `PWDIR/shadow`, and `PWDIR/security/passwd.adjunct` files, as appropriate.

### How to Install the NIS Master Server Package

Normally, the NIS master server package is installed when appropriate with the Oracle Solaris release. If the package was not included when the system was installed, use the following procedure to install the package.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Install the NIS master server package.**

   ```bash
   # pkg install pkg:/service/network/nis
   ```
How to Set Up the Master Server

The ypinit script sets up the master server and the slave servers and clients to use NIS. It also initially runs the make command to create the maps on the master server.

To use the ypinit command to build a fresh set of NIS maps on the master server, complete the following procedure.

1. **Become an administrator on the NIS master server.**
   For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Edit the /etc/inet/hosts file.**
   Add the host name and IP address of each NIS server. Use the following format: IPaddress FQDN-hostname aliases.
   For example:
   
   172.16.0.1 master.example.com master
   172.16.0.2 slave1.example.com slave1
   172.16.0.3 slave2.example.com slave2

3. **Build new maps on the master server.**

   `#/usr/sbin/ypinit -m`

4. **Type the names of the NIS servers.**
   When ypinit prompts for a list of other machines to become NIS slave servers, type the name of the server you are working on, along with the names of the NIS slave servers that you specified in the /etc/inet/hosts file.

5. **Verify that the NIS domain name is set.**

   `#/ domaimname`

   `example.com`

6. **Type y to select to stop the process if a nonfatal error occurs.**
   When ypinit asks whether you want the procedure to terminate at the first nonfatal error or continue despite nonfatal errors, type y. When you choose y, ypinit exits upon encountering the first problem. You can then fix it and restart ypinit. This is recommended if you are running ypinit for the first time. If you prefer to continue, you can try to manually fix all problems that occur, and then restart ypinit.
How to Support Multiple NIS Domains on One Master Server

Note - A nonfatal error can appear when some of the map files are not present. This is not an error that affects the functioning of NIS. You might need to add maps manually if they were not created automatically. Refer to “Default NIS Maps” on page 57 for a description of all default NIS maps.

7. Choose if the source files should be deleted.

The `ypinit` command asks whether the existing files in the `/var/yp/domain-name` directory can be destroyed. This message is displayed only if NIS has been previously installed. Normally, you would choose to delete the source files if you want to clean up the files from a previous installation.

8. After the `ypinit` command has constructed the list of servers, it invokes the `make` command.

This program uses the instructions contained in the `Makefile` (either the default file or the one you modified) located in `/var/yp`. The `make` command cleans any remaining comment lines from the files that you designated. It also runs `makedbm` on the files, creating the appropriate maps and establishing the name of the master server for each map.

If the map or maps being pushed by the `Makefile` correspond to a domain other than the one returned by the `domainname` command on the master, you can make sure that they are pushed to the correct domain by starting `make` in the `ypinit` shell script with a proper identification of the variable `DOM`, as follows:

```
# make DOM=domain-name passwd
```

This command pushes the `passwd` map to the intended domain, instead of the domain to which the master belongs.

9. If needed, make changes to the name service switch.

See “Configuring the Name Service Switch” on page 27.

How to Support Multiple NIS Domains on One Master Server

Normally, an NIS master server supports only one NIS domain. However, if you are using a master server to support multiple domains, you must slightly modify the steps, as described in “How to Set Up the Master Server” on page 72, when setting up the server to serve the additional domains.
1. **Become an administrator on the NIS master server.**
   For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Change the NIS domain name.**
   ```
   # domainname sales.example.com
   ```

3. **Build the NIS files.**
   ```
   # make DOM=sales.example.com
   ```

---

**Starting and Stopping NIS Services on an NIS Server**

Now that the master maps are created, you can start the NIS daemons on the master server and begin service. When you enable the NIS service, the `ypserv` and `ypbind` daemons start on the server. When a client requests information from the server, `ypserv` is the daemon that responds to information requests from clients after looking them up in the NIS maps. The `ypserv` and `ypbind` daemons are administered as a unit.

The following are the three ways that the NIS service can be started or stopped on a server:

- The SMF service automatically starts the NIS service during the boot process, if the NIS service was enabled previously.
- Using the `svcadm enable` and `svcadm disable` commands is the preferred manual method.
- The `ypstart` and `ypstop` commands, provide another manual method, although the `svcadm` command is preferred so that you can use SMF to administer the NIS service.

**Starting and Stopping NIS Services on an NIS Server (Task Map)**

The following table lists the tasks for starting and stopping the NIS services by using the `svcadm` command.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable the NIS server services manually.</td>
<td>Uses the <code>svcadm enable</code> command to enable the NIS server services.</td>
<td>“How to Enable the NIS Server Services Manually” on page 75</td>
</tr>
</tbody>
</table>
## How to Enable the NIS Server Services Manually

When you use the `svcadm` command, the instance name is required only if you are running more than one instance of the service. For more information, see “NIS and the Service Management Facility” on page 64, or the `svcadm(1M)` man page.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Start the required NIS server services.**

   ```bash
   # svcadm enable network/nis/domain
   # svcadm enable network/nis/server
   ```

   **Note** - The NIS service can also be enabled by using the `ypstart` command although the `svcadm` command is preferred.

---

### Task Description For Instructions

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable the NIS server services.</td>
<td>Uses the <code>svcadm disable</code> command to disable the NIS server services.</td>
<td>“How to Disable the NIS Server Services” on page 76</td>
</tr>
<tr>
<td>Refresh the NIS server service.</td>
<td>Uses the <code>svcadm refresh</code> command to refresh the NIS services.</td>
<td>“How to Refresh the NIS Server Service” on page 76</td>
</tr>
</tbody>
</table>

---

### Starting the NIS Service Automatically

When the `svc:/network/nis/server` service is enabled, then the `ypserv` daemon is automatically started up at boot. See “How to Set Up the Master Server” on page 72 for more information.

---

---
How to Disable the NIS Server Services

When you use the `svcadm` command, a specific instance name is required only if you are running more than one instance of the service. For more information, see “NIS and the Service Management Facility” on page 64, or the `svcadm(1M)` man page.

1. **Become an administrator.**
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Disable the required NIS server services.**

   ```bash
   # svcadm disable network/nis/domain
   # svcadm disable network/nis/server
   ```

   **Note** - The NIS service can also be disabled using the `ypstop` command.

How to Refresh the NIS Server Service

This procedure explains how to refresh the NIS server services after a configuration change has been made.

When you use the `svcadm` command, a specific instance name is required only if you are running more than one instance of the service. For more information, see “NIS and the Service Management Facility” on page 64, or the `svcadm(1M)` man page.

1. **Become an administrator.**
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Refresh the required NIS server services.**

   ```bash
   # svcadm refresh network/nis/domain
   # svcadm refresh network/nis/server
   ```
Setting Up NIS Slave Servers

Your network can have one or more slave servers. Having slave servers ensures the continuity of NIS services when the master server is not available.

Setting Up NIS Slave Servers (Task Map)

The following table lists the tasks for setting up NIS slave servers.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up a slave server.</td>
<td>Configures a system as an NIS slave server.</td>
<td>&quot;How to Set Up a Slave Server” on page 77</td>
</tr>
<tr>
<td>Start NIS on a slave server.</td>
<td>Uses the svcadm command to start the NIS client and server services.</td>
<td>&quot;How to Start NIS on a Slave Server&quot; on page 79</td>
</tr>
<tr>
<td>Add a new slave server.</td>
<td>Configures a new slave server after starting the NIS services.</td>
<td>&quot;How to Add a New Slave Server&quot; on page 80</td>
</tr>
</tbody>
</table>

Preparing a Slave Server

Before actually running the ypinit command to create the slave servers, first make sure that the svc:/network/nis/domain service has been configured.

**Note** - NIS domain names are case-sensitive, although DNS domain names are not.

Make sure that the network is working properly before you configure an NIS slave server. In particular, make sure that you can use the sshd command to send files from the master NIS server to NIS slaves.

**▼ How to Set Up a Slave Server**

The following procedure explains how to set up a slave server. Repeat this procedure for each machine you want configured as an NIS slave server.

1. **Become an administrator.**
2. **Edit the `/etc/inet/hosts` file.**

Add the name and IP address of each of the other NIS servers. Use the following format: 

```
IPaddress FQDN-hostname aliases
```

For example:

```
172.16.0.1 master.example.com master
172.16.0.2 slave1.example.com slave1
172.16.0.3 slave2.example.com slave2
```

3. **Change directory to `/var/yp` on the slave server.**

**Note** - You must first configure the new slave server as an NIS client so that it can obtain the NIS maps from the master server for the first time. See “Administering NIS Clients” on page 82 for details. After a NIS master map is changed, use the `yppush` command to propagate the new map to the NIS slave server. For information about propagating an NIS map, see “Updating and Modifying Existing Maps” on page 96.

4. **Initialize the slave server as an NIS client.**

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

The `ypinit` command prompts you for a list of NIS servers. Type the name of the local slave you are working on first, then type the name of the master server, followed by names of the other NIS slave servers in your domain. For the other slave servers, follow the order from the physically closest to the furthest in network terms.

5. **Determine if the client services are running, then start or restart the services as needed.**

```
#/svc -n
```

If the services are displayed with an `online` state, then NIS is running. If the service state is disabled, then NIS is not running.

- **If the client services are running, restart the client services.**

```
#/svcadm restart network/nis/domain
#/svcadm restart network/nis/client
```
How to Start NIS on a Slave Server

The following procedure explains how to start NIS on a slave server.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Restart the client service and start all NIS server processes.**

   - `# svcadm restart network/nis/domain`
   - `# svcadm restart network/nis/client`
   - `# svcadm enable network/nis/server`

---

b. If the client services are not running, start the client services.

   - `# svcadm enable network/nis/domain`
   - `# svcadm enable network/nis/client`

6. **Determine if the NIS master server is running, then start or restart the service as needed.**

   - `# svcs network/nis/server
     STATE           STIME       FMRI
     offline        20:32:56   svc:/network/nis/server:default`

   a. If the master NIS server is running, restart the service.

      - `# svcadm restart network/nis/server`

   b. If the master NIS server is not running, start the service.

      - `# svcadm enable network/nis/server`

7. **Initialize this machine as a slave server.**

   - `# /usr/sbin/ypinit -s master`

   where `master` is the machine name of the existing NIS master server.
How to Add a New Slave Server

After NIS is running, you might need to create an NIS slave server that you did not include in the initial list given to the `ypinit` command. Use this procedure to add a new NIS slave server.

1. **Become an administrator on the NIS master server.**
   For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Change to the NIS domain directory.**
   
   ```
   # cd /var/yp/domainname
   ```

3. **Disassemble the `ypservers` file.**
   
   ```
   # makedbm -u ypservers >/tmp/temp_file
   ```
   The `makedbm` command converts `ypservers` from `ndbm` format to a temporary ASCII file `/tmp/temp_file`.

4. **Edit the `/tmp/temp_file` file.**
   Add the name of the new slave server to the list of servers. Then, save and close the file.

5. **Run the `makedbm` command with `temp_file` as the input file and `ypservers` as the output file.**
   
   ```
   # makedbm /tmp/temp_file ypservers
   ```
   The `makedbm` command then converts `ypservers` back into `ndbm` format.

6. **Verify that the `ypservers` map is correct.**
   Because there is no ASCII file for `ypservers`, type the following on the slave server:
   
   ```
   slave3# makedbm -u ypservers
   ```
   The `makedbm` command displays each entry in `ypservers` on your screen.

   **Note** - If a machine name is not in `ypservers`, it will not receive updates to the map files because `yppush` consults this map for the list of slave servers.

7. **Become an administrator on the new NIS slave server.**
   For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.
8. **Verify that the NIS domain name is set.**
   
   ```
   # domainname
   example.com
   ```

9. **Set up the new slave server's NIS domain directory.**
   
   Copy the NIS map set from the master server, then start the NIS client. When running the `ypinit` command, follow the prompts and list the NIS servers in order of preference.
   
   ```
   slave3# cd /var/yp
   slave3# ypinit -c
   ```

10. **Initialize this machine as a slave.**
    
    ```
    slave3# /usr/sbin/ypinit -s ypmaster
    ```
    
    where `ypmaster` is the machine name of the existing NIS master server.

11. **Stop the machine running as an NIS client.**
    
    ```
    slave3# svcadm disable network/nis/client
    ```

12. **Determine if the client services are running, then start or restart the services as needed.**
    
    ```
    # svcs \*nis\*
    STATE    STIME     FMRI
    online   20:32:56  svc:/network/nis/domain:default
    online   20:32:56  svc:/network/nis/client:default
    ```
    
    If the services are displayed with an online state, then NIS is running. If the service state is disabled, then NIS is not running.

    **a. If the client services are running, restart the client services.**
    
    ```
    # svcadm restart network/nis/domain
    # svcadm restart network/nis/client
    ```

    **b. If the client services are not running, start the client services.**
    
    ```
    # svcadm enable network/nis/domain
    # svcadm enable network/nis/client
    ```

13. **Determine if the NIS server is running, then start or restart the service as needed.**
    
    ```
    # svcs network/nis/server
    STATE    STIME     FMRI
    offline   20:32:56  svc:/network/nis/server:default
    ```
    
    **a. If the NIS server is running, restart the service.**
slave3# svcadm restart network/nis/server

b. If the NIS server is not running, start the service.

slave3# svcadm enable network/nis/server

Administering NIS Clients

The two methods for configuring a client machine to use NIS as its naming service are explained in this section.

**Note** - The Oracle Solaris OS does not support a configuration in which an NIS client and a native LDAP client coexist on the same client machine.

- **Broadcast method** — The preferred method of configuring a client machine to use NIS. See “How to Configure an NIS Client in Broadcast Mode” on page 83 for instructions.
- **Server-list method** — Another method for configuring a client machine by using the `ypinit` command to specify the servers. See “How to Configure an NIS Client Using Specific NIS Servers” on page 83 for instructions.

**Administering NIS Clients (Task Map)**

The following table lists the tasks for administering NIS clients.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure an NIS client in broadcast mode.</td>
<td>Configures the NIS clients by searching for the an NIS server that exists on the local subnet.</td>
<td>“How to Configure an NIS Client in Broadcast Mode” on page 83</td>
</tr>
<tr>
<td>Configure an NIS client using specific NIS servers.</td>
<td>Configures an NIS client using specific NIS master and slave servers.</td>
<td>“How to Configure an NIS Client Using Specific NIS Servers” on page 83</td>
</tr>
<tr>
<td>Disable the NIS client service.</td>
<td>Uses the svcadm command to disable the NIS client service.</td>
<td>“Disabling the NIS Client Services” on page 84</td>
</tr>
</tbody>
</table>
How to Configure an NIS Client in Broadcast Mode

This is the preferred method for establishing an NIS client.

When you start the nis/client service, the service runs the ypbind command, which searches the local subnet for an NIS server. If a subnet is found, ypbind binds to it. This search is referred to as broadcasting. If no NIS server exists on the client's local subnet, ypbind fails to bind and the client machine cannot obtain namespace data from the NIS service. See “How to Configure an NIS Client Using Specific NIS Servers” on page 83 for instructions.

1. **Become an administrator.**
   
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Set the NIS domain name.**

   ```
   # domainname
   example.com
   ```

3. **If needed, make changes to the name service switch.**

   See “Configuring the Name Service Switch” on page 27.

4. **Start the NIS client services.**

   ```
   # svcadm enable network/nis/domain
   # svcadm enable network/nis/client
   ```

How to Configure an NIS Client Using Specific NIS Servers

**Before You Begin**

The following procedure requires that the hostnames that are entered in step 3 can be resolved by DNS. If you are not using DNS or you type in a hostname instead of an IP address, make sure to add an appropriate entry for each NIS server to the /etc/hosts file on the client. For more information, see the ypinit(1M) man page.

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

Chapter 6 • Setting Up and Configuring Network Information Service  83
2. **Set the NIS domain name.**

   ```shell
   # domainname
   example.com
   # svcadm enable network/nis/domain
   ```

3. **Run the client configuration script.**

   ```shell
   # ypinit -c
   ```

   You are prompted to name the NIS servers from which the client obtains naming service information. You can list the master server and as many slave servers as you want. The servers that you list can be located anywhere in the domain. It is a better practice to first list the servers closest (in network terms) to the machine, than those servers that are located on more distant parts of the network.

### Disabling the NIS Client Services

1. **Become an administrator.**

   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Disable the NIS client services.**

   ```shell
   # svcadm disable network/nis/domain
   # svcadm disable network/nis/client
   ```
Administering Network Information Service

This chapter describes how to administer NIS. The following topics are covered:

- “Password Files and Namespace Security” on page 85
- “Administering NIS Users” on page 86
- “Working With NIS Maps” on page 89
- “Updating and Modifying Existing Maps” on page 96
- “Working With NIS Servers” on page 101

**Note** - The NIS service is managed by the Service Management Facility. Administrative actions on this service, such as enabling, disabling, or restarting, can be performed by using the `svcadm` command. See “NIS and the Service Management Facility” on page 64 for more information about using SMF with NIS. For an overview of SMF, refer to Chapter 1, “Introduction to the Service Management Facility,” in “Managing System Services in Oracle Solaris 11.2”. Also refer to the `svcadm(1M)` and `svcs(1)` man pages for more details.

NIS services can also be started and stopped by using the `ypstart` and `ypstop` commands. See the `ypstart(1M)` and `ypstop(1M)` man pages for more information.

**Password Files and Namespace Security**

For security reasons, follow these guidelines.

- It is best to limit access to the NIS maps on the master server.
- The files used to build the NIS password maps should not contain an entry for `root` to protect against unauthorized access. To accomplish this, the password files used to build the password maps should have the `root` entry removed from them and be located in a directory other than the master server’s `/etc` directory. This directory should be secured against unauthorized access.

For example, the master server password input files could be stored in a directory such as `/var/yp`, or any directory of your choice, as long as the file itself is not a link to another file and is
specified in the Makefile. When you use either the Service Management Facility or the ypstart script to start the NIS service, the correct directory option is set according to the configuration specified in your Makefile.

**Note** - In addition to the older Solaris 1 version passwd file format, this implementation of NIS accepts the Solaris 2 passwd and shadow file formats as input for building the NIS password maps.

## Administering NIS Users

This section includes information about setting user passwords, adding new users to an NIS domain, and assigning users to netgroups.

### How to Add a New NIS User to an NIS Domain

1. **Become an administrator on the NIS master server.**
   
   For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Create the new user’s login ID with the useradd command.**
   
   ```
   # useradd userID
   ```
   
   where `userID` is the login ID of the new user. This command creates entries in the `/etc/passwd` and `/etc/shadow` files on the master NIS server.

3. **Create the new user’s initial password.**
   
   To create an initial password that the new user can use to log in, run the passwd command.

   ```
   # passwd userID
   ```
   
   where `userID` is the login ID of the new user. You will be prompted for the password to assign to this user.

   This step is necessary because the password entry created by the `useradd` command is locked, which means that the new user cannot log in. By specifying an initial password, you unlock the entry.

4. **Copy the new entry into the master server’s passwd map input files.**
How to Add a New NIS User to an NIS Domain

The map source files on your master server should be in a directory other than /etc. Copy and paste the new lines from the /etc/passwd and /etc/shadow files into the passwd map input files on the server. See “Password Files and Namespace Security” on page 85 for additional information.

For example, if you added the new user brown, the line from /etc/passwd that you would copy to your passwd input file would look like the following.

```
brown:x:123:10:User brown:/home/brown:/bin/csh:
```

The line for brown that you would copy from /etc/shadow would look like:

```
brown:$5$AfiPpYmXb26jKjkG/gKdfKXt/bem0RnbeH.qsvO9MwB03ulTihq9B:6445:6:6::
```

5. **Make sure that the Makefile correctly specifies the directory where the password input file resides.**

6. **Delete the new user’s entries from the /etc/passwd and /etc/shadow input files.**

For security reasons, do not keep user entries in the NIS master server /etc/passwd and /etc/shadow files. After copying the entries for the new user to the NIS map source files that are stored in some other directory, use the userdel command on the master server to delete the new user.

For example, to delete the new user brown from the master server's /etc files, you would enter the following.

```
# userdel brown
```

For more information about userdel, see the `userdel(1M)` man page.

7. **Update the NIS passwd maps.**

After you have updated the passwd input file on the master server, update the passwd maps by running `make` in the directory containing the source file.

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

8. **Tell the new user the initial password you have assigned to his or her login ID.**

After logging in, the new user can run passwd at any time to establish a different password.

**Setting User Passwords**

Users run passwd to change their passwords.
Before users can change their passwords, you must start the `rpc.yppasswd` daemon on the master server to update the password file.

The `rpc.yppasswd` daemon starts automatically on the master server. Notice that when the `-m` option is given to `rpc.yppasswd`, the make command is run in `/var/yp` immediately following a modification of the file. If you want to avoid having the `make` command run each time the `passwd` file is changed, remove the `-m` option from the `rpc.yppasswd` command in the `ypstart` script and control the pushing of the `passwd` maps through the `crontab` file.

**NIS Netgroups**

NIS netgroups are groups (sets) of users or machines that you define for your administrative purposes. For example, you can create netgroups that do the following.

- Define a set of users who can access a specific machine
- Define a set of NFS client machines to be given some specific file system access
- Define a set of users who are to have administrator privileges on all the machines in a particular NIS domain

Each netgroup is given a netgroup name. Netgroups do not directly set permissions or access rights. Instead, the netgroup names are used by other NIS maps in places where a user name or machine name would normally be used. For example, suppose you created a netgroup of network administrators called `netadmins`. To grant all members of the `netadmins` netgroup access to a given machine, you only need to add a `netadmin` entry to that machine's `/etc/passwd` file. Netgroup names can also be added to the `/etc/netgroup` file and propagated to the NIS netgroup map. See the `netgroup(4)` man page for more detailed information on using netgroups.

On a network using NIS, the `netgroup` input file on the master NIS server is used for generating three maps: `netgroup`, `netgroup.byuser`, and `netgroup.byhost`. The `netgroup` map contains the basic information in the `netgroup` input file. The two other NIS maps contain information in a format that speeds lookups of netgroup information, given the machine or user name.

Entries in the `netgroup` input file are in the format: `name ID`, where `name` is the name you give to a netgroup, and `ID` identifies a machine or user who belongs to the netgroup. You can specify as many IDs (members) to a netgroup as you want, separated by commas. For example, to create a netgroup with three members, the `netgroup` input file entry would be in the format: `name ID, ID, ID`. The member IDs in a `netgroup` input file entry are in the following format.

```
(machine), (user), (domain)
```

Where `machine` is a machine name, `user` is a user ID, and `domain` is the machine or user's NIS domain. The `domain` element is optional and should only be used to identify machines or users.
in some other NIS domain. The *machine* and *user* element of each member's entry are required, but a dash (-) is used to denote a null. There is no necessary relationship between the machine and user elements in an entry.

The following are two sample netgroup input file entries, each of which create a netgroup named `admins` composed of the users `hauri` and `juanita` who is in the remote domain `sales` and the machines `altair` and `sirius`.

```
admins (altair,hauri,) (sirius,juanita,sales)
admins (altair,-,) (sirius,-,) (-,hauri,) (-,juanita,sales)
```

Various programs use the netgroup NIS maps for permission checking during login, remote mount, remote login, and remote shell creation. These programs include `mountd` and `login`. The `login` command consults the netgroup maps for user classifications if it encounters netgroup names in the `passwd` database. The `mountd` daemon consults the netgroup maps for machine classifications if it encounters netgroup names in the `/etc/dfs/dfstab` file. In fact, any program that uses the `ruserok` interface checks the netgroup maps for both machine and user classifications if they encounter netgroup names in the `/etc/hosts.equiv` or `.rhosts` file.

If you add a new NIS user or machine to your network, be sure to add them to appropriate netgroups in the netgroup input file. Then use the `make` and `yppush` commands to create the netgroup maps and push them to all of your NIS servers. See the `netgroup(4)` man page for detailed information on using netgroups and netgroup input file syntax.

---

**Working With NIS Maps**

This section contains the following information:

- “Obtaining Map Information” on page 89
- “Changing a Map’s Master Server” on page 90
- “Modifying Configuration Files” on page 92
- “Modifying and Using `/var/yp/Makefile`” on page 92

### Obtaining Map Information

Users can obtain information from and about the maps at any time by using the `ypcat`, `ypwhich`, and `ypmatch` commands. In the examples that follow, `mapname` refers both to the official name of a map and to its nickname, if any.

To list all the values in a map, type the following:

```
% ypcat mapname
```
To list both the keys and the values (if any) in a map, type the following:

```
% ypcat -k mapname
```

To list all the map nicknames, type any of the following commands:

```
% ypcat -x
% ypmatch -x
% ypwhich -x
```

To list all the available maps and their masters, type the following:

```
% ypwhich -m
```

To list the master server for a particular map, type the following:

```
% ypwhich -m mapname
```

To match a key with an entry in a map, type the following:

```
% ypmatch key mapname
```

If the item you are looking for is not a key in a map, type the following:

```
% ypcat mapname | grep item
```

where `item` is the information for which you are searching. To obtain information about other domains, use the `-d domainname` option of these commands.

If the machine requesting information for a domain other than its default does not have a binding for the requested domain, `ypbind` consults the `/var/yp/binding/domainname/ypservers` file for a list of servers for that domain. If this file does not exist it issues an RPC broadcast for a server. In this case, there must be a server for the requested domain on the same subnet as the requesting machine.

# Changing a Map's Master Server

To change the master server for a selected map, you first have to build the map on the new NIS master. Since the old master server name occurs as a key-value pair in the existing map (this pair is inserted automatically by `makedbm`), copying the map to the new master or transferring a copy to the new master with `ypxfr` is insufficient. You have to reassociate the key with the new master server name. If the map has an ASCII source file, you should copy this file to the new master.

## How to Change a Map's Master Server

1. Become an administrator on the NIS master server.
For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Change directories.**

   ```sh
   newmaster# cd /var/yp
   ```

3. **The /var/yp/Makefile must have an entry for the new map before you specify the map to make.**

   If this is not the case, edit the Makefile now. For this example, add an entry for a map called `sites.byname`.

4. **To update or remake the map, type the following:**

   ```sh
   newmaster# make sites.byname
   ```

5. **If the old master remains an NIS server, remote log in (ssh) to the old master and edit /var/yp/Makefile.**

   Make sure that you comment out the section of the Makefile that made the `sites.byname` map so that it is no longer made there.

6. **If `sites.byname` only exists as an ndbm file, remake it on the new master server.**

   First, disassemble a copy of the `sites.byname` file by using the `ypcat` command. Then, run the disassembled version through `makedbm`.

   ```sh
   newmaster# cd /var/yp
   newmaster# ypcat sites.byname | makedbm domain/sites.byname
   ```

   After making the map on the new master, you must send a copy of the new map to the other slave servers. Do not use `yppush`, because the other slaves will try to get new copies from the old master, rather than the new one. A typical method for circumventing this is to transfer a copy of the map from the new master back to the old master. To do this, become superuser, or assume an equivalent role, on the old master server and type the following.

   ```sh
   oldmaster# /usr/lib/netsvc/yp/ypxfr -h newmaster sites.byname
   ```

   Now it is safe to run `yppush`. Any remaining slave servers still believe that the old master is the current master and will attempt to get the current version of the map from the old master. When clients do so, they will get the new map, which names the new master as the current master.

   If this method fails, you can log in as root on each NIS server and execute the `ypxfr` command as shown.
Modifying Configuration Files

NIS intelligently parses the setup files. Although this makes NIS administration easier, it does make the behavior of NIS more sensitive to changes in the setup and configuration files.

Use the procedures in this section when doing any of the following:

- /var/yp/Makefile to add or delete supported maps
- Adding or deleting $PWDIR/security/passwd.adjunct to allow or deny C2 security ($PWDIR is defined in /var/yp/Makefile)

How to Modify Configuration Files

Keep the following in mind.

- Deleting a map or source file from an NIS master server does not automatically result in corresponding deletions from slave servers. You must delete maps and source files from slave servers by hand.
- New maps do not automatically get pushed to existing slave servers. You must run ypxfr from the slaves.

1. Become an administrator.
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. Stop the NIS server.
   
   # svcadm disable network/nis/server

3. Make the necessary changes to your files.

4. Start the NIS server.
   
   # svcadm enable network/nis/server

Modifying and Using /var/yp/Makefile

You can modify the Makefile provided by default in /var/yp to suit your needs. You can add or delete maps, and you can change the names of some of the directories.
**Working With the Makefile**

To add a new NIS map, you must get copies of the *ndbm* files for the map into the `/var/yp/domainname` directory on each of the NIS servers in the domain. This is normally done for you by the Makefile. After deciding which NIS server is the master of the map, modify the Makefile on the master server so that you can conveniently rebuild the map. Different servers can be masters of different maps, but in most cases this leads to administrative confusion. Try to set only one server as the master of all maps.

Typically a human-readable text file is filtered through `awk`, `sed`, or `grep` to make it suitable for input to `makedbm`. Refer to the default Makefile for examples. See the `make(1S)` for general information about the `make` command.

Use the mechanisms already in place in the Makefile when deciding how to create dependencies that `make` will recognize. Be aware that `make` is very sensitive to the presence or absence of tabs at the beginning of lines within the dependency rules. A missing tab can invalidate an entry that is otherwise well formed.

Adding an entry to the Makefile involves the following.

- Adding the name of the database to the `all` rule
- Writing the `time` rule
- Adding the rule for the database

For example, in order for the Makefile to work on automounter input files, you would have to add the `auto_direct.time` and `auto_home.time` maps to the NIS database.

To add these maps to the NIS database you need to modify the Makefile.

**Changing Makefile Macros/Variables**

You can change the settings of the variables defined at the top of the Makefile by changing the value to the right of the equal sign (=). For instance, if you do not want to use the files located in `/etc` as input for the maps, but you would rather use files located in another directory, such as `/var/etc/domainname`, you should change `DIR` from `DIR=/etc` to `DIR=/var/etc/domainname`. You should also change `PWDIR` from `PWDIR=/etc` to `PWDIR=/var/etc/domainname`.

The variables are the following:

---

**Tip** - Keep an unmodified copy of the original Makefile for future reference.
**DIR**—The directory containing all of the NIS input files except passwd and shadow. The default value is /etc. Since it is not good practice to use the files in the master server's /etc directory as NIS input files, you should change this value.

**PWDIR**—The directory containing the passwd and shadow NIS input files. Since it is not good practice to use the files in the master server's /etc directory as NIS input files, you should change this value.

**DOM**—The NIS domain name. The default value of DOM can be set by using the domainname command.

---

**Modifying Makefile Entries**

The following procedure describes how to add and delete databases from the Makefile.

**How to Modify /var/yp/Makefile to Use Specific Databases**

This procedure requires that you have already configured an NIS master server.

1. **Become an administrator.**
   
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Modify the line that starts with the word all by adding the names of the database you want to add:**

   ```
   all: passwd group hosts ethers networks rpc services protocols 
   netgroup bootparams aliases netid netmasks 
   audit_user auth_attr exec_attr prof_attr 
   auto_direct
   ```

   The order of the entries is not relevant, but the blank space at the beginning of the continuation lines must be a Tab, not spaces.

3. **Add the following lines at the end of the Makefile:**

   ```
   auto_direct: auto_direct.time
   auto_home: auto_home.time
   ```

4. **Add an entry for auto_direct.time in the middle of the file.**

   ```
   auto_direct.time: $(DIR)/auto_direct
   @while read L; do echo $$L; done < $(DIR)/auto_direct
   $(CHKPIPE)) | \ (sed -e "/^#/d" -e "/^ */d" -e "/^ */s/$$/" -e "/^ */s/$$/" -e "/^ */s/$$/$$$$/" -e "/^ */s/$$/" -e "/^ */s/$$/" -e "/^ */s/$$/"
   $(CHKPIPE)) | $(MAKE) - $(YPDBDIR)/$(DOM)/auto_direct;
   ```
@touch auto_direct.time;
@echo "updated auto_direct";
@if [ ! $(NOPUSH) ]; then $(YPPUSH) auto_direct; fi
@if [ ! $(NOPUSH) ]; then echo "pushed auto_direct"; fi

where

- **CHKPIPE** makes certain that the operations to the left of the pipe (|) are successfully completed before piping the results to next commands. If the operations to the left of the pipe do not successfully complete, the process is terminated with a NIS `make terminated` message.
- **NOPUSH** prevents the makefile from calling `yppush` to transfer the new map to the slave servers. If NOPUSH is not set, the push is done automatically.

The **while** loop at the beginning is designed to eliminate any backslash-extended lines in the input file. The **sed** script eliminates comment and empty lines.

Follow the same procedure for all other automounter maps, such as `auto_home` or any other non-default maps.

5. **Run the `make` command.**

   `make mapname`

   where `mapname` is the name of the map you want to make.

### How to Modify the Makefile to Delete Databases

If you do not want the Makefile to produce maps for a specific database, edit the Makefile as follows.

1. **Delete the name of the database from the `all` rule.**

2. **Delete or comment out the database rule for the database you want to delete.**

   For example, to delete the `hosts` database, the `hosts.time` entry should be removed.

3. **Remove the time rule.**

   For example, to delete the `hosts` database, the `hosts: hosts.time` entry should be removed.

4. **Remove the map from the master and slave servers.**
Updating and Modifying Existing Maps

After you have installed NIS, you might discover that some maps require frequent updating while others never need to change. For example, the passwd.byname map can change frequently on a large company's network, while the auto_master map changes little, if at all.

As mentioned in “Default NIS Maps” on page 57, the default location of the default NIS maps is on the master server in /var/yp/domainname, where domainname is the name of the NIS domain. When you need to update a map, you can use one of two updating procedures, depending on whether or not it is a default map.

- A default map is a map in the default set that is created by the ypinit command from the network databases.
- Non-default maps can be any of the following:
  - Maps that are included with an application purchased from a vendor
  - Maps that are created specifically for your site
  - Maps that are created from a non-text file

The following sections explain how to use various updating tools. In practice, you might decide to only use them if you add non-default maps or change the set of NIS servers after the system is already up and running.

▶ How to Update Maps Supplied With the Default Set

Use the following procedure to update maps that are supplied with the default set.

1. **Become an administrator on the NIS master server.**
   For more information, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2 ”.

2. **Edit the source file for the map that you want to change.**
   The file could reside in /etc or in some other directory of your choice.

3. **Run the make command.**

   ```
   # cd /var/yp
   # make mapname
   ```

   The make command then updates your map according to the changes you made in its corresponding file. It also propagates the changes among the other servers.
Maintaining Updated Maps

The following sections describe additional procedures after you have completed updating maps that are supplied with the default set.

Propagating an NIS Map

After a map is changed, the Makefile uses yppush to propagate a new map to the slave servers (unless NOPUSH is set in the Makefile). It does this by informing the ypserv daemon and sending a map transfer request. The ypserv daemon on the slave then starts a ypxfr process, which in turn contacts the ypxfrd daemon on the master server. Some basic checks are made (for example did the map really change?) and then the map is transferred. ypxfr on the slave then sends a response to the yppush process indicating whether the transfer succeeded.

The config/local_only property of the svc:/network/rpc/bind service must be set to false. Otherwise, the NIS master cannot transfer the updated version of NIS master map to the NIS slave server by using the yppush command.

Note - The above procedure will not work for newly created maps that do not yet exist on the slave servers. New maps must be sent to the slave servers by running ypxfr on the slaves.

Occasionally, maps fail to propagate and you must to use ypxfr manually to send new map information. You can choose to use ypxfr in two different ways: periodically through the root crontab file, or interactively on the command line. These approaches are discussed in the following sections.

Using the crontab Command for Map Transfers

Maps have different rates of change. For instance, some maps might not change for months at a time, such as protocols.byname among the default maps and auto_master among the non-default maps. However passwd.byname can change several times a day. Scheduling map transfer by using the crontab command enables you to set specific propagation times for individual maps.

To periodically run ypxfr at a rate appropriate for the map, the root crontab file on each slave server should contain the appropriate ypxfr entries. ypxfr contacts the master server and transfers the map only if the copy on the master server is more recent than the local copy.
Note - If your master server runs `rpc.yp passwd` with the default `-m` option, then each time someone changes their yp password, the passwd daemon runs `make`, which rebuilds the passwd maps.

Using Shell Scripts With `cron` and `ypxfr`

As an alternative to creating separate crontab entries for each map, you might prefer to have the root `crontab` command run a shell script that periodically updates all maps. Sample map-updating shell scripts are in the `/usr/lib/netsvc/yp` directory. The script names are `ypxfr_1perday`, `ypxfr_1perhour`, and `ypxfr_2perday`. You can modify or replace these shell scripts to accommodate your site requirements. The following example shows the default `ypxfr_1perday` shell script.

**EXAMPLE 7-1  ypxfr_1perday Shell Script**

```bash
#!/bin/sh
#
# ypxfr_1perday.sh - Do daily yp map check/updates
PATH=/bin:/usr/bin:/usr/lib/netsvc/yp:$PATH
export PATH
# set -xv
ypxfr group.byname
ypxfr group.bygid
ypxfr protocols.byname
ypxfr protocols.bynumber
ypxfr networks.byname
ypxfr networks.byaddr
ypxfr services.byname
ypxfr ypservers
```

This shell script updates the maps once per day, if the root `crontab` is executed daily. You can also have scripts that update maps once a week, once a month, once every hour, and so forth. However, be aware of the performance degradation that is implied in frequently propagating the maps. For more information, see the `crontab(1)` man page.

Run the same shell scripts as root on each slave server configured for the NIS domain. Alter the exact time of execution from one server to another to avoid bogging down the master.

If you want to transfer the map from a particular slave server, use the `-h machine` option of `ypxfr` within the shell script. Here is the syntax of the commands you put in the script.

```bash
# /usr/lib/netsvc/yp/ypxfr -h machine [ -c ] mapname
```

Where `machine` is the name of the server with the maps you want to transfer, and `mapname` is the name of the requested map. If you use the `-h` option without specifying a machine, `ypxfr`
tries to get the map from the master server. If ypserver is not running locally at the time ypxf r is executed, you must use the -c flag so that ypxf r does not send a clear current map request to the local ypserver.

You can use the -s domain option to transfer maps from another domain to your local domain. These maps must be the same across domains. For example, two domains might share the same services.byn ame and services.byadd r maps. Alternatively, for more control you can use rcp or rsync to transfer files across domains.

**Directly Invoking the ypxf r Command**

The second method of invoking the ypxf r command is to run it as a command. Typically, you do this only in exceptional situations – for example, when setting up a temporary NIS server to create a test environment or when trying to quickly get an NIS server that has been out of service consistent with the other servers.

**Logging ypxf r Activity**

The transfer attempts and results of ypxf r can be captured in a log file. If a file called /var/yp/ypxf r.log exists, results are appended to it. No attempt to limit the size of the log file is made. To prevent it from growing indefinitely, empty it from time to time by typing the following.

```
# cd /var/yp
# cp ypxf r.log ypxf r.log.old
# cat /dev/null > /var/yp/ypxf r.log
```

You can have crontab execute these commands once a week. To turn off logging, remove the log file.

**Modifying Non-Default Maps**

To update a non-default map, you must do the following:

1. Create or edit its corresponding text file.
2. Build (or rebuild) the new or updated map. There are two ways to build a map.
   - Use the Makefile. Using the Makefile is the preferred method of building a non-default map. If the map has an entry in the Makefile, run `make name` where `name` is the name of map you want to build. If the map does not have a Makefile entry, try to create one following the instructions in “Modifying and Using /var/yp/Makefile” on page 92.
   - Use the `/usr/sbin/makedbm` program. The `makedbm(1M)` man page fully describes this command.
Using the `makedbm` Command to Modify a Non-Default Map

There are two different methods for using `makedbm` to modify maps if you do not have an input file:

- Redirect the `makedbm -u` output to a temporary file, modify the file, then use the modified file as input to `makedbm`.
- Have the output of `makedbm -u` operated on within a pipeline that feeds into `makedbm`. This is appropriate if you can update the disassembled map with either `awk`, `sed`, or a `cat append`.

Creating New Maps From Text Files

Assume that a text file `/var/yp/mymap.asc` was created with an editor or a shell script on the master. You want to create an NIS map from this file and locate it in the `home-domain` subdirectory. To do this, type the following on the master server.

```bash
# cd /var/yp
# makedbm mymap.asc home-domain/mymap
```

The `mymap` map now exists on the master server in the directory `home-domain`. To distribute the new map to slave servers run `ypxfr`.

Adding Entries to a File-Based Map

Adding entries to `mymap` is simple. First, you must modify the text file `/var/yp/mymap.asc`. If you modify the actual dbm files without modifying the corresponding text file, the modifications are lost. Then run `makedbm` as shown above.

Creating Maps From Standard Input

When no original text file exists, create the NIS map from the keyboard by typing input to `makedbm`, as shown below (end with Control-D).

```bash
ypmaster# cd /var/yp
```
Modifying Maps Made From Standard Input

If you later need to modify the map, you can use `makedbm` to disassemble the map and create a temporary text intermediate file. To disassemble the map and create a temporary file, type the following:

```
% cd /var/yp
% makedbm -u homedomain/mymap > mymap.temp
```

The resulting temporary file `mymap.temp` has one entry per line. You can edit this file as needed, using any text editor.

To update the map, give the name of the modified temporary file to `makedbm` by typing the following:

```
% makedbm mymap.temp homedomain/mymap
% rm mymap.temp
```

Then propagate the map to the slave servers, by becoming `root` and typing the following.

```
# yppush mymap
```

The preceding paragraphs explained how to use `makedbm` to create maps. However, almost everything you actually have to do can be done by the `ypinit` command and by using `/var/yp/Makefile` unless you add non-default maps to the database or change the set of NIS servers after the system is already up and running.

Whether you use the `Makefile` in `/var/yp` or some other procedure the goal is the same. A new pair of well-formed dbm files must end up in the maps directory on the master server.

Working With NIS Servers

The following procedures show ways to modify the NIS configuration by binding to a specific NIS server, setting the NIS domain name, forwarding host lookups to DNS, and by turning off the NIS services.

Binding to a Specific NIS Server

Use the following steps to bind to an NIS server that you specify. For more information, see the `ypinit(1M), ypstart(1M), and svcadm(1M)` man pages.
1. Add the host name of the NIS server and its IP address to the /etc/hosts file.
2. Verify that the NIS domain name is set.
   
   ```sh
   # domainname
e.example.com
   ```
3. Prompt for the NIS server host name.
   
   ```sh
   # /usr/sbin/ypinit -c
   Server name: Type the NIS server host name
   ```
4. Restart the NIS services by performing one of the following steps:
   - For the services to persist across reboots, run the `svcadm` command.
     
     ```sh
     # svcadm enable svc:/network/nis/client
     ```
   - For the services to persist until reboot only, run the `ypstop` and `ypstart` commands.
     
     ```sh
     # /usr/lib/netsvc/yp/ypstop
     # /usr/lib/netsvc/yp/ypstart
     ```

### How to Set a Machine's NIS Domain Name

To change the NIS domain name of a machine, use the following procedure.

1. **Become an administrator.**
   
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. **Define the NIS domain name.**
   
   ```sh
   # domainname research.example.com
   ```

3. **Update and run the domain name services.**
   
   ```sh
   # svccfg -s nis/domain:default refresh
   # svcadm enable nis/domain
   ```

4. **Set up the machine as an NIS client, a slave server, or a master server.**
   
How to Configure Machine Host Name and Address Lookup Through NIS and DNS

Typically, NIS clients are configured with the nsswitch.conf file to use only NIS for machine name and address lookups. If this type of lookup fails, an NIS server can forward these lookups to DNS.

1. Become an administrator.
   For more information about obtaining the appropriate rights to perform specific tasks, see “Using Your Assigned Administrative Rights” in “Securing Users and Processes in Oracle Solaris 11.2”.

2. Add the YP_INTERDOMAIN key.
   The two map files, hosts.byname and hosts.byaddr must include the YP_INTERDOMAIN key. To test this key, edit /var/yp/Makefile and modify the following lines.

   ```
   #B=-b
   B=

   to

   B=-b
   #B=
   ```

   makedbm will now start with the -b flag when it makes the maps, and the YP_INTERDOMAIN key will be inserted into the ndbm files.

3. Run the make command to rebuild maps.
   ```
   # make hosts
   ```

4. Check that DNS name servers are set properly.
   The following command lists all of the IP addresses for the DNS name servers:

   ```
   # svcprop -p config/nameserver network/dns/client
   ```

5. To enable DNS forwarding, restart each server.
   ```
   # svcadm restart network/nis/server:instance
   ```

   In this implementation of NIS, the ypserv daemon automatically starts with the -d option to forward requests to DNS.
Turning Off NIS Services

If the `ypserv` daemon on the NIS master is disabled, you can no longer update any of the NIS maps.

- To disable NIS on a client, type the following:
  
  ```
  # svcadm disable network/nis/domain
  # svcadm disable network/nis/client
  ```

- To disable NIS on a specific slave or master server, type the following on the server:

  ```
  # svcadm disable network/nis/domain
  # svcadm disable network/nis/server
  ```
This chapter explains how to resolve problems encountered on networks running NIS. It covers problems that are encountered on both NIS clients and NIS servers.

Before trying to debug an NIS server or client, review Chapter 5, “About the Network Information Service” which explains the NIS environment. Then, look for the subheading in this section that best describes your problem.

Note - The NIS service is managed by the Service Management Facility. Administrative actions on this service, such as enabling, disabling, or restarting, can be performed by using the svcadm command. See “NIS and the Service Management Facility” on page 64 for more information about using SMF with NIS. For an overview of SMF, refer to Chapter 1, “Introduction to the Service Management Facility,” in “Managing System Services in Oracle Solaris 11.2”. Also refer to the svcadm(1M) and svcs(1) man pages for more details.

NIS services can also be started and stopped by using the ypstart and ypstop commands. See the ypstart(1M) and ypstop(1M) man pages for more information.

NIS Binding Problems

Symptoms of NIS Binding Problems

Common symptoms of NIS binding problems include the following.

- Messages saying that ypbind can't find or communicate with a server
- Messages saying that server not responding
- Messages saying that NIS is unavailable
- Commands on a client limp along in background mode or function much slower than normal
NIS Binding Problems

- Commands on a client hang. Sometimes commands hang even though the system as a whole seems fine and you can run new commands
- Commands on a client crash with obscure messages, or no message at all

NIS Problems Affecting One Client

If only one or two clients are experiencing symptoms that indicate NIS binding difficulty, the problems probably are on those clients. If many NIS clients are failing to bind properly, the problem probably exists on one or more of the NIS servers. See “NIS Problems Affecting Many Clients” on page 109.

ypbind Not Running on Client

One client has problems, but other clients on the same subnet are operating normally. On the problem client, run `ls -l` on a directory, such as `/usr`, that contains files owned by many users, including some not in the client `/etc/passwd` file. If the resulting display lists file owners who are not in the local `/etc/passwd` as numbers, rather than names, this indicates that NIS service is not working on the client.

These symptoms usually mean that the client `ypbind` process is not running. Verify whether the NIS client services are running.

```
client# svc -s \nis\*
STATE STIME FMRI
disabled Sep_01 svc:/network/nis/domain:default
disabled Sep_01 svc:/network/nis/client:default
```

If the services are in a disabled state, log in as root or assume an equivalent role, and start the NIS client service.

```
client# svcadm enable network/nis/domain
client# svcadm enable network/nis/client
```

Missing or Incorrect Domain Name

One client has problems, the other clients are operating normally, but `ypbind` is running on the problem client. The client might have an incorrectly set domain.

On the client, run the `domainname` command to see which domain name is set.

```
client7# domainname
example.com
```
Compare the output with the actual domain name in /var/yp on the NIS master server. The actual NIS domain is shown as a subdirectory in the /var/yp directory.

```bash
client7# ls
-l /var/yp
-rwxr-xr-x 1 root Makefile
drwxr-xr-x 2 root binding
drx------ 2 root example.com
```

If the domain name returned by running `domainname` on a machine is not the same as the server domain name listed as a directory in /var/yp, the domain name specified in the machine's /etc/defaultdomain file is incorrect. Reset the NIS domain name as shown in “How to Set a Machine's NIS Domain Name” on page 102.

---

**Note** - The NIS domain name is case-sensitive.

---

### Client Not Bound to Server

If your domain name is set correctly, ypbind is running, and commands still hang, then make sure that the client is bound to a server by running the `ypwhich` command. If you have just started ypbind, then run `ypwhich` several times (typically, the first one reports that the domain is not bound and the second succeeds normally).

---

### No Server Available

If your domain name is set correctly, ypbind is running, and you get messages indicating that the client cannot communicate with a server, this might indicate a number of different problems:

- Does the client have a `/var/yp/binding/domainname/ypservers` file containing a list of servers to bind to? If not, run `ypinit -c` and specify in order of preference the servers that this client should bind to.
- If the client does have a `/var/yp/binding/domainname/ypservers` file, are there enough servers listed in it if one or two become unavailable? If not, add additional servers to the list by running `ypinit -c`.
- Do the selected NIS servers have entries in the `/etc/inet/hosts` file? To view the selected NIS servers, use the `svcprop -p config/ypservers nis/domain` command. If these hosts are not in the local `/etc/inet/hosts` file, add the servers to the hosts NIS
NIS Binding Problems

maps and rebuild your maps by running the ypinit -c or ypinit -s command as described in “Working With NIS Maps” on page 89.

- Is the name service switch set up to check the machine's local hosts file in addition to NIS? See Chapter 2, “About the Name Service Switch” for more information on the switch.
- Is the name service switch set up to check files first for services and rpc? See Chapter 2, “About the Name Service Switch” for more information about the switch.

**ypwhich Displays Are Inconsistent**

When you use ypwhich several times on the same client, the resulting display varies because the NIS server changes. This is normal. The binding of the NIS client to the NIS server changes over time when the network or the NIS servers are busy. Whenever possible, the network becomes stable at a point where all clients get acceptable response time from the NIS servers. As long as your client machine gets NIS service, it does not matter where the service comes from. For example, an NIS server machine can get its own NIS services from another NIS server on the network.

**When Server Binding is Not Possible**

In extreme cases where local server binding is not possible, use of the ypset command can temporarily allow binding to another server, if available, on another network or subnet. However, in order to use the -ypset option, ypbind must be started with either the -ypset or -ypsetme options. For more information, see the ypbind(1M) man page.

```
# /usr/lib/netsvc/yp/ypbind -ypset
```

For another method, see “Binding to a Specific NIS Server” on page 101.

---

**Caution** - For security reasons, the use of the -ypset and -ypsetme options is not recommended. Only use these options for debugging purposes under controlled circumstances. Use of the -ypset and -ypsetme options can result in serious security breaches because while the daemons are running, anyone can alter server bindings, causing trouble for others and permitting unauthorized access to sensitive data. If you must start the ypbind daemon with these options, after you have fixed the problem you must kill the ypbind process and restart it again without those options.

To restart the ypbind daemon, use SMF as follows:

```
# svcadm enable -r svc:/network/nis/client:default
```
ypbind Crashes

If the ypbind daemon crashes almost immediately each time it is started, look for a problem in the svc:/network/nis/client:default service log. Check for the presence of the rpcbind daemon by typing the following:

```
% ps -e |grep rpcbind
```

If rpcbind is not present or does not stay up or behaves strangely, check the svc:/network/rpc/bind:default log file. For more information, see the `rpcbind(1M)` and `rpcinfo(1M)` man pages.

You might be able to communicate with rpcbind on the problematic client from a machine operating normally. From the functioning machine, type the following:

```
% rpcinfo client
```

If rpcbind on the problematic machine is fine, rpcinfo produces the following output:

```
program version netid address service owner
...
100007  3  udp6  ::191.161.0.0 ypbind 1
100007  3  tcp6  ::135.200.0.0 ypbind 1
100007  3  udp  0.0.0.0.240.221 ypbind 1
100007  2  udp  0.0.0.0.240.221 ypbind 1
100007  1  udp  0.0.0.0.240.221 ypbind 1
100007  3  tcp  0.0.0.0.250.107 ypbind 1
100007  2  tcp  0.0.0.0.250.107 ypbind 1
100007  1  tcp  0.0.0.0.250.107 ypbind 1
100007  3  ticlts 2\000\000\000 ypbind 1
100007  2  ticlts 2\000\000\000 ypbind 1
100007  3  ticotsord 0.000.000.0 ypbind 1
100007  2  ticotsord 0.000.000.0 ypbind 1
100007  3  ticots 0.000.000.0 ypbind 1
...
```

Your machine will have different addresses. If the addresses are not displayed, ypbind has been unable to register its services. Reboot the machine and run rpcinfo again. If the ypbind processes are there and they change each time you try to restart the NIS service, reboot the system, even if the rpcbind daemon is running.

NIS Problems Affecting Many Clients

If only one or two clients are experiencing symptoms that indicate NIS binding difficulty, the problems probably are on those clients. See “NIS Problems Affecting One Client” on page 106. If many NIS clients are failing to bind properly, the problem probably exists on one or more of the NIS servers.
**rpc.yppasswdd Considers a Non-Restricted Shell That Begins With r to Be Restricted**

1. Create /etc/default/yppasswdd that contains a special string:
   
   "check_restricted_shell_name=1".
   
2. If the "check_restricted_shell_name=1" string is commented out, the 'r' check will not occur.

**Network or Servers Are Unreachable**

NIS can hang if the network or NIS servers are so overloaded that the ypserv daemon cannot receive a response back to the client ypbind process within the timeout period. NIS can also hang if the network is down.

Under these circumstances, every client on the network experiences the same or similar problems. In most cases, the condition is temporary. The messages usually go away when the NIS server reboots and restarts ypserv, when the load on the NIS servers or the network itself decreases, or when the network resumes normal operations.

**Server Malfunction**

Make sure the servers are up and running. If you are not physically near the servers, use the ping command.

**NIS Daemons Not Running**

If the servers are up and running, try to find a client machine behaving normally, and run the ypwhich command. If ypwhich does not respond, kill it. Then log in as root on the NIS server and check if the NIS process is running by typing the following:

```
# ptree |grep ypbind
100759 /usr/lib/netsvc/yp/ypbind -broadcast
527360 grep yp
```

If neither the ypserv (NIS server) nor the ypbind (NIS client) daemons are running, restart them by typing the following:

```
# svcadm restart network/nis/client
```

If both the ypserv and ypbind processes are running on the NIS server, then run the ypwhich command. If the command does not respond, the ypserv daemon has probably hung and
should be restarted. While logged in as root on the server, restart the NIS service by typing the following:

```
# svcadm restart network/nis/server
```

### Servers Have Different Versions of an NIS Map

Because NIS propagates maps among servers, occasionally you might find different versions of the same map on various NIS servers on the network. This version discrepancy is normal and acceptable if the differences do not last for more than a short time.

The most common cause of map discrepancy is that something is preventing normal map propagation. For example, an NIS server or router between NIS servers is down. When all NIS servers and the routers between them are running, `ypxfr` should succeed.

If the servers and routers are functioning properly, check the following:

- Check the `ypxfr` log output. See “Logging `ypxfr` Output” on page 111.
- Check the `svc:/network/nis/xfr:default` log file for errors.
- Check the control files. See “Check the `crontab` File and `ypxfr` Shell Script” on page 112.
- Check the `ypservers` map on the master server. See “Check the `ypservers` Map” on page 112.

### Logging `ypxfr` Output

If a particular slave server has problems updating maps, log in to that server and run the `ypxfr` command interactively. If the command fails, it indicates why it failed, and you can fix the problem. If the command succeeds, but you suspect it has occasionally failed, create a log file to enable the logging of messages. To create a log file, type the following on the slave.

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

This creates a `ypxfr.log` file that saves all output from `ypxfr`.

The output resembles the output `ypxfr` displays when run interactively, but each line in the log file is time stamped. (You might see unusual ordering in the timestamps. That is okay – the timestamp tells you when `ypxfr` started to run. If copies of `ypxfr` ran simultaneously but their work took differing amounts of time, they might actually write their summary status line to the log files in an order different from that which they were invoked.) Any pattern of intermittent failure shows up in the log.
Note - When you have fixed the problem, turn off logging by removing the log file. If you forget to remove it, the file continues to grow without limit.

Check the crontab File and ypxfr Shell Script

Inspect the root crontab file, and check the ypxfr shell script it invokes. Typographical errors in these files can cause propagation problems. Failures to refer to a shell script within the /var/spool/cron/crontabs/root file, or failures to refer to a map within any shell script can also cause errors.

Check the ypservers Map

Also, make sure that the NIS slave server is listed in the ypservers map on the master server for the domain. If it is not, the slave server still operates perfectly as a server, but yppush does not propagate map changes to the slave server.

Workaround to Update Maps on a Broken Slave Server

If the NIS slave server problem is not obvious, you can perform a workaround while you debug the problem, by using the scp or ssh command to copy a recent version of the inconsistent map from any healthy NIS server. The following shows how to transfer the problem map:

```bash
ypslave# scp ypmaster:/var/yp/mydomain/* /var/yp/mydomain
```

The * character has been escaped in the command line, so that it will be expanded on ypmaster, instead of locally on ypslave.

ypserv Crashes

When the ypserv process crashes almost immediately and does not stay up even with repeated activations, the debugging process is virtually identical to that described in “ypbind Crashes” on page 109. First, run the following command to see if any errors are being reported:

```bash
# svcs -vx nis/server
```

Check for the existence of the rpcbind daemon as follows:

```bash
# ptree |grep rpcbind
```
Reboot the server if you do not find the daemon. Otherwise, if the daemon is running, type the following and look for similar output:

```
% rpcinfo -p ypserver
% program  vers  proto  port  service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100068 2 udp 32813 cmsd
...
100007 1 tcp 34000 ypbind
100004 2 udp 731 ypserv
100004 1 udp 731 ypserv
100004 1 tcp 732 ypserv
100004 2 tcp 32772 ypserv
```

Your machine might have different port numbers. The four entries representing the `ypserv` process are the following:

```
100004 2 udp 731 ypserv
100004 1 udp 731 ypserv
100004 1 tcp 732 ypserv
100004 2 tcp 32772 ypserv
```

If there are no entries, and `ypserv` is unable to register its services with `rpcbind`, reboot the machine. If there are entries, de-register the service from `rpcbind` before restarting `ypserv`. To de-register the service from `rpcbind`, on the server type the following.

```
# rpcinfo -d number 1
# rpcinfo -d number 2
```

where `number` is the ID number reported by `rpcinfo` (100004, in the preceding example).
**application-level naming service**

Application-level naming services are incorporated in applications offering services such as files, mail, and printing. Application-level naming services are bound below enterprise-level naming services. The enterprise-level naming services provide contexts in which contexts of application-level naming services can be bound.

**attribute**

Each LDAP entry consists of a number of named *attributes* each of which has one or more values.

Also, the N2L service mapping and configuration files each consist of a number of named *attributes*. Each attribute has one or more values.

**authentication**

The means by which a server can verify a client's identity.

**baseDN**

The DN where part of the DIT is rooted. When this is the baseDN for an NIS domains entries it is also referred to as a *context*.

**client**

1. The client is a principal (machine or user) requesting an naming service from an naming server.

2. In the client-server model for file systems, the client is a machine that remotely accesses resources of a compute server, such as compute power and large memory capacity.

3. In the client-server model, the client is an *application* that accesses services from a “server process.” In this model, the client and the server can run on the same machine or on separate machines.

**client-server model**

A common way to describe network services and the model user processes (programs) of those services. Examples include the name-server/name-resolver paradigm of the *Domain Name System (DNS)*. See also *client*.

**context**

For the N2L service, a context is something under which a NIS domain is generally mapped. See also baseDN.

**credentials**

The authentication information that the client software sends along with each request to a naming server. This information verifies the identity of a user or machine.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
</table>
| data encrypting key                       | A key used to encipher and decipher data intended for programs that perform encryption. Contrast with *key encrypting key*.
| data encryption standard (DES)            | A commonly used, highly sophisticated algorithm developed by the U.S. National Bureau of Standards for encrypting and decrypting data. See also SUN-DES-1.
| databaseID                                | For the N2L service, a databaseID is an alias for a group of maps containing NIS entries of the same format (having the same mappings to LDAP). The maps might have differing keys.
| DBM                                       | DBM is the database originally used to store NIS maps.
| decimal dotted notation                   | The syntactic representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. Used to represent IP addresses in the Internet as in: 192.168.67.20.
| DES                                       | See *data encryption standard (DES)*.
| directory                                 | (1) An LDAP directory is a container for LDAP objects. In UNIX, a container for files and subdirectories.
| directory cache                           | A local file used to store data associated with directory objects.
| directory information tree                | The DIT is the distributed directory structure for a given network. By default, clients access the information assuming that the DIT has a given structure. For each domain supported by the LDAP server, there is an assumed subtree with an assumed structure.
| distinguished name                        | A distinguished name is an entry in an X.500 directory information base (DIB) composed of selected attributes from each entry in the tree along a path leading from the root down to the named entry.
| DIT                                       | See directory information tree.
| DN                                        | A distinguished name in LDAP. A tree-like structured addressing scheme of the LDAP directory which gives a unique name to each LDAP entry.
| DNS                                       | See *Domain Name System*.
| DNS zone files                            | A set of files wherein the DNS software stores the names and IP addresses of all the workstations in a domain.
| DNS zones                                 | Administrative boundaries within a network domain, often made up of one or more subdomains.
| DNS-forwarding                            | An NIS server forwards requests it cannot answer to DNS servers.
domain (1) In the Internet, a part of a naming hierarchy usually corresponding to a Local Area Network (LAN) or Wide Area Network (WAN) or a portion of such a network. Syntactically, an Internet domain name consists of a sequence of names (labels) separated by periods (dots). For example, sales.example.com.

(2) In International Organization for Standardization's open systems interconnection (OSI), “domain” is generally used as an administrative partition of a complex distributed system, as in MHS private management domain (PRMD), and directory management domain (DMD).

domain name The name assigned to a group of systems on a local network that share DNS administrative files. The domain name is required for the network information service database to work properly. See also domain.

Domain naming service (DNS) A service that provides the naming policy and mechanisms for mapping domain and machine names to addresses outside of the enterprise, such as those on the Internet. DNS is the network information service used by the Internet.

encryption The means by which the privacy of data is protected.

encryption key See data encrypting key.

enterprise-level network An “enterprise-level” network can be a single Local Area Network (LAN) communicating over cables, infra-red beams, or radio broadcast; or a cluster of two or more LANs linked together by cable or direct phone connections. Within an enterprise-level network, every machine is able to communicate with every other machine without reference to a global naming service such as DNS or X.500/LDAP.

entry A single row of data in a database table, such as an LDAP element in a DIT.

field A NIS map entry might consist of a number of components and separator characters. As part of the N2L service mapping process the entry is first broken down into a number of named fields.

GID See group ID.

global naming service A global naming service identifies (names) those enterprise-level networks around the world that are linked together by phone, satellite, or other communication systems. This world-wide collection of linked networks is known as the “Internet.” In addition to naming networks, a global naming service also identifies individual machines and users within a given network.

group ID A number that identifies the default group for a user.

indexed name A naming format used to identify an entry in a table.

Internet address A 32-bit address assigned to hosts using TCP/IP. See decimal dotted notation.

IP Internet Protocol. The network layer protocol for the Internet protocol suite.
<table>
<thead>
<tr>
<th><strong>IP address</strong></th>
<th>A unique number that identifies each host in a network.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>key</strong> (encrypting)</td>
<td>A key used to encipher and decipher other keys, as part of a key management and distribution system. Contrast with <em>data encrypting key</em>.</td>
</tr>
<tr>
<td><strong>key server</strong></td>
<td>An Oracle Solaris operating environment process that stores private keys.</td>
</tr>
<tr>
<td><strong>LDAP</strong></td>
<td>Lightweight Directory Access Protocol is a standard, extensible directory access protocol used by LDAP naming service clients and servers to communicate with each other.</td>
</tr>
<tr>
<td><strong>local-area network (LAN)</strong></td>
<td>Multiple systems at a single geographical site connected together for the purpose of sharing and exchanging data and software.</td>
</tr>
<tr>
<td><strong>mail exchange records</strong></td>
<td>Files that contain a list of DNS domain names and their corresponding mail hosts.</td>
</tr>
<tr>
<td><strong>mail hosts</strong></td>
<td>A workstation that functions as an email router and receiver for a site.</td>
</tr>
<tr>
<td><strong>mapping</strong></td>
<td>The process of converting NIS entries to or from DIT entries. This process is controlled by a <em>mapping</em> file.</td>
</tr>
<tr>
<td><strong>master server</strong></td>
<td>The server that maintains the master copy of the network information service database for a particular domain. Namespace changes are always made to the naming service database kept by the domain's master server. Each domain has only one master server.</td>
</tr>
<tr>
<td><strong>MIS</strong></td>
<td>Management information systems (or services).</td>
</tr>
<tr>
<td><strong>N2L server</strong></td>
<td>NIS-to-LDAP server. An NIS master server that has been reconfigured as an N2L server by using the N2L service. Reconfiguration includes replacing NIS daemons and adding new configuration files.</td>
</tr>
<tr>
<td><strong>name resolution</strong></td>
<td>The process of translating workstation or user names to addresses.</td>
</tr>
<tr>
<td><strong>name server</strong></td>
<td>Servers that run one or more network naming services.</td>
</tr>
<tr>
<td><strong>name service switch</strong></td>
<td>The <em>svc:/system/name-service/switch</em> service which defines the sources from which an naming client can obtain its network information.</td>
</tr>
</tbody>
</table>
| **namespace** | (1) A namespace stores information that users, workstations, and applications must have to communicate across the network.  
(2) The set of all names in a naming system. |
<p>| <strong>naming service</strong> | A network service that handles machine, user, printer, domain, router, an other network names and addresses. |
| <strong>NDBM</strong> | NDBM is an improved version of DBM. |</p>
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>network mask</strong></td>
<td>A number used by software to separate the local subnet address from the rest of a given Internet protocol address.</td>
</tr>
<tr>
<td><strong>network password</strong></td>
<td>See Secure RPC password.</td>
</tr>
<tr>
<td><strong>NIS</strong></td>
<td>A distributed network information service containing key information about the systems and the users on the network. The NIS database is stored on the master server and all the replica or slave servers.</td>
</tr>
<tr>
<td><strong>NIS maps</strong></td>
<td>A file used by NIS that holds information of a particular type, for example, the password entries of all users on a network or the names of all host machines on a network. Programs that are part of the NIS service query these maps. See also NIS.</td>
</tr>
<tr>
<td><strong>preferred server list</strong></td>
<td>A client_info table or a client_info file. Preferred server lists specify the preferred servers for a client or domain.</td>
</tr>
<tr>
<td><strong>private key</strong></td>
<td>The private component of a pair of mathematically generated numbers, which, when combined with a private key, generates the DES key. The DES key in turn is used to encode and decode information. The private key of the sender is only available to the owner of the key. Every user or machine has its own public and private key pair.</td>
</tr>
<tr>
<td><strong>public key</strong></td>
<td>The public component of a pair of mathematically generated numbers, which, when combined with a private key, generates the DES key. The DES key in turn is used to encode and decode information. The public key is available to all users and machines. Every user or machine has their own public and private key pair.</td>
</tr>
<tr>
<td><strong>RDN</strong></td>
<td>Relative Distinguished Name. One part of a DN.</td>
</tr>
<tr>
<td><strong>record</strong></td>
<td>See entry.</td>
</tr>
<tr>
<td><strong>remote procedure call (RPC)</strong></td>
<td>An easy and popular paradigm for implementing the client-server model of distributed computing. A request is sent to a remote system to execute a designated procedure, using arguments supplied, and the result is returned to the caller.</td>
</tr>
<tr>
<td><strong>reverse resolution</strong></td>
<td>The process of converting workstation IP addresses to workstation names using the DNS software.</td>
</tr>
<tr>
<td><strong>RFC 2307</strong></td>
<td>RFC specifying a mapping of information from the standard NIS maps to DIT entries. By default, the N2L service implements the mapping specified in an updated version RFC 2307bis.</td>
</tr>
<tr>
<td><strong>RPC</strong></td>
<td>See remote procedure call (RPC).</td>
</tr>
<tr>
<td><strong>SASL</strong></td>
<td>The simple authentication and security layer. A framework for negotiating authentication and security layer semantics in application-layer protocols.</td>
</tr>
<tr>
<td><strong>schema</strong></td>
<td>A set of rules defining what types of data can be stored in any given LDAP DIT.</td>
</tr>
<tr>
<td><strong>searchTriple</strong></td>
<td>A description of where to look for a given attribute in the DIT. The searchTriple is composed of a 'base dn', 'scope' and 'filter'. This is part of the LDAP URL format as defined in RFC 2255.</td>
</tr>
</tbody>
</table>
Secure RPC password

Password required by the secure RPC protocol. This password is used to encrypt the private key. This password should always be identical to the user's login password.

server

(1) In NIS, DNS, and LDAP a host machine providing naming services to a network.

(2) In the client-server model for file systems, the server is a machine with computing resources (and is sometimes called the compute server), and large memory capacity. Client machines can remotely access and make use of these resources. In the client-server model for window systems, the server is a process that provides windowing services to an application, or “client process.” In this model, the client and the server can run on the same machine or on separate machines.

(3) A daemon that actually handles the providing of files.

server list

See preferred server list.

slave server

A server system that maintains a copy of the NIS database. It has a disk and a complete copy of the operating environment.

source

NIS source files

SSL

SSL is the secure sockets layer protocol. It is a generic transport-layer security mechanism designed to make application protocols such as LDAP secure.

subnet

A working scheme that divides a single logical network into smaller physical networks to simplify routing.

suffix

In LDAP, the distinguished name (DN) of the DIT.

TCP

See Transport Control Protocol (TCP).

TCP/IP

Acronym for Transport Control Protocol/Interface Program. The protocol suite originally developed for the Internet. It is also called the Internet protocol suite. Oracle Solaris networks run on TCP/IP by default.

Transport Control Protocol (TCP)

The major transport protocol in the Internet suite of protocols providing reliable, connection-oriented, full-duplex streams. Uses IP for delivery. See TCP/IP.

Transport Layer Security (TLS)

TLS secures communication between an LDAP client and the directory server, providing both privacy and data integrity. The TLS protocol is a super set of the Secure Sockets Layer (SSL) protocol.

wide-area network (WAN)

A network that connects multiple local-area networks (LANs) or systems at different geographical sites by phone, fiber-optic, or satellite links.
**X.500**  
A global-level directory service defined by an Open Systems Interconnection (OSI) standard. A precursor to LDAP.

**yp**  
Yellow Pages™. The old name for NIS which is still used within the NIS code.
Index

A
AAAA records, 19
Active Directory
  AD naming service, 45
  configuring nss_ad, 46
retrieving
  group information, 49
  passwd information, 48
  shadow information, 49
setting up clients, 45
updating passwords, 48
adjunct file, 71
aliases file, 69
attribute
definition, 115
audit_attr map
described, 57
audit_user map
described, 57
authentication
definition, 115
auto_direct.time map, 93
auto_home table
  name service switch and, 26
auto_home.time map, 93
auto_master table
  name service switch and, 26

B
baseDN
definition, 115
bootparams map
described, 57
broadcast

NIS binding, 60

C
CHKPIPE, 95
client
definition, 115
client-server model
definition, 115
clients
  NIS, 53
  NIS setup, 82
commands
  DNS, 42
  NIS, 55
compile flags
  DNS, 43
configure
  DNS server, 34
  DNS server options, 36
context
definition, 115
creating
  rndc.conf file, 35
credentials
definition, 115
crontab file
  NIS problems and, 112
  ypxfr and, 97

diagnostics
  DNS, 42
  NIS, 54
  not running, 110
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>data encrypting key</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>data encryption standard</td>
<td>DES</td>
</tr>
<tr>
<td>databaseID</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>dbm files</td>
<td>100</td>
</tr>
<tr>
<td>101</td>
<td></td>
</tr>
<tr>
<td>decimal dotted notation</td>
<td>116</td>
</tr>
<tr>
<td>DES</td>
<td>116, 116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>dig command</td>
<td>42</td>
</tr>
<tr>
<td>description</td>
<td>42</td>
</tr>
<tr>
<td>DIR directory</td>
<td>69</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>directory cache</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>directory information tree</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>distinguished name</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>DIT</td>
<td></td>
</tr>
<tr>
<td>See directory information tree</td>
<td></td>
</tr>
<tr>
<td>DN</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>DNS</td>
<td></td>
</tr>
<tr>
<td>advertising resources</td>
<td>41</td>
</tr>
<tr>
<td>commands</td>
<td>42</td>
</tr>
<tr>
<td>compile flags</td>
<td>43</td>
</tr>
<tr>
<td>daemons</td>
<td>42</td>
</tr>
<tr>
<td>definition</td>
<td>116, 117</td>
</tr>
<tr>
<td>files</td>
<td>42</td>
</tr>
<tr>
<td>FMRRIs</td>
<td>33</td>
</tr>
<tr>
<td>name service switch and</td>
<td>21</td>
</tr>
<tr>
<td>NIS and</td>
<td>51, 52, 103</td>
</tr>
<tr>
<td>overview</td>
<td>15, 31</td>
</tr>
<tr>
<td>related information</td>
<td>32</td>
</tr>
<tr>
<td>SMF and</td>
<td>32</td>
</tr>
<tr>
<td>tasks</td>
<td>34</td>
</tr>
<tr>
<td>user authorizations</td>
<td>37</td>
</tr>
<tr>
<td>DNS client</td>
<td></td>
</tr>
<tr>
<td>install</td>
<td>38</td>
</tr>
<tr>
<td>DNS package</td>
<td></td>
</tr>
<tr>
<td>install</td>
<td>34</td>
</tr>
<tr>
<td>DNS server</td>
<td></td>
</tr>
<tr>
<td>configure</td>
<td>34</td>
</tr>
<tr>
<td>configure options</td>
<td>36</td>
</tr>
<tr>
<td>troubleshooting</td>
<td>38</td>
</tr>
<tr>
<td>DNS service discovery</td>
<td></td>
</tr>
<tr>
<td>configuration</td>
<td>40</td>
</tr>
<tr>
<td>overview</td>
<td>16, 32</td>
</tr>
<tr>
<td>DNS zone files</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>DNS zones</td>
<td>116</td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>DNS-forwarding</td>
<td></td>
</tr>
<tr>
<td>definition</td>
<td>116</td>
</tr>
<tr>
<td>dns -sd command</td>
<td></td>
</tr>
<tr>
<td>advertising resources</td>
<td>41</td>
</tr>
<tr>
<td>description</td>
<td>42</td>
</tr>
<tr>
<td>dnssec-dsfromkey command</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>43</td>
</tr>
<tr>
<td>dnssec-keyfromlabel command</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>43</td>
</tr>
<tr>
<td>dnssec-keygen command</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>43</td>
</tr>
<tr>
<td>dnssec-signzone command</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>43</td>
</tr>
<tr>
<td>DMO variable</td>
<td>73, 73</td>
</tr>
<tr>
<td>domain</td>
<td></td>
</tr>
<tr>
<td>definition</td>
<td>117</td>
</tr>
<tr>
<td>domain name</td>
<td>117</td>
</tr>
<tr>
<td>NIS slave servers and</td>
<td>77</td>
</tr>
<tr>
<td>setting</td>
<td>66</td>
</tr>
<tr>
<td>domain name system</td>
<td></td>
</tr>
<tr>
<td>See DNS domain name system (DNS)</td>
<td></td>
</tr>
<tr>
<td>extensions for IPv6</td>
<td>19</td>
</tr>
<tr>
<td>domainname command</td>
<td></td>
</tr>
<tr>
<td>NIS and</td>
<td>107</td>
</tr>
<tr>
<td>domains</td>
<td></td>
</tr>
<tr>
<td>multiple NIS</td>
<td>73</td>
</tr>
<tr>
<td>NIS, 52, 54, 66</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
</tr>
<tr>
<td>/etc files</td>
<td>56</td>
</tr>
<tr>
<td>naming and</td>
<td>15</td>
</tr>
<tr>
<td>/etc/inet/hosts file</td>
<td>10</td>
</tr>
<tr>
<td>NIS slave servers and</td>
<td>78</td>
</tr>
<tr>
<td>/etc/mail directory</td>
<td>69</td>
</tr>
<tr>
<td>/etc/mail/aliases file</td>
<td>69</td>
</tr>
</tbody>
</table>
/etc/named.conf file
  description, 42
  DNS user authorizations, 37
  verifying configuration, 39
/etc/rndc.conf file
  description, 42
  encryption
    definition, 117
  encryption key
    definition, 117
  enterprise-level network
    definition, 117
entry
  definition, 117
ethers.byaddr map
  described, 57
ethers.byname map
  described, 57
exec_attr map
  described, 57

F
field
  definition, 117
files
  DNS, 42
files-based naming, 16
FMRI
  DNS, 33
  mDNS, 41
  NIS, 64

global naming service
  definition, 117
group ID
  definition, 117
group.bygid map
  described, 57
group.byname map
  described, 57
groups
  netgroups (NIS), 88, 89

H
host command
  description, 43
host name
  setting, 66
host.byaddr map
  described, 57
hostbyname map
  described, 57
hosts
  (machines)
    changing NIS domains, 102
    NIS clients, 53
    NIS servers, 53
hosts database, 95
hosts file
  NIS slave servers and, 78
hosts.byaddr map, 56
hostsbyname map, 56

I
indexed name
  definition, 117
install
  DNS client, 38
  DNS package, 34
Internet
  NIS and, 52
Internet access
  name service switch and, 21
Internet address
  definition, 117
IP
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition, 117</td>
</tr>
<tr>
<td>IP address definition, 118</td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>key (encrypting) definition, 118</td>
</tr>
<tr>
<td>key server definition, 118</td>
</tr>
<tr>
<td>keyserv name service switch and, 26</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>LAN definition, 118</td>
</tr>
<tr>
<td>LDAP definition, 118</td>
</tr>
<tr>
<td>lightweight directory access protocol See LDAP</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>mail exchange records definition, 118</td>
</tr>
<tr>
<td>mail hosts definition, 118</td>
</tr>
<tr>
<td>mail.aliases map described, 57</td>
</tr>
<tr>
<td>mail.byaddr map described, 57</td>
</tr>
<tr>
<td>make command after updating maps, 96</td>
</tr>
<tr>
<td>description, 55</td>
</tr>
<tr>
<td>Makefile syntax, 93</td>
</tr>
<tr>
<td>NIS maps, 59</td>
</tr>
<tr>
<td>ypinit and, 73</td>
</tr>
<tr>
<td>makedbm command adding slave servers, 80</td>
</tr>
<tr>
<td>changing map server, 91</td>
</tr>
<tr>
<td>description, 55</td>
</tr>
<tr>
<td>make command and, 56</td>
</tr>
<tr>
<td>Makefile and, 71</td>
</tr>
<tr>
<td>non-default maps and, 100</td>
</tr>
<tr>
<td>ypinit and, 73</td>
</tr>
<tr>
<td>Makefile file automounter maps and, 93</td>
</tr>
<tr>
<td>changing a map's master server, 91</td>
</tr>
<tr>
<td>changing source directory, 67, 71</td>
</tr>
<tr>
<td>conversion to NIS and, 70</td>
</tr>
<tr>
<td>maps supported list, 92</td>
</tr>
<tr>
<td>NIS, 56</td>
</tr>
<tr>
<td>NIS security, 85</td>
</tr>
<tr>
<td>non-default maps modifying, 99</td>
</tr>
<tr>
<td>passwd maps and, 71</td>
</tr>
<tr>
<td>preparing, 70</td>
</tr>
<tr>
<td>setting up primary server, 73</td>
</tr>
<tr>
<td>mapname.dir file, 71</td>
</tr>
<tr>
<td>mapname.pag file, 71</td>
</tr>
<tr>
<td>mapping definition, 118</td>
</tr>
<tr>
<td>master server definition, 118</td>
</tr>
<tr>
<td>mDNS configuration, 40</td>
</tr>
<tr>
<td>error log, 41</td>
</tr>
<tr>
<td>overview, 16, 31</td>
</tr>
<tr>
<td>MIS definition, 118</td>
</tr>
<tr>
<td>multicast DNS See mDNS</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>“not responding” messages (NIS), 105</td>
</tr>
<tr>
<td>name resolution definition, 118</td>
</tr>
<tr>
<td>name server definition, 118</td>
</tr>
<tr>
<td>name service switch actions, 25</td>
</tr>
<tr>
<td>auto_home table, 26</td>
</tr>
<tr>
<td>auto_master table, 26</td>
</tr>
<tr>
<td>configuring default sources, 29</td>
</tr>
<tr>
<td>search criteria, 28</td>
</tr>
<tr>
<td>sources for databases, 27</td>
</tr>
<tr>
<td>databases, 22</td>
</tr>
<tr>
<td>definition, 118</td>
</tr>
<tr>
<td>DNS and, 21</td>
</tr>
</tbody>
</table>
Internet access, 21
introduction, 21
keyserv service, 26
mDNS and, 41
messages, 24
migrating from nsswitch.conf file, 29
modifying, 26
NIS, 52
NOTFOUND=continue search criteria, 25
options, 25
password data and, 30
password information, 30
publickey properties, 26
search criteria, 25
SMF properties, 22
source formats, 24
status messages, 24, 25
SUCCESS=return search criteria, 25
timezone table and, 26
TRYAGAIN=3 search criteria, 25
TRYAGAIN=forever search criteria, 25
UNAVAIL=continue search criteria, 25
named daemon
configuration file
description, 42
description, 43
showing compile flags, 43
SMF and, 32
troubleshooting with, 38
user authorizations and, 37
named-checkconf command
configure DNS server, 34
description, 43
verifying /etc/named.conf file, 39
named-checkzone command
description, 43
named-compilezone command
description, 43
named.conf file See /etc/named.conf file
namespace
definition, 118
naming
comparison, 18
DNS, 18
files-based, 16
IPv6 extensions, 18
LDAP, 17
NIS, 17
Oracle Solaris naming services, 15
overview, 9
naming service
definition, 118
ndbm format, 70
NIS maps and, 56
netgroup map
tables, 89
overview, 88
netgroup.byhost map
described, 57
overview, 88
netgroup.byuser map
described, 57
overview, 88
netid.byname map
described, 58
netmasks.byaddr map
described, 58
network mask
definition, 119
network password See secure RPC password
network services
DNS and, 32
networks.byaddr map
described, 58
networks.bynename map
described, 58
nicknames file, 60
NIS, 17
/var/yp/domainname directory and, 57
architecture, 52
automatic starting, 75
binding, 60
broadcast binding, 61
client problems, 106
client setup, 82
clients, 53, 53
commands, 55
commands hang, 106
components, 54
daemons, 54
Index

definition, 119
dns and, 52, 103
domain names, 66
domains, 52, 54
halting, 104
internet and, 52
introduction, 51
makefile, 56
makefile filtering, 93
makefile preparation, 70
manual binding, 101
master servers, 53
modifying configuration files, 92
multiple domains, 73
ndbm format, 56
netgroups, 88, 89
overloaded servers and, 110
passwd maps auto update, 98
password data, 67, 67
preparation for, 64
problems, 105
root entry, 85
rpc.ypassword daemon, 88
security, 85
server binding not possible, 108
server-list binding, 61
servers, 53, 53
servers not available, 107
servers, maps different versions, 111
setup preparation, 67
slave server setup, 77, 77
slave servers, 53
SMF and, 64
source files, 67, 68
starting daemons, 74
stopping, 104
structure of, 52
updating passwd maps, 87
user password locked, 86
user passwords, 87
useradd, 86
userdel, 87
users, administering, 86
ypbind daemon, 61
ypbind fails, 109
ypbind “can’t” messages, 105
ypinit, 72
ypservers file, 80
ypwhich, 60
ypwhich inconsistent displays, 108
“not responding” messages, 105
“unavailable” messages, 105
NIS clients
not bound to server, 107
NIS daemons
not running, 110
NIS domain names
incorrect, 106
missing, 106
NIS domains
changing, 102
NIS hosts
changing domain of, 102
NIS maps
/var/yp/domainname directory and, 57
administering, 89
changing Makefile macros, 93
changing Makefile variables, 93
changing server, 90
CHKPIPE in Makefile, 95
creating from files, 100
creating from keyboard, 100
default, 57
definition, 119
displaying contents, 89
displaying contents of, 59
list of, 57
locating, 59
Makefile and, 92
Makefile DIR variable, 94
Makefile DOM variable, 94
Makefile filtering, 93
Makefile PWDIR variable, 94
making, 59
modifying configuration files, 92
ndbm format, 56
nicknames, 59
non-default, 96
NOSEIT in Makefile, 95
updating, 59

128  Working With Oracle Solaris 11.2 Directory and Naming Services: DNS and NIS • July 2014
working with, 59
yppush in Makefile, 95
NIS servers
   malfunction, 110
NIS slave servers
   adding, 80
   initializing, 81
node name
   setting, 66
NOPUSH in Makefile, 95
NOTFOUND=continue search criteria
   name service switch and, 25
ns daemon
   description, 55
ns command
   description, 43
nslookup command
   description, 43
nsstyle.conf
   migrating to SMF, 29
nsupdate command
   description, 43
passwd.adjunct
   file, 71
   file
   Solaris 1.x formats, 86
   passwd file
   Solaris 1.x formats, 86
   passwd map, 67
   passwd maps
   users, adding, 86
passwd.adjunct file, 71, 92
passwd.adjunctbyname map
   described, 58
passwd.bynumber map
   description, 43
passwd.byuid map
   described, 58
password data
   name service switch, 30
   NIS, 67, 67
   NIS, and, 85
   root in NIS maps, 85
passwords
   NIS, 87
   rpc.yppasswdd daemon, 88
private key
   definition, 119
prof.attr map
   described, 58
protocols.bynumber map
   described, 58
protocols.byuid map
   described, 58
public key
   definition, 119
publickey.byuid map
   described, 58
PWDIR, 68

O
Oracle Solaris naming services, 15
   and Service Management Facility (SMF), 15

P
$PWDIR/security/passwd.adjunct, 92
/PWDIR/shadow file, 71
/PWDIR/security/passwd.adjunct, 71
passwd
   NIS map auto updated, 98
passwd command, 87
passwd file
passwd map, 67
passwd maps
   users, adding, 86
passwd.adjunct file, 71, 92
passwd.adjunctbyname map
   described, 58
passwd.bynumber map
   description, 43
passwd.byuid map
   described, 58
password data
   name service switch, 30
   NIS, 67, 67
   NIS, and, 85
   root in NIS maps, 85
passwords
   NIS, 87
   rpc.yppasswdd daemon, 88
private key
   definition, 119
prof.attr map
   described, 58
protocols.bynumber map
   described, 58
protocols.byuid map
   described, 58
public key
   definition, 119
publickey.byuid map
   described, 58
PWDIR, 68

R
record
   definition, 119
reverse resolution
   definition, 119
rndc command
   configuration file
      description, 42
   description, 43
rndc-confgen command
   configure DNS server, 34
   create rndc.conf file, 35
   description, 43
rndc.conf file
   creating, 35
RPC
   definition, 119, 119
rpc.bynumber map
described, 58
rpc.yppasswd daemon
description, 55
NIS passwords and, 88
passwd command updates maps, 98
rpc.yputdated daemon
description, 55

S
SASL
definition, 119
schema
definition, 119
searchTriple
definition, 119
secure RPC password
definition, 120
secure sockets layer See SSL
security
NIS, 67, 67
NIS, and, 85
root in NIS maps, 85
server
definition, 120
server list
definition, 120
NIS binding, 60
servers
NIS slave setup, 77, 77
not available (NIS), 107
preparing NIS servers, 67
ypservers file, 80
service discovery See DNS service discovery
Service Management Facility See SMF
Service Management Facility (SMF)
and Oracle Solaris naming services, 15
services.bbyname map
described, 58
services.byservice map
described, 58
setup
multiple NIS domains, 73
NIS clients, 82
NIS Makefile, 70
NIS slave servers, 77, 77
preparation for NIS, 64, 67
shadow file, 71
Solaris 1.x formats, 86
sites.bbyname map
changing map server, 91
slave server
definition, 120
SMF, 74
DNS and, 32
NIS and, 64
source
definition, 120
SSL
definition, 120
starting
NIS daemons, 74
stopping
NIS daemons, 74
subnet
definition, 120
SUCCESS=return search criteria
name service switch and, 25
suffix
definition, 120
svc:/network/dns/client
described, 33
svc:/network/dns/server
described, 33
svcadm
with NIS, 81

T
tasks
DNS, 34
TCP See transport control protocol
TCP/IP
definition, 120
timezone table, 26
TLS See transport layer security
transport control protocol
definition, 120
transport layer security
definition, 120
troubleshooting
  DNS server, 38

U
/usr/bin/dns-sd command
description, 42
/usr/sbin/dig command
description, 42
/usr/sbin/dnssec-dsfromkey command
description, 43
/usr/sbin/dnssec-keyfromlabel command
description, 43
/usr/sbin/dnssec-keygen command
description, 43
/usr/sbin/dnssec-signzone command
description, 43
/usr/sbin/host command
description, 43
/usr/sbin/makedbm command
  modifying non-default maps, 100
/usr/sbin/named daemon
description, 43
/usr/sbin/named-checkconf command
description, 43
/usr/sbin/named-checkzone command
description, 43
/usr/sbin/named-compilezone command
description, 43
/usr/sbin/nscfg command
description, 43
/usr/sbin/nslookup command
description, 43
/usr/sbin/nsupdate command
description, 43
/usr/sbin/rndc command
description, 43
/usr/sbin/rndc-confgen command
description, 43
  “unavailable” messages (NIS), 105
UNAVAIL=continue search criteria
  name service switch and, 25
user authorizations
  for DNS, 37
user_attr map
  described, 58
useradd, 86
  password is locked, 86
userdel, 87, 87
usermod command
  DNS user authorizations, 37
users
  netgroups, 88, 89
  NIS, 86
  NIS passwords, 87
  updating passwd maps, 87
useradd, 86
userdel (NIS), 87

V
/var/spool/cron/crontabs/root file
  NIS problems and, 112
/var/svc/log/network-dns-multicast:default.log file, 41
/var/svc/log/network-dns-server:default.log file
  troubleshooting, 38
/var/yp directory
  NIS security, 85
/var/yp/binding/domainname/ypservers file, 107
/var/yp/domainname directory, 57
/var/yp/Makefile, 73
  maps
    supported list, 92
/var/yp/mymap.asc file, 100
/var/yp/nicknames file, 60
verifying
  /etc/named.conf file, 39

W
WAN
  definition, 120

X
X.500
definition, 121

Y
yp
  definition, 121
ypbind daemon, 74
adding slave servers, 81
broadcast mode, 61, 82
client not bound, 107
description, 55
fails, 109
overloaded servers and, 110
server-list mode, 61
“can’t” messages, 105
ypcat command, 59
description, 55
ypinit command
  adding slave servers, 81
  client setup, 82
default maps, 96
description, 55
  initializing a slave server, 77
  make command and, 73
  Makefile file and, 70
  master server setup, 72
  slave servers and, 77
  starting yperv, 75
ypmatch command
  description, 55
yppush command
  changing map server, 91
description, 56
  Makefile and, 95
  NIS problems, 112
ypserv daemon, 61, 74
  broadcast mode, 61
description, 55
  failure of, 112
  overloaded servers and, 110
ypservers file
  adding slave server, 80
  creating, 80
  NIS troubleshooting with, 107
ypservers map
  described, 58
  NIS problems, 112
ypset command
description, 56
ypwhich command
description, 56
display inconsistent, 108
identifying bound server, 60
identifying master server, 59
ypxfr command
  changing map server, 91
crontab file and, 97
description, 56
distributing new maps to slave servers, 100
logging output, 111
shell script, 112
ypxfrd daemon
description, 55