The NIS+ Environment

Student Guide

Sun

Sun Microsystems Computer Corporation
Technical Education Services
MS UMIL07-14
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part Number SA-285
Revision B, July 1993

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun logo, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCworks, and SPARCompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark and product of the Massachusetts Institute of Technology. Ethernet is a registered trademark of Xerox Corporation. Intel is a trademark of Intel Corporation. MS-DOS is a registered trademark of Microsoft Corporation. VMS is a registered trademark of Digital Equipment Corporation.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Contents

The NIS+ Environment 1

Objectives

Upon completion of this lesson, you will be able to:

- Define the function of NIS+.
- List the components of an NIS+ domain.
- Name the two reserved directory objects and describe their contents in an NIS+ domain.
- Explain the difference between authentication and authorization.
- Describe the four types of access rights and the four principal categories used in NIS+ authorization.
- Describe the three types of NIS+ security levels.
- Name the NIS+ service daemon and the script that starts it.
- Describe the name service switch process.
- Determine which of the four versions of the name service switch configuration files is appropriate for your network.

References

SunOS 5.1 Administering NIS+ and DNS

Introduction

With increased network connectivity comes an increase in the amount of information that is required by each machine on the network.

- All Internet networks require that each host know the address and host name of other machines on the network.
- Maintaining a common view of home directories with the automounter requires consistent `auto_master` and `auto_home` maps and common password information.
- Boot servers must have `bootparams` information available in order to identify their clients.

Maintaining all of this information and more in data files stored on each machine is cumbersome on a small network of two or three machines, and unmanageable on a larger network.

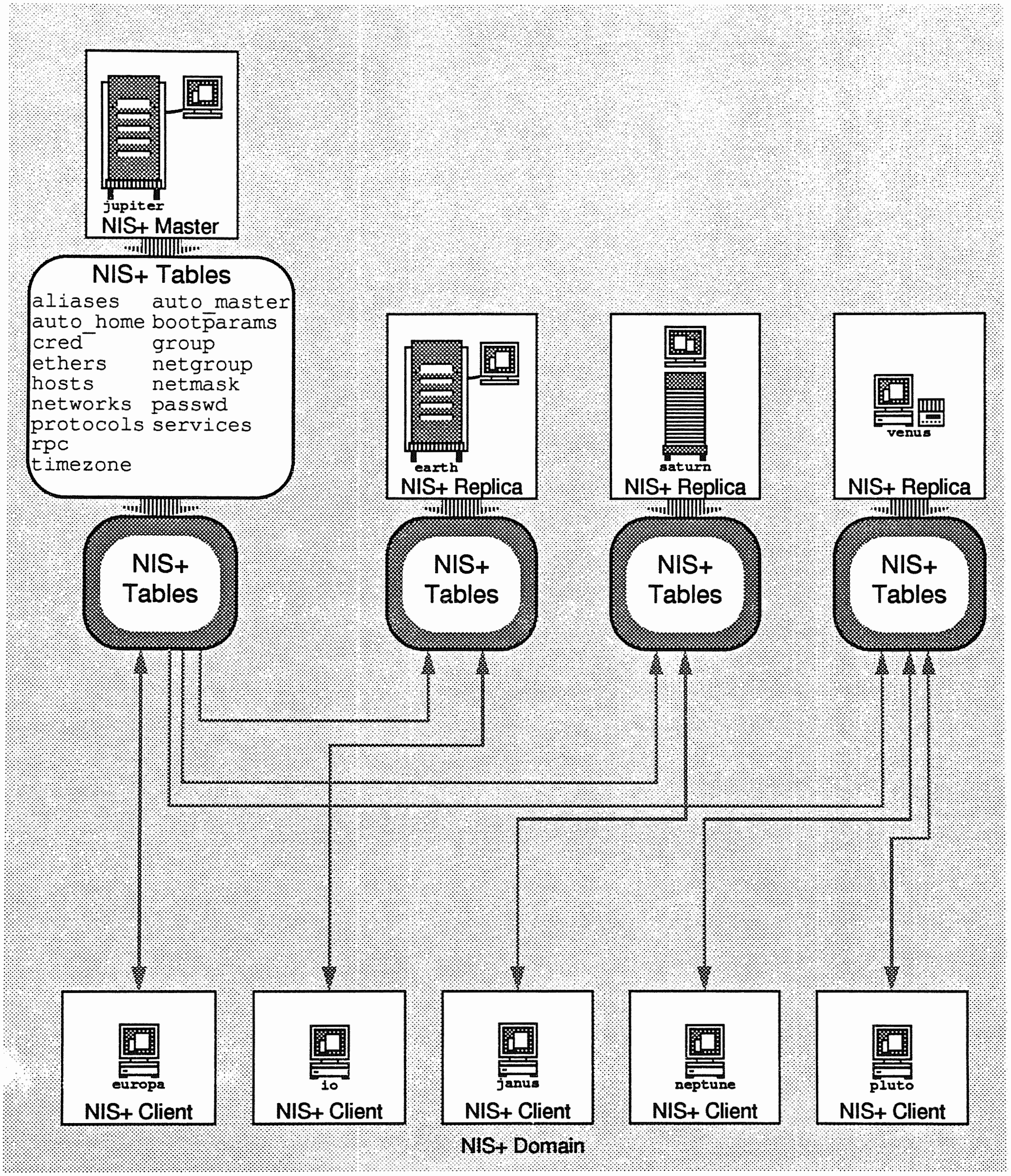
The NIS+ environment takes care of this problem by making information available to a set of machines from servers. NIS+ is an enhanced version of Sun's NIS product.

In this lesson you will learn the supporting concepts needed to configure the NIS+ environment in a local area network.

What is NIS+?

NIS+ is a service that provides information about users, workstations, and network resources. It makes this information available to users and applications that request it and provides security measures to protect against unauthorized access.

The NIS+ Client-Server Model



The NIS+ Client-Server Model

NIS+ solves the problem of having common information available to multiple machines by storing this information in databases on server machines that make the information available to clients on demand.

Clients

A client is a process or machine that sends requests for information to the network. These processes use calls to the RPC library to make requests.

Servers

A server is a process that gets client process requests, looks up requested information in a database, and returns the information to the client process. The term server is also applied to machines that run server processes and store databases on their disks. Every domain is served by one *master server* and 0 or more *replica servers*.

Master Server

A master server contains the master set of database information in the form of tables. Changes are made to these tables, and are automatically pushed to the replica servers.

Replica Server

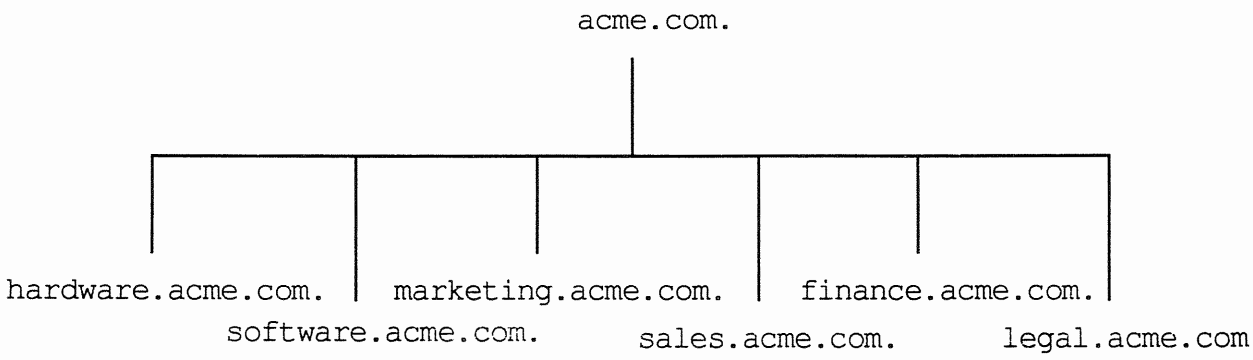
A replica server maintains copies of the tables to distribute the burden of answering client requests and provide backup sources of information in case the master server is down.

Hierarchical NIS+ Domains

A set of machines and the information that is served to those machines is called a *domain*. NIS+ domains can be divided into subdomains to accurately reflect the hierarchical structure of an organization.

An Example of Hierarchical Domains

For example, a company named Acme Inc. with six divisions, Hardware, Software, Marketing, Sales, Finance, and Legal, might set up the following domain hierarchy.



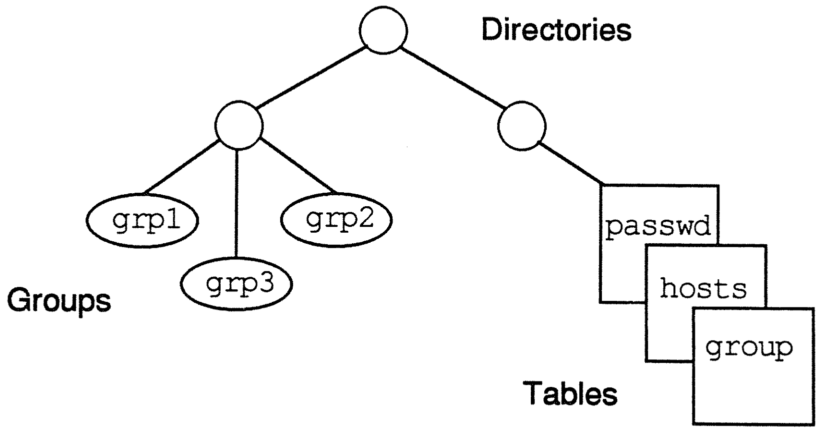
NIS+ Objects

The *NIS+ namespace* is a hierarchical structure in which the NIS+ information is stored. Each namespace has a *root master server* that serves the root domain at the top of the namespace.

It is similar in structure to the Solaris® file system, but you cannot access it with Solaris file commands. The NIS+ namespace is accessed with NIS+ commands (which are covered in the next lesson.)

The three most common kinds of objects in the NIS+ namespace are *directory objects*, *table objects*, and *group objects*.

- Directory objects are the main components of the namespace. They contain other directory objects, table objects, and group objects.
- Table objects store the information in the NIS+ namespace. The Solaris 2.*x* environment provides 16 types of tables, each of which stores a different type of information about users, workstations, or resources on the network. A set of NIS+ tables stores information for that particular domain only.
- Group objects are used for NIS+ security. An NIS+ group is a collection of users and workstations that are identified by a single name that are used to facilitate NIS+ security. Although NIS+ groups are optional, they are a security convenience, allowing the administrator to assign the same access rights to a group of users and workstations.



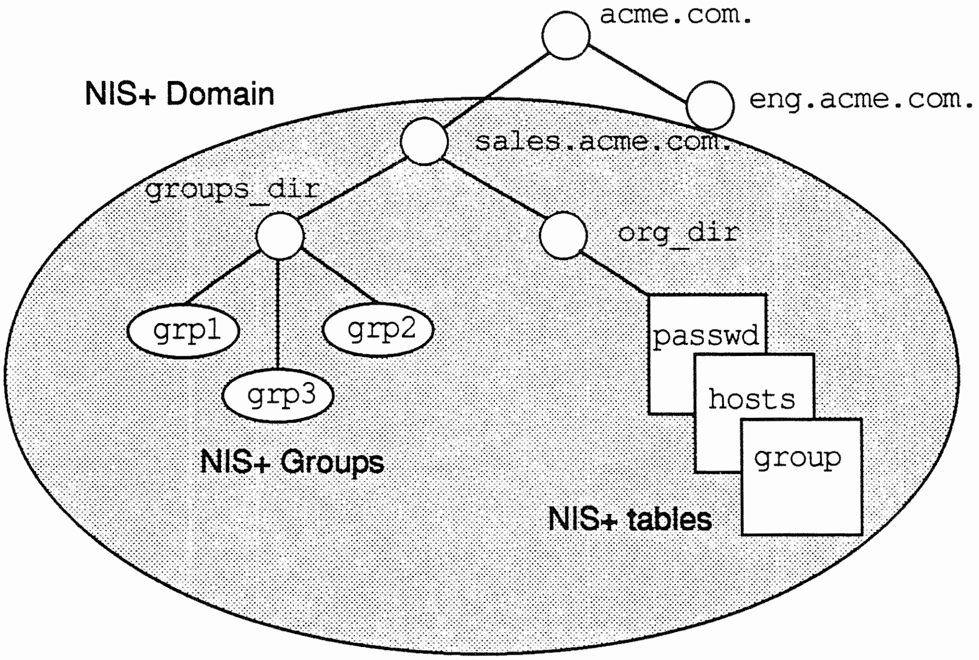
Directory Objects

Directory objects make up the framework of the namespace. The directory object at the top of the namespace is called the *root directory.* The root directory name must always end with a period. This directory represents the root (or top) domain in the namespace hierarchy.

The `org_dir` directory stores the NIS+ table objects. The `groups_dir` directory stores NIS+ group objects.

An *NIS+ domain* consists of a directory object, an `org_dir` subdirectory with a set of NIS+ tables, and a `groups_dir` subdirectory. Although it is not necessary for a domain to contain NIS+ groups, they are also considered part of the domain.

The topmost directory is the *root directory.* If a namespace is flat, it has only one directory, but that directory is still the root directory. The directory objects beneath root directory are simply called "directories."



The above NIS+ domain `acme.com.` is structured to accurately reflect its company-wide organizational hierarchy. The two subdirectories `eng.acme.com.` and `sales.acme.com.` represent two subdomains (or non-root domains) within the `acme.com.` domain.

Using NIS+ Object Names

The NIS+ object names are formed by appending the root directory name (including the period) to their names. This is called a *fully qualified* name. For example, `sales.acme.com.` and `eng.acme.com.` are both fully qualified names[1] that represent the `sales.acme.com.` and `eng.acme.com.` subdomains within the `acme.com.` domain.

Like UNIX® files and directories, NIS+ objects can be referred to by their partial or full names.

Partially Qualified Names

A partially-qualified NIS+ component name is analogous to a relative path name and is simply the name of the component. For example, the `hosts` table's partially qualified name is as follows:

```
hosts
```

Fully Qualified Names

A fully qualified name is the complete name of the component. It is formed by starting with the name of the component and appending the names of all involved components up to the root domain, separated by dots (`.`). For example, the fully qualified name of the Sales division's `auto_home` table at Acme Inc. is as follows:

```
auto_home.org_dir.sales.acme.com.
```

Root Domain Names

Root domain names must contain two components ending with a dot.

```
acme.com.
```

use a standard inet domain name

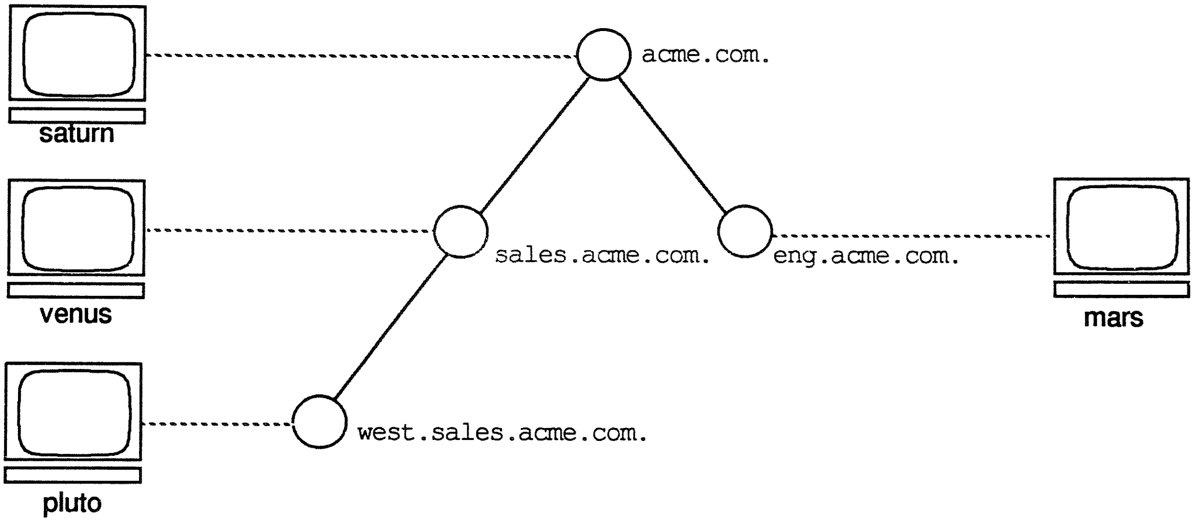
[1] NIS+ domain names are case insensitive. Thus, `Sales.Acme.` and `sales.acme.` are equivalent.

NIS+ Servers and Clients

Objects in the NIS+ namespace are stored on NIS+ servers. The servers provide information to the clients that request it. Every NIS+ domain specifies a list of servers that stores the information in its domain.

An NIS+ client belongs to an NIS+ domain. When a client is initialized, its domain name is identified and stored in the kernel. At that time, a *coldstart* file is created for the client. This file lists all the NIS+ servers that support the client's domain. When a client sends a request in its domain, it sends it to one of these supporting servers.

An NIS+ server is also a client, and it belongs to a domain as well as supports a domain. The domain the server belongs to is always above the domain it serves except when dealing with the root domain. The server supporting the root domain also belongs to the root domain.

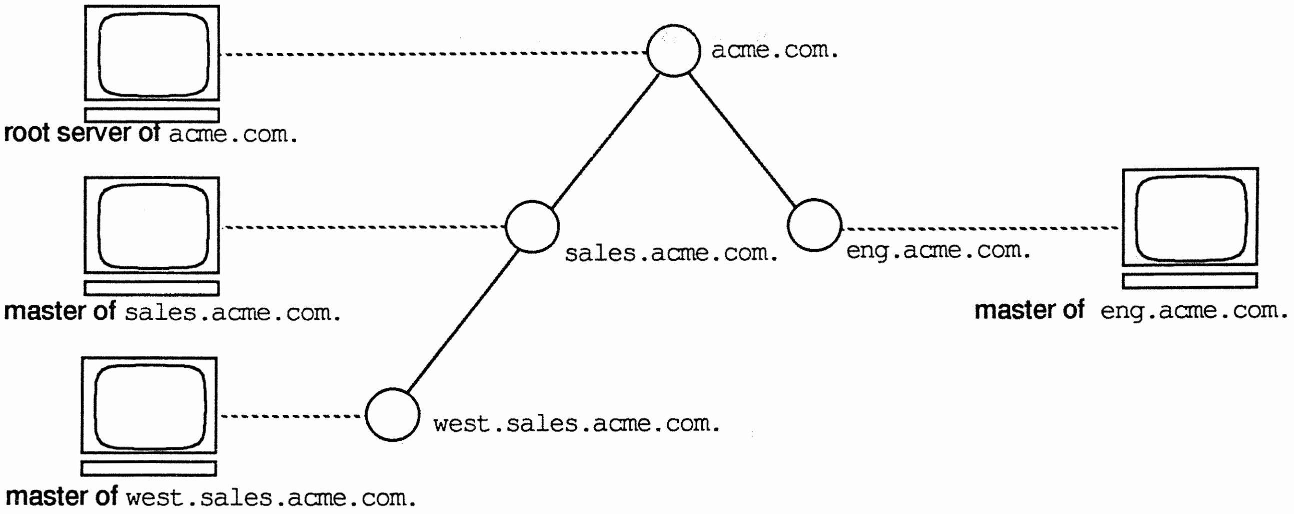


Server	Belongs To	Supports
saturn	acme.com.	acme.com.
venus	acme.com.	sales.acme.com.
mars	acme.com.	eng.acme.com.
pluto	sales.acme.com.	west.sales.acme.com.

NIS+ Master Servers and Replica Servers

For simplicity, the previous pages show each domain supported by one server. In fact, NIS+ domains are supported by one master server and one or more replica servers. At the top of the NIS+ domain is the *root server,* who is a client of the root domain because there is no domain above it. There is only one root server per domain.

namespace



Both master and replica servers store NIS+ table information and answer client requests. Only the master, however, stores the master copy of the tables. The replicas store only duplicates of the master copies.

One advantage of having replica servers is reliability. If the master server is unable to handle a request, one of the replica servers can reply. An additional advantage is efficiency. If there are many requests, they can be better handled by multiple servers.

Another advantage is simplicity for the system administrator. The administrator loads the table information in one location (the master server), and the master server propagates it to the replica servers. Similarly, updates are made to the master server, and the master server propagates the updates to the replica server.

An added benefit of using the NIS+ name service is that table information can be updated using Administration Tool from any NIS+ client (with previously defined authority and access permissions.)

NIS+ Security

NIS+ Principals

NIS+ security protects the information from unauthorized access. Access is granted only for *NIS+ principals.*

An NIS+ principal is:

- A user logged onto an NIS+ client, or
- A user logged in as root on an NIS+ client

For NIS+ security purposes, a user logged in as root is considered to be the workstation itself. *Thus, an NIS+ principal is either a user or a workstation.*

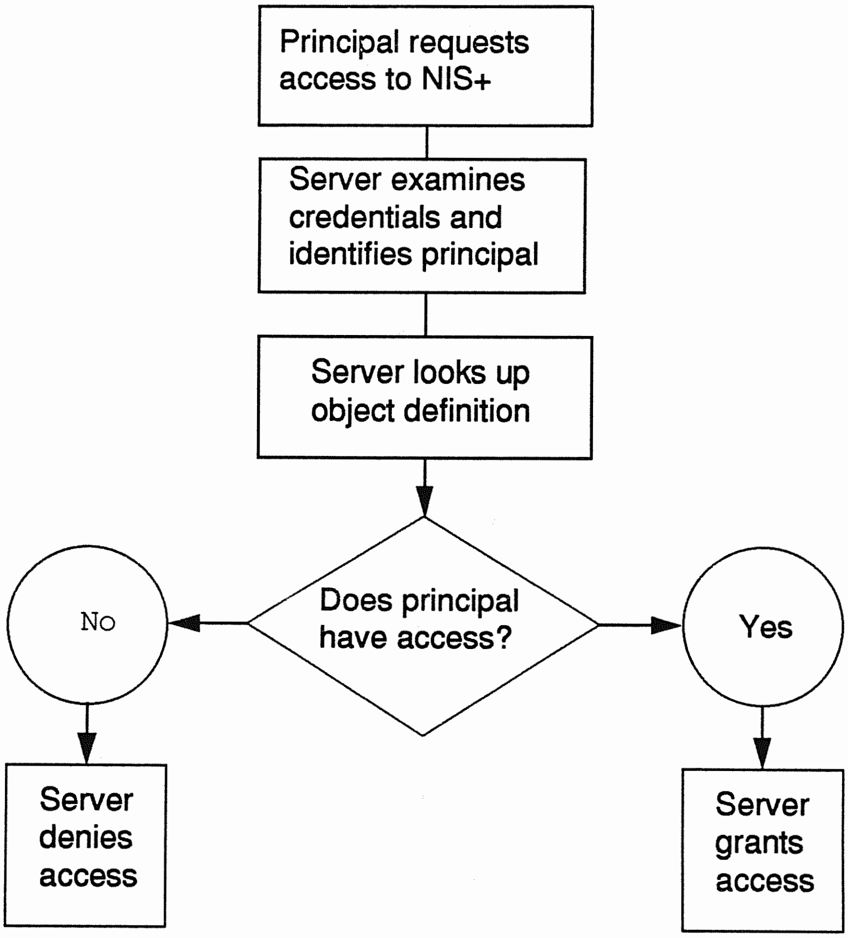
NIS+ security privileges are assigned to NIS+ principals in two stages.

1. Credentials that identify a principal are stored in a domain's `cred` table.
2. Each object in the namespace assigns access rights to different categories of NIS+ principals. This security information is stored in the object definitions.

When a principal requests access to an object, the NIS+ server finds out what access rights are assigned to that principal by that particular object. If the access rights match, the server answers the request. If they do not match, the server denies the request and returns an error message.

NIS+ Security

When an NIS+ server receives a request from an NIS+ client, it first identifies the principal. Then it finds the object the principal wants to access and determines whether the principal has proper access to that object. If the object's definition indicates that the principal has the proper access rights, the server grants it.


Principal requests access to NIS+
Server examines credentials and identifies principal
Server looks up object definition
Does principal have access?
No
Yes
Server denies access
Server grants access

The process of identifying the principal is known as *authentication*. The process of checking its access rights to an object is known as *authorization*.

NIS+ Security

Authentication

Authentication is the process of identifying the principal making the request to the NIS+ server. The purpose of authentication is to obtain the principal's name so that its access rights to an object can be looked up (the authorization process).

An NIS+ server authenticates a principal by checking its credentials. NIS+ accepts two types of credentials:

- LOCAL credentials

  LOCAL credentials are used to map a client's UID to its NIS+ principal name. They are created by extracting the client user's UID and GID from the password record and placing them in the domain's `cred` table.

- DES credentials

  DES credentials are generated by creating an additional password (or key) that is required to authenticate the principal. If this additional key is not provided, the principal is considered unauthenticated and is denied access to the object. Usually, the principal's login password and DES key are the same.

The information for authenticating NIS+ principals is stored in the `cred` table. There is one `cred` table for each NIS+ domain. It stores authentication information for the NIS+ principals that want to access that particular domain.

NIS+ Security

Authorization

NIS+ authorization is the process of granting NIS+ principals access rights to an NIS+ object. There are four types of access rights.

Access Right	Description
Read	Principal can read the contents of the object.
Modify	Principal can modify the contents of the object.
Create	Principal can create new objects in a table or directory.
Destroy	Principal can destroy objects in a table or directory.

You can think of NIS+ access rights as being similar to regular file permissions.

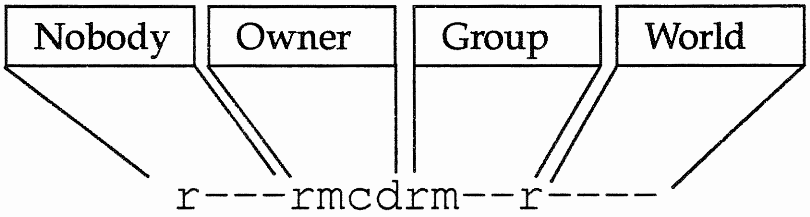
NIS+ Security

Access Rights

For the purpose of authorization and the granting of access rights, NIS+ classifies principals into four categories.

Category	Description
Owner	A single NIS+ principal
Group	A collection of NIS+ principals
World	All principals authenticated by NIS+
Nobody	Unauthenticated principals

Access rights are displayed as a list of 16 characters. These access rights are specified as part of an object's definition.



An NIS+ group is one or more NIS+ principals grouped together as a security convenience. Information about NIS+ groups is stored in NIS+ *group objects*, under the `groups_dir` subdirectory of every NIS+ domain. Note that it is *not* stored in the NIS+ `group` table—that table stores information about UNIX groups.

Access rights can be displayed using the `nisls` command, which is covered in the next lesson.

NIS+ Security Levels

The implementation of the authorization scheme described on the previous page is determined by the domain's level of security.

An NIS+ server can operate at one of three security levels: 0, 1, or 2. These security levels determine the degree to which the principal's credentials are checked.

Security Level	Description
0	No checking of the principal's credentials is done. Any client is allowed to perform any operation. This level is designed for testing and setting up the initial NIS+ namespace.
1	Checks the principal's credentials and accepts either LOCAL or DES authentication. Because LOCAL credentials are easily forged, do not use it on networks to which untrusted servers may have access.
2	Checks the principal's credentials and accepts only DES authentication. This is the highest level of security currently provided and is the default level assigned to an NIS+ server.

The NIS+ service daemon called `rpc.nisd` that runs on an NIS+ server is started from the `/etc/init.d/rpc` script. The default security level is 2.

To run the NIS+ environment at a lower security level for testing purposes, modify the `rpc` script on the root master server to include the `-S` option.

```
/usr/sbin/rpc.nisd -r -S 0
```

Name Service Switch Process

The name service switch process allows NIS+ clients to obtain information from one or more sources, such as the local /etc files or the NIS+ tables.

The /etc/nsswitch.conf file, or the *name service switch* configuration file, contains a list of 15 types of information, their sources, and the order in which these sources are searched.

The format of the file is (using several but not all examples):

```
group               source(s)
hosts               source(s)
passwd              source(s)
```

Sources

One or more sources may be specified for each database.

Source	Description
files	The client's local /etc files
nisplus	An NIS+ table
nis	An NIS map
compat	Supports old-style "+" syntax for passwd and group information
dns	Applies only to the hosts entry

+: at the end of a password file tells the program to look at NIS if it can't find the name.

+: is gone in 2.x

Name Service Switch Process

Example:

Note the following entries in the /etc/nsswitch.conf file:

```
passwd:       files nisplus
group:        files nisplus
```

This syntax says that the NIS+ passwd and group tables are searched if the information is not found in the local passwd and group files. To limit access to your system based on your local passwd and group files only, remove the nisplus references on these lines.

Name Service Switch File

Example

The following nsswitch.conf file uses the NIS+ tables and the local /etc files as its sources for information.

```
# /etc/nsswitch.nisplus:
# An example file that could be copied over to /etc/nsswitch.conf;
# it uses NIS+ (NIS Version 3) in conjuction with files.

# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file contains "switch.so" as a nametoaddr
# library for "inet" transports.

# the following two lines obviate the "+" entry in /etc/passwd
# and /etc/group
passwd:       files nisplus
group:        files nisplus

# consult /etc "files" only if nisplus is down.
hosts:        nisplus [NOTFOUND=return] files
# Uncomment the following line and comment out the above,to use
# both DNS and NIS+
# hosts: nisplus dns [NOTFOUND=return] files

services:     nisplus [NOTFOUND=return] files
networks:     nisplus [NOTFOUND=return] files
protocols:    nisplus [NOTFOUND=return] files
rpc:          nisplus [NOTFOUND=return] files
ethers:       nisplus [NOTFOUND=return] files
netmasks:     nisplus [NOTFOUND=return] files
bootparams:   nisplus [NOTFOUND=return] files

publickey:    nisplus
netgroup:     nisplus
automount:    files nisplus
aliases:      files nisplus
sendmailvars: files nisplus
```

Control the order resources are searched.

Name Service Switch Status/Action Values

Return Status

Each source returns a status code that returns a value to the user requesting NIS+ information.

Status Code	Description
`SUCCESS`	Found the requested entry
`UNAVAIL`	The source was unavailable
`NOT FOUND`	Source contains no such entry
`TRY AGAIN`	Source returned "I'm busy, try later" message

Actions

For each status code, two actions are possible.

Action	Description
`continue`	Try the next source
`return`	Stop looking for the entry

The default actions are:

- SUCCESS = return
- UNAVAIL = continue
- NOT FOUND = continue
- TRY AGAIN = continue

Name Service Switch Status/Action Values

Example:

Note the following entry in the /etc/nsswitch.conf file:

```
hosts:        nisplus [NOTFOUND=return] f iles
```

This syntax means that only the NIS+ hosts table is searched. Remove the [NOTFOUND=return] entry if you want to search the NIS+ hosts table and the local hosts file.

Name Service Switch Configuration Files

Four versions of the name service switch configuration file are included with the Solaris 2.*x* release:

- The /etc/nsswitch.conf file is the default configuration file that specifies the name service that was selected during system installation as the source to be searched for network information.
- The /etc/nsswitch.files file is an alternate name service switch file that only searches the local system's /etc files.
- The /etc/nsswitch.nis file uses the NIS database as the primary source of all information except the passwd, group, automount, and aliases maps, which use the local /etc files first and then the NIS databases. Because the search order for the passwd and group files is the local files first and then the NIS database, there is no need for a plus (+) in the passwd file.
- The /etc/nsswitch.nisplus file uses NIS+ as the primary source for all information except the passwd, group, automount, and aliases tables, which use the local /etc files first and then the NIS+ databases.

The default /etc/nsswitch.conf file is determined by what name service was selected during installation. The other switch files listed above can be copied to the /etc/nsswitch.conf file when changing your name service configuration.

Summary

In this lesson, you learned that:

- NIS+ is a service that provides information about users, workstations, and network resources.

- The NIS+ namespace is a hierarchical structure in which the NIS+ information is stored.

- An NIS+ domain consists of a directory object, an `org_dir` subdirectory, a `groups_dir` subdirectory, and a set of NIS+ tables.

- An NIS+ client is a process or machine that sends requests for information to an NIS+ server.

- The process of identifying an NIS+ principal is called authentication.

- The process of checking an NIS+ principal's access rights to an object is known as authorization.

- The NIS+ environment provides three levels of security: 0, 1, and 2.

- The name service switch process allows NIS+ clients to obtain information from one or more sources.

Exercise 1-1

Write down your answers to the following questions.

1. List the components of an NIS+ domain.

   ____

   ____

   ____

2. Identify the NIS+ directory that contains the NIS+ group information.

   ____

3. Explain the difference between authentication and authorization.

   ____

   ____

   ____

4. List the access rights used in NIS+ authorization.

   ____

   ____

   ____

   ____

5. List the types of principals used in NIS+ authentication.

   ____

   ____

   ____

   ____

6. Name the NIS+ service daemon and the script that starts it.

   ____

   ____

7. Briefly describe the name service switch process.

   ____

   ____

   ____

Configuring the NIS+ Environment 2

Objectives

Upon completion of this lesson, you will be able to:

- Configure a root master server.
- Configure a replica server.
- Configure an NIS+ client.
- Describe the function of these commands: `nisinit`, `nissetup`, `nisaddent`, `nistbladm`, `nismkdir`, `nisping`, `nisls`, `niscat`, `nismatch`, `nisgrep`, `nisdefaults`, and `nispasswd`.
- Use Administration Tool in the NIS+ environment to automatically mount a user's home directory.

References

SunOS 5.1 Administering NIS+ and DNS

SunOS 5.1 Setting Up User Accounts, Printers, and Mail,
Chapter 1, "Setting Up Users Accounts and Groups"

Introduction

This lesson introduces the commands and procedures used to configure the NIS+ environment.

It also describes the procedure for automounting a user's home directory using Administration Tool.

The manual customization of the automounter maps is covered in an earlier module.

Configuring a Root Master Server

All of the configuration steps are covered at the end of this lesson. The key steps are summarized below.

1. Use the `nisinit` command to initialize the root server.
2. Start the NIS+ server daemon at security level 0.
3. Create empty standard tables with the `nissetup` command.

The commands used are explained over the next several pages.

Initializing NIS+ Servers and Clients

The nisinit Command

The /usr/sbin/nisinit command initializes a system to be an NIS+ client or server.

Command format:

```
nisinit -r
nisinit -c -H host | -B | -C coldstart_file
```

Example:

1. This example initializes the root server of the solar.com. domain.

   Only run at initial setup.

   ```
   # nisinit -r
   This machine is in the solar.com. NIS+ domain.
   Setting up root server ...
   All done.
   #
   ```

2. This example initializes a client of the solar.com. domain.

   ```
   # nisinit -c -H venus
   This machine is in the solar.com. NIS+ domain.
   Setting up NIS+ client ...
   All done.
   #
   ```

Setting Up Standard Tables

The nissetup Command

The /usr/lib/nis/nissetup shell script creates empty versions of the standard tables in an NIS+ directory.

This command is used once per domain and only on the master server.

The domain must already exist prior to executing this command. The nissetup script creates the org_dir and groups_dir directories.

After the directories are created, it creates the default tables that NIS+ serves.

Command format:

nissetup *domainname*

Example:

```
# nissetup solar.com.
org_dir.solar.com. created
groups_dir.solar.com. created
auto_master.org_dir.solar.com. created
auto_home.org_dir.solar.com. created
bootparams.org_dir.solar.com. created
cred.org_dir.solar.com. created
ethers.org_dir.solar.com. created
group.org_dir.solar.com. created
hosts.org_dir.solar.com. created
mail_aliases.org_dir.solar.com. created
sendmailvars.org_dir.solar.com. created
netmasks.org_dir.solar.com. created
netgroup.org_dir.solar.com. created
networks.org_dir.solar.com. created
passwd.org_dir.solar.com. created
protocols.org_dir.solar.com. created
rpc.org_dir.solar.com. created
services.org_dir.solar.com. created
timezone.org_dir.solar.com. created
#
```

The nisaddent Command

The /usr/lib/nis/nisaddent command adds data into the NIS+ table by reading from a source file or standard input.

Command format:

```
nisaddent [ -r ] -f file type [ domainname ]
```

Use the -r option to replace any existing entries in the table. Use the -f option to specify the source file name.

Examples:

1. Create the passwd table from the existing /etc/passwd file using the cat command.

   ```
   # cat /etc/passwd | nisaddent passwd
   #
   ```

2. Replace the existing NIS+ hosts table contents with the contents of the hosts file.

   ```
   # nisaddent -r -f /etc/inet/hosts hosts
   #
   ```

Allows to convert your password, host, etc files to NIS+.

Configuring a Replica Server

All of the configuration steps are covered at the end of this lesson. The key steps are summarized below.

The procedure for configuring a non-root replica server is the same for replicating a root server. When replicating a root server, only the root master server runs the NIS+ daemon with the `-r` option; all the replicas simply run the NIS+ daemon with no options.

1. Specify the host as a replica server on the master server.
2. Start `rpc.nisd` on the replica server.
3. Force initial propagation of the tables to the replica server.

The nismkdir Command

The /usr/lib/nis/nismkdir command is used to create new NIS+ subdirectories (for subdomains) within an existing NIS+ domain.

This command is also needed to create the directory structure for a replica server.

Command format:

nismkdir [-s *hostname*] *directoryname*

Example:

This example creates the necessary directories for the replica server saturn on the root master in the solar.com. domain.

```
# nismkdir -s saturn solar.com.
#
# nismkdir -s saturn org_dir.solar.com.
#
```

The `nisping` Command

The `/usr/lib/nis/nisping` command is used to send a ping to all replica servers. When a replica server receives a ping, it checks with the master server for any table updates.

Command format:

`nisping [ -f ]  [ -H` *hostname* `]` *directoryname*

Use the `-f` option to force a ping even though time stamps show no reason to do so. The `-H` option is used just to specify one replica host name.

Example:

This example sends a ping of the `solar.com.` directory and the `org_dir` subdirectory to all replicas in the domain.

```
# nisping solar.com.
Pinging replicas serving directory solar.com. :
Master server is venus.solar.com.
     Last update occurred at Thu Jun 9 11:43:58 1993
Replica server is saturn.solar.com.
     Pinging ... saturn.solar.com.
# nisping org_dir.solar.com.
Pinging replicas serving directory org_dir.solar.com. :
Master server is venus.solar.com.
     Last update occurred at Thu Jun 9 11:42:59 1993
Replica server is saturn.solar.com.
     Pinging ... saturn.solar.com.
#
```

Configuring an NIS+ Client

All of the configuration steps are covered at the end of this lesson. The key steps are summarized below.

1. Set the domain name.
2. Add an `/etc/inet/hosts` entry for the master server.
3. Run the `nisinit` command.

NIS+ Client Commands

The nisls Command

The nisls command lists the objects of an NIS+ directory.

Command format:

nisls [-l] [*directory_name*]

Example:

```
# nisls -l
org_dir.solar.com.:
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:43 1994 auto_master
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:44 1994 auto_home
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:45 1994 bootparams
T r---rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:46 1994 cred
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:47 1994 ethers
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:48 1994 group
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:49 1994 hosts
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:50 1994 mail_aliases
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:51 1994 sendmailvars
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:52 1994 netmasks
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:53 1994 netgroup
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:54 1994 networks
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:55 1994 passwd
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:56 1994 protocols
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:57 1994 rpc
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:58 1994 services
T ----rmcdrmcdr--- venus.solar.com. Thu Jun 9 11:37:59 1994 timezone
#
```

The columns in the listing are:

- Type
- Permissions
- Owner's principal name
- Date of creation
- Object name

NIS+ Client Commands

The niscat Command

The niscat command displays the contents of NIS+ tables.

Command format:

niscat [-h] *tablename*

Examples:

```
# niscat passwd.org_dir
root::0:1:0000-Admin(0000):/:/sbin/sh:
daemon::1:1:0000-Admin(0000):/::
bin::2:2:0000-Admin(0000):/usr/bin::
sys::3:3:0000-Admin(0000):/::
adm::4:4:0000-Admin(0000):/var/adm::
lp::71:8:0000-lp(0000):/usr/spool/lp::
smtp::0:0:mail daemon user:/::
uucp::5:5:0000-uucp(0000):/usr/lib/uucp::
nuucp::9:9:0000-uucp(0000):/var/spool/uucppublic:
/usr/lib/uucp/uucico:
listen::37:4:Network Admin:/usr/net/nls::
nobody::60001:60001:uid no body:/: :
noaccess::60002:60002:uid no access:/::
lister::112:110:Dave Lister:/export/home/lister:/bin/sh:
rimmer::113:110:Arnold J.Rimmer:/export/home/rimmer:/bin/sh:
kryten::114:110:Kryten Model 3500:/export/home/kryten:/bin/sh:
#
```

```
# niscat -h passwd.org_dir
# name:passwd:uid:gid:gcos:home:shell:shadow
root::0:1:0000-Admin(0000):/:/sbin/sh:
daemon::1:1:0000-Admin(0000):/::
bin::2:2:0000-Admin(0000):/usr/bin::
sys::3:3:0000-Admin(0000):/::
adm::4:4:0000-Admin(0000):/var/adm::
lp::71:8:0000-lp(0000):/usr/spool/lp::
                    .
                    .
                    .
#
```

NIS+ Client Commands

The nismatch Command

The nismatch and nisgrep commands allow shell scripts to search NIS+ tables. The nisgrep commands differs from nismatch in accepting regular expressions for the search criteria rather than simple text strings.

Command format:

nismatch *key* *tablename*

Example:

```
# nismatch rimmer passwd.org_dir
rimmer::113:110:Arnold J.Rimmer:/export/home/rimmer:/bin/sh:
#
```

The nisgrep Command

Command format:

nisgrep *colname=keypat* *tablename*

Example:

```
# nisgrep 'uid=10[234]' passwd.org_dir
lister::112:110:Dave Lister:/export/home/lister:/bin/sh:
rimmer::113:110:Arnold J.Rimmer:/export/home/rimmer:/bin/sh:
kryten::114:110:Kryten Model 3500:/export/home/kryten:/bin/sh:
#
```

Changing Your NIS+ Password

The nispasswd Command

The nispasswd command changes entries in the NIS+ passwd table. The nispasswd command does not read or modify local information stored in /etc/passwd or /etc/shadow files.

It uses secure RPC to communicate with the NIS+ server, and it never sends unencrypted passwords over the network.

```
$ nispasswd
Changing password for hollie on NIS+ server.
Old login password:
New password:
Re-enter new  password:
NIS+ password information changed for hollie
NIS+ credential information changed for hollie
$
```

NIS+ Client Commands

The nisdefaults Command

The nisdefaults commands is a convenient way for administrators to display default NIS+ values such as current principal name, domain name, host name in the domain, and so on.

```
# nisdefaults
Principal Name : venus.solar.com.
Domain Name    : solar.com.
Host Name      : venus.solar.com.
Group Name     :
Access Rights  : ----rmcdr---r---
Time to live   : 12:00:00
Search Path    : solar.com.
#
```

Adding Data Into NIS+ Tables

Data can be added into the NIS+ tables in several different ways.

- Administration Tool is convenient for adding individual entries because the procedure is the same as adding entries to the local `/etc` files.
- The `nisaddent` command is the most useful way for adding bulk entries into a table from the local `/etc` files.
- The `nistbladm` command is used for adding individual entries.

Adding Bulk Table Entries

The `nisaddent` utility eliminates the task of re-entering existing name data from existing maps or files. The utility can use as a source any of the following:

- NIS maps
- `/etc` files
- NIS+ tables
- Command line

The `nisaddent` utility allows the addition of entries to an NIS+ table in several ways:

- Single entries specified from a command
- Batch entries from a file
- Batch entries from a table into a file

Using Administration Tool

Adding Single Entries to Tables

Use Administration Tool instead of NIS+ commands to browse, add, modify, or delete entries from NIS+ tables.

Automounting a User's Home Directory

The following steps describe the procedure for automounting a user's home directory using Administration Tool to update the NIS+ `auto_home` table.

1. Make sure that the user's home directory is shared on the system that contains the user's home directory.

2. Use Administration Tool to create a new user and select the AutoHome Setup check box. Or, use the Database Manager (of Administration Tool) to add an entry for an existing user to the `auto_home` table.

3. Have the user log in to a system other than the server that contains the home directory to see if the home directory is mounted.

An example of adding a new user using the `auto_home` setup is outlined on the following pages.

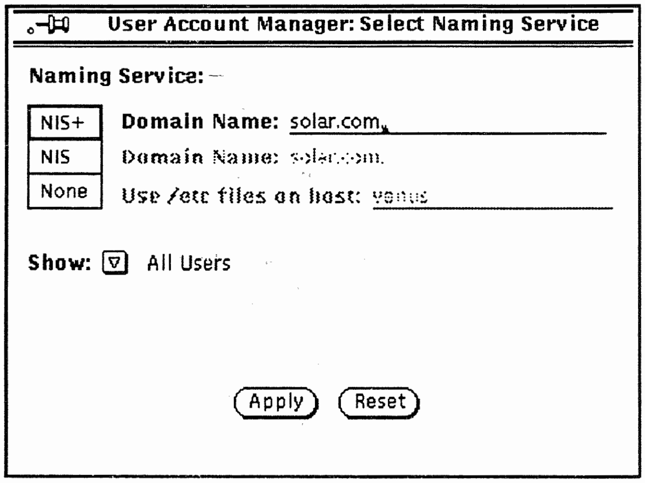
Using Administration Tool

User Account Manager

1. Start Administration Tool and click on the User Account Manager icon.

2. Set the Naming Service to the NIS+.

3. Click on Apply.



User Account Manager: Select Naming Service

Naming Service:

NIS+ Domain Name: solar.com

NIS Domain Name: solar.com.

None Use /etc files on host: venus

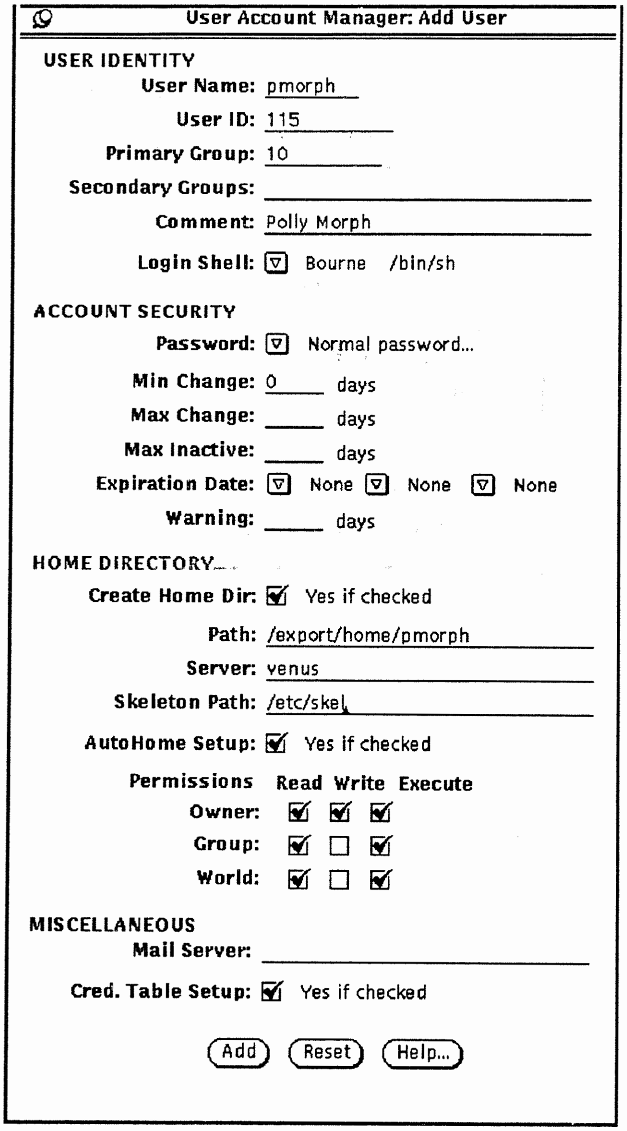
Show: ▽ All Users

Using Administration Tool

User Account Manager (continued)

4. Fill out the Add User form to create a new user and select the AutoHome Setup check box.

5. Click on Add.



User Account Manager: Add User

USER IDENTITY

User Name: pmorph

User ID: 115

Primary Group: 10

Secondary Groups:

Comment: Polly Morph

Login Shell: ▽ Bourne /bin/sh

ACCOUNT SECURITY

Password: ▽ Normal password...

Min Change: 0 days

Max Change: days

Max Inactive: days

Expiration Date: ▽ None ▽ None ▽ None

Warning: days

HOME DIRECTORY

Create Home Dir: ☑ Yes if checked

Path: /export/home/pmorph

Server: venus

Skeleton Path: /etc/skel

AutoHome Setup: ☑ Yes if checked

Permissions	Read	Write	Execute
Owner:	☑	☑	☑
Group:	☑	□	☑
World:	☑	□	☑

MISCELLANEOUS

Mail Server:

Cred. Table Setup: ☑ Yes if checked

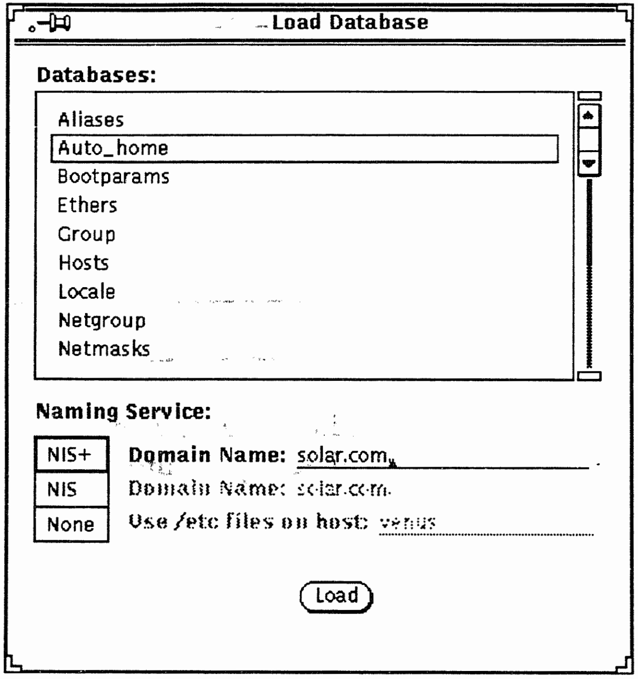
6. Dismiss the User Account Manager window.

Using Administration Tool

The Database Manager

Verify that the `auto_home` map has been updated with the user's information using the Database Manager.

7. Click on the Database Manager icon and with the Naming Service set to NIS+, click on the `Auto_home` entry.



Load Database

Databases:

- Aliases
- Auto_home
- Bootparams
- Ethers
- Group
- Hosts
- Locale
- Netgroup
- Netmasks

Naming Service:

NIS+ Domain Name: solar.com

NIS Domain Name: solar.com

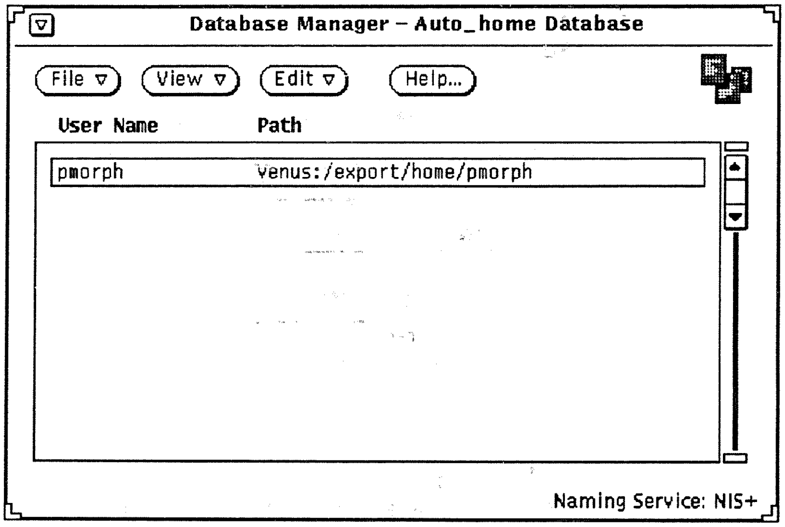
None Use /etc files on host: venus

8. Click on Load.

Using Administration Tool

The Database Manager (continued)

The auto_home table appears with the new user's entry.



User Name	Path
pmorph	venus:/export/home/pmorph

Now try logging in as the new user on another NIS+ client machine to see if the home directory is automatically mounted.

Configuring an NIS+ Root Master

1. Log in as root on the NIS+ root master.

2. Set your path to include the NIS+ administrative command directory.

```
# PATH=/usr/lib/nis:$PATH; export PATH
```

3. Set the domain name.

```
# domainname solar.com.
# domainname > /etc/defaultdomain
```

4. Change the name service switch file to include NIS+ sources.

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

5. Initialize the root master.

```
# nisinit -r
```

6. Start the NIS+ daemon with security level 0.

```
# rpc.nisd -r -S 0
```

7. Set up the NIS+ directory structure.

```
# nissetup solar.com.
```

8. Add the data file information to the NIS tables.

```
# cd /etc
# nisaddent -r -f hosts hosts
# nisaddent -r -f passwd passwd
```

9. Verify the NIS+ tables.

```
# niscat hosts.org_dir
# niscat passwd.org_dir
```

Configuring an NIS+ Root Master

10. Set the NIS_PATH variable so you won't have to use fully-qualified path names.

```
# NIS_PATH='org_dir.$:$'
# export NIS_PATH
# niscat passwd
#
```

Setting Up the NIS+ Replica Server

1. Log in to the NIS+ root master and identify the replica server.

   ```
   # rlogin master_name
   # nismkdir -s replica_hostname solar.com.
   # nismkdir -s replica_hostname org_dir.solar.com.
   ```

2. Log in as root on the NIS+ replica server.

3. Set the domain name.

   ```
   # domainname solar.com.
   # domainname > /etc/defaultdomain
   ```

4. Change the name service switch file to include NIS+.

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

5. Add the NIS+ master to the /etc/inet/hosts file.

   ```
   # vi /etc/inet/hosts
   ip_address            master_name
   ```

6. Initialize the replica server as a client first.

   ```
   # nisinit -c -H master_name
   ```

7. Start the NIS+ daemon.

   ```
   # rpc.nisd
   ```

8. On the NIS+ root master, force the propagation of tables to the replica server.

   ```
   # nisping solar.com.
   # nisping org_dir.solar.com.
   ```

Setting Up the NIS+ Client

1. Log in as root on the NIS+ client.

2. Set the domain name.

```
# domainname solar.com.
# domainname > /etc/defaultdomain
```

3. Change the name service switch file to include NIS+.

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

4. Add the NIS+ master to the /etc/inet/hosts file.

```
# vi /etc/inet/hosts
ip_address          master_name
```

5. Initialize the client.

```
# nisinit -c -H master_name
```

Summary

In this lesson, you learned that:

- The NIS+ administration commands are needed to configure a root master server, replica server, and client.
- NIS+ commands or Administration Tool can be used to browse, add, modify, or delete entries from NIS+ tables.
- A user's home directory can be automatically mounted by adding an entry to the `auto_home` map using Administration Tool.

Exercise 2-1

The purpose of this exercise is to practice using the commands needed to set up the NIS+ environment.

The tasks performed in the exercise are:

- Setting up an NIS+ root master server.
- Setting up an NIS+ replica server.
- Setting up an NIS+ client.
- Using basic NIS+ commands.

Your instructor will let you know whether this exercise will be a group activity or if you will work in teams.

Configuring an NIS+ Root Master Server

Follow the configuration steps outlined on pages 22-23.

Setting Up an NIS+ Replica Server

Follow the configuration steps outlined on page 24.

Setting Up an NIS+ Client

Follow the configuration steps outlined on page 25.

Exercise 2-2 (Optional)

The purpose of this exercise is to automount a home directory using Administration Tool to update the NIS+ `auto_home` table.

Work with the same partner from the previous NIS+ exercise, if possible.

1. On both systems: Share the `/export/home` file system.

2. On both systems: Make sure the following line is uncommented from the `/etc/auto_master` map:

   ```
   +auto_master
   ```

3. On both systems: Start the User Account Manager. Set the name service to NIS+.

4. On both systems:

   a. Use the Add User form to create a new user.

   b. Create the home directory.

   c. Set the path name to `/export/home/`*username*, and the server name to your system.

   d. Create a normal password for the user.

   e. Select the AutoHome Setup check box.

5. On both systems: Share the new user's home directory.

6. Log in as the new user on your partner's system to see if the home directory is automatically mounted.

Answer Key

Lesson 1: The NIS+ Environment

Exercise 1-1

1. A directory object, the `org_dir` subdirectory with a set of NIS+ tables, and the `groups_dir` subdirectory

2. `groups_dir` subdirectory

3. Authentication is the process of identifying NIS+ principals. Authorization is the process of granting NIS+ principals access rights to an NIS+ object.

4. Read
   Modify
   Create
   Destroy

5. Owner
   Group
   World
   Nobody

6. `rcp.nisd`
   `/etc/init.d/rpc`

7. The name service switch process allows NIS+ clients to obtain information from one or more name service sources.

Lesson 2: Configuring the NIS+ Environment

Exercise 2-1

Follow the steps as described in the exercise.

Exercise 2-2 (Optional)

Follow the steps as described in the exercise.