Concepts

About NetInfo

SA4padPointRule2.eps ¬

In this topic

1PointDashedRule3Black.eps ¬

Summary

A NetInfo network stores information about computers, users, and network resources in NetInfo *domains*. When you set up a network, you set up a hierarchy of domains.

Consider a network with a three-level domain structure. The root domain stores

information about the entire network. Midlevel domains may store information about a subset of computers, users, and resources, especially if that subset should be restricted from the rest of the network. The local domain stores information about a single computer.

Domains have *parent, child,* and *sibling* relationships.

Every NEXTSTEP computer has its own local domain, which is maintained by a database named **local.nidb** in the **/etc/netinfo** directory. Each parent domain is maintained by a database in the **/etc/netinfo** directory. (These databases are *instances* of the domain. When you create clones, you create additional instances of the domain.)

Each NetInfo database is a hierarchy of *directories*. Network information is stored as *property keys* and *values* in these directories.

You can use the Manager applications in **/NextAdmin** to set up and change a NetInfo network. Most of these applications perform specific tasksÐsuch as setting up user account information or exporting files. NetInfoManager is a general-purpose application for examining and changing the NetInfo databases directly. You can use also use the command-line utilities in **/usr/bin** to work with NetInfo domain data. The **niutil** command is the general-purpose utility.

CAUTION NetInfoManager and niutil are powerful tools for changing every detail of the network database. They provide no protections against errors. So be sure you know what you're doing when you work with these tools, and work carefully.

While a NetInfo network differs from a traditional UNIX network in some ways, many of the basics of UNIX networking apply. If you need more information about UNIX networks, be sure to refer to some of the standard UNIX texts.

892859_PointDashedRule3Black.eps ¬

The NetInfo advantage

A Netinfo network is easy to set up, maintain, and use. One system administrator can control hundreds or even thousands of computersÐan efficiency that other network administration systems can't approach.

This power results from NetInfo's unique structure and its associated tools. Unlike a traditional network database that uses table records, a NetInfo database is organized as a hierarchy of directories.

Everything about the database is extensible. You can set up multiple levels of subdirectories. You can also define the contents of these subdirectories.

At the same time, the Manager applications can automatically create a network structure as a starting place. They use a standard graphical user interface to walk you through each step. The command-line utilities give you more direct access to the contents of the database.

To take advantage of the power of a NetInfo network, you need to understand the structure of a NetInfo database, starting from the basic data, organized as properties, up to the domain level.

762711_PointDashedRule3Black.eps ¬

Properties: keys and values

Computer processes routinely query NetInfo for the data they need to function properlyÐIP addresses, user account passwords, or file system mount points, for example. This information is stored in a NetInfo domain database as a property.

A property consists of a *key* with zero or more *values*. Most keys typically have a single value, such as an Internet address or a user's name. As an example, the password for a user is stored as a value for the **passwd** key. The login name for a user is stored as a value for the **name** key.

A key may have multiple values, however, such as two or more Internet addresses or multiple options for a process. As an example the **name** key for MAILER-AGENT has two values, MAILER-AGENT and mailer-agent.

A key with no value serves as a flag: the fact that the key exists is all the information that's necessary.

Keys correspond approximately to field names in a relational database record. Values correspond to the data in one of those fields.

Properties are always stored in NetInfo directories.

791752_PointDashedRule3Black.eps ¬

Directories

A NetInfo database is a hierarchy of directories.

These are not filesystem directories. NetInfo directories are internal to the database and correspond to table records in relational databases.

The top of the directory hierarchy is the root directory. The slash (*/*) is used for the name of the root directory. The root directory contains several standard

subdirectories. Each of these subdirectories may contain several levels of subdirectories.

Every directory has at least one propertyÐthe **name** value. This property gives the directory its name. You specify a directory by its absolute pathname. For example **/user/jpeters** is the subdirectory for the user account of the person who logs in as **jpeters**.

Every directory has an ID number. The root directory has ID 0. When you create a directory, it is assigned the next sequential ID number, regardless of its place in the hierarchy. Typically, the **/machines** directory has ID 1, and one of its subdirectories may have ID 2. You can see how the database was created by inspecting the ID numbers.

Directories are stored in a domain database.

793970_PointDashedRule3Black.eps ¬

Domains

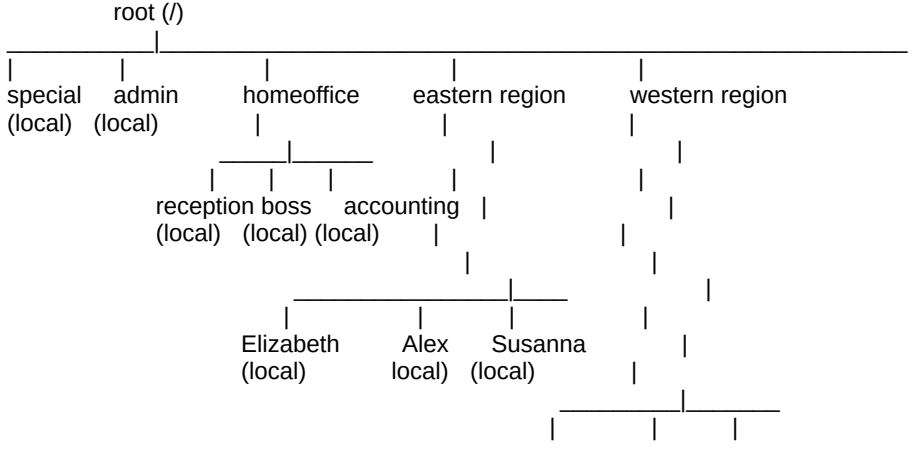
At its highest level, NetInfo is organized as a hierarchy of domainsÐwith *parent*, *child*, and *sibling* relationships.

The simplest domain structure is a two-level domain hierarchy.

52935_F59.eps ,

The upper-level domain, or **root domain**, stores information information common to all the computers on the network.

You can set up more complex structures with midlevel domains, but NeXT doesn't recommend building domain hierarchies of more than four levels.



Bill (local) Pat (local) Angus (local)

The top level is the *root domain*.
It's always a parent.

A domain that serves another domain is a child of that domain.

Domains that are all children of the same parent are siblings.

You can set up midlevel domains to store the information for departments or other divisions on a large network. Midlevel domains may have the same parentÐfor example, they may all be children of the root domain. Or they may be groupedÐsome as children of one parent, some as children of another. However, a domain can have only one immediate parent.

When a process needs network information, it looks first in the local domain for that computer. If it finds what it's looking for, it uses that. If it doesn't, it looks in the parent of the local domain. If the immediate parent doesn't have the information, the search continues up the hierarchy to the root domain. The search can't, however, extend

laterally to a sibling domain.

You can use SimpleNetworkStarter to set up a basic two-level domain structure. You can use NetInfo Manager to set up a three- or four-level domain structure.

980683_PointDashedRule3Black.eps ¬

Database files

A NetInfo domain database is a directory in the NEXTSTEP file system. It has a *tagname* and the extension **.nidb** and is stored in **/etc/netinfo**.

Every NEXTSTEP computer has the file **/etc/netinfo/local.nidb**. This is the database for the local domain. The local domain is required, and the name **local** is reserved.

In the file system, the database directory for each domain has a *collection file*, possibly with the extension ***file**. It may also have a *transaction file* and one or more *extension files*. Be careful not to delete any of these files.

The collection file contains the NetInfo directories and up to 1KB of property data for each directory. The extension files contain overflow information for directories with more than 1KB of data.

The transaction files are ephemeral, containing data being used by some process.

The contents of these files are binary. You can use the Manager applications or the command line utilities to change them. *Don't use text editors to inspect them or make changes.*

You can use NetInfoManager or niutil to examine the contents of a NetInfo domain database.

193498_PointDashedRule3Black.eps ¬

Naming conventions

Because domains exist in a hierarchy, and because NetInfo directories within a domain exist in a hierarchy, you refer to a particular domain or directory using a pathname convention. As with UNIX filenames, the . and .. characters mean "current" and "parent" respectively.

Within a domain, the root directory is always a slash (/). To refer to a NetInfo directories by its names, use a pathname convention. For example:

/machines is a subdirectory of the root directory for network

computers.

/machines/frontdesk is a subdirectory for a computer that has the name frontdesk.

You can also refer to domains by pathnames, with a slash (/) for the root domain. For example:

/frontdesk is the local domain for the computer at the front desk in a two-level network

/admin/frontdesk is the local domain for a front desk computer in a three-level domain, where /admin is a midlevel domain

The browsers in the Manager applications generally use these database names.

You can also reference a domain by the UNIX file name of its NetInfo database in **/etc/netinfo**.These are called *tags*. For example:

local is the tag for the database for the local domain of every computer. It resides in **/etc/netinfo/local.nidb**. (The name **local** is a reserved name that triggers special NetInfo activity when the computer starts up.)

network	is the tag for the root or midlevel domain database that resides in **/etc/netinfo/network.nidb**. (The name **network** is not a reserved name, but NEXT recommends that all databases for domains immediately above the local domain use the **network** tag.)

A computer gets its name and the name of its local domain from the **/machines** directory of a parent domain.

To name a computer, enter the value for the name property of the proper subdirectory, optionally followed by the tag of its local database. For example, in the **/machines/frontdesk** subdirectory of a parent domain, enter **frontdesk/local** to name a computer **frontdesk**.

To give a name to a local domain, enter the value for the serves property of the proper subdirectory, including the tag of the local database. For example, in the **/machines/frontdesk** subdirectory of a parent domain, enter **frontdesk/local** to give the name **frontdesk** to the local domain.

Note that using **frontdesk/local** for the name property treats **frontdesk** as the name

of the computer and that using **frontdesk/local** for the **serves** property treats **frontdesk** as the name of the local domain. You could give the computer a different name from its local domain, if you wish, but NeXT support *strongly* recommends against this practice.

The . and .. characters represent relative pathnames for the domains. A single dot specifies the current domain. Two dots specify the parent domain.

For example, suppose you have a midlevel domain named **network**, with a parent domain and a child domain named **frontdesk**. Its **serves** property would include the following values:

../topdog	shows that its parent domain, whatever its name, is maintained by the **/etc/netinfo/topdog.nidb** database
./network	shows that its own domain, whatever its name, is maintained by the **/etc/netinfo/network.nidb** database
frontdesk/local	shows that it has a child domain named **frontdesk** maintained by the **/etc/netinfo/local.nidb** database

380814_PointDashedRule3Black.eps ¬

NetInfo processes

Several processes use the NetInfo databases to provide basic network service:

netinfod This process constitutes a domain server. A computer typically runs one **netinfod** process for every domain it hosts. Each computer should run a **netinfod** local process. A computer hosting a parent domain for the local domain would probably run a **netinfod local** and a **netinfod network** process. A computer hosting three or four domains would run as many **netinfod** processes, each showing the database tag for its domain.

nibindd This process binds child domains to their processes. It initiates a **netinfod** process for each database in the **/etc/netinfo** directory at boot time. As **netinfod** processes broadcast requests for parents, the **nibindd** process tries to match the requests to a **netinfod** process running on the computer. If there's a match, **nibindd** passes the request. Otherwise, **nibindd** does nothing. All local domain servers issue requests for

parents. All upper-level domains issues requests for parents only if they receive requests from child domains.

lookupd This process provides a central request facility. Many processesÐincluding kernel routines, UNIX utilities, and applicationsÐmay need network and system information. They have been designed to place their requests through **lookupd**, which then queries the **netinfod** processes for the domain hierarchy, beginning with the local domain.

bootpd and **bootparamd** These processes together serve as a configuration server to respond to broadcasts from computers as they join the network. These processes are initiated by **/etc/rc** scripts at boot time if the NETMASTER setting in **/etc/hostconfig** is YES.

inetd This process is initiated early in the boot process to start other server processes on demandÐ especially multiple instances of server processes that involve **ftp**, **tcp**, or multiple **netinfod** processes.

969521_PointDashedRule3Black.eps ¬

Master servers and clones

When you set up a network, you begin by setting up a master NetInfo server for a domain to manage all the network information and NetInfo processes.

If you have more than 10 computers on your network, you should also set up clone servers that provide backup service in case the master fails. Each clone server is on a separate computer, so if one computer fails, the remaining clones are still available.

Each time you set up a clone, you set up a new instance of the network database. You can create clones when you first set up the network or add them as it grows.

The master and clone servers use **netinfod** to identify themselves. Each domain server inspects the root directory to find the value for the master property, expressed in *host/tag* notation. If the computer's name matches the hostname given for the master, the local **netinfod** process recognizes itself as the master server. If not, that netinfod server process recognizes itself as a clone server.

When you update information for a network domain, you change the master database. The master propogates the changes to the clones.

You can use SimpleNetworkStarter to set up a master server or a clone

server.

SA2PointRule2.eps ¬

Related topics (*click a* LinkDiamond.tiff ¬)
161062_SA1pt-Xref-2.eps ¬
Concepts
Ready to begin setting up your network? Not sure how many levels you need? Or how a NetInfo network works with other kinds of networks?
;../Setup/AboutSetup.rtfd;;¬ **About setup**

SA1pt-XrefDashedRuleBlack.eps ¬
How to
;OpenNetInfoDomain.rtfd;;¬ Open a NetInfo domain
;OpenNetInfoDirectory.rtfd;;¬ Open a NetInfo directory
;AddSubdirectory.rtfd;;¬ Add a NetInfo subdirectory
;AddProperty.rtfd;;¬ Add a property
;AddValue.rtfd;;¬ Add a value
;ExamineDomain.rtfd;;¬ Examine a domain
;ExamineHostEntry.rtfd;;¬ Examine a host entry
;ExamineFileServerInformation.rtfd;;¬ Examine file server information
;ExamineClientInformation.rtfd;;¬ Examine client information
;ExamineUserAccount.rtfd;;¬ Examine a user account

;ExamineGroup.rtfd;;¬ Examine a group
;ChangeDirectoryDisplay.rtfd;;¬ Change the way directories are displayed
;ChangePropertyDisplay.rtfd;;¬ Change the way properties are displayed
;ChangeNetInfoManagerPreferences.rtfd;;¬ Change NetInfoManager preferences