Using LDAP for Naming Services in AIX

Yantian Tom Lu, Ph.D.

IBM Corporation
11501 Burnet Road
Austin, TX 78758

Version 1.0
September 23, 2003

Using LDAP for Naming Services in AIX

1. Introduction

In a networking environment, each host needs information about the network and other hosts to communicate to each other. Naming service provides a centralized mechanism for hosts in a network to lookup network information. Naming service eliminates the need for each host to store such information locally and greatly reduces the effort of administering networks.

AIX® provides a variety of mechanisms to resolve names in a network. These mechanisms include DNS, NIS/NIS+, local /etc files, and LDAP. DNS provides host name/host address resolution. NIS/NIS+ provides a centralized database for services to common network information on users, groups, hosts, networks, protocols, services, rpc, etc. LDAP is a new addition to the mechanism family. It provides naming service for users, groups, and hosts using the emerging LDAP (lightweight directory access protocol) protocol.

User authentication and user and group naming resolution through LDAP have been introduced in detail in two other papers[1,2]. The LDAP user and group naming service supports RFC 2307[3] which is becoming the industry standard for LDAP naming service. However, the LDAP host naming service does not support RFC 2307.

A new NIS_LDAP mechanism, supporting RFC 2307, was implemented in AIX 5L™ version 5.2 to support naming resolution for hosts, networks, protocols, services, netgroups, and rpc. The goal of this paper is to introduce to readers what is NIS_LDAP naming service, and how to configure AIX systems to use such service.

Note: Although the name of the mechanism is NIS_LDAP, this mechanism does not use or require any NIS services or software. However if you are already using NIS or NIS+, section 4.5 of this paper explains how to migrate the naming data from NIS/NIS+ to LDAP.

2. Naming Service in AIX

AIX supports the following mechanisms for naming resolution for hosts, networks, protocols, services, netgroups, and rpc:

- **dns** - Domain Name Service. The AIX named daemon is a port of ISC BIND (Berkeley Internet Name Domain). Name servers resolve host names to Internet addresses.
- **nis** - Network Information Service.
- **nis+** - Network Information Service Plus.
- **/etc files** - local naming service. Searches the files in the /etc directory for resolving names.
- **nis_ldap** - Lightweight Directory Access Protocol. Provides naming resolution for host, networks, protocols, rpc, services, and netgroups. This mechanism works with any directory server which stores entity data using schema defined in RFC 2307.

- **ldap** - Lightweight Directory Access Protocol. Provides naming resolution for host only. This mechanism only works with directory servers which store host data using AIX specific schema. This mechanism has been deprecated and is noted here only for completeness.

An AIX system can be configured to use a combination of the above services for naming resolution. There is a sequential ordering that AIX follows to use these services, thus determining which of these services is tried first, which next. The default ordering can be overridden in several ways:

- NSORDER environment variable
- /etc/netsvc.conf configuration file
- /etc/irs.conf configuration file

2.1 NSORDER

NSORDER is an environment variable that can be used to specify the orderifor resolving host names to addresses and vise versa. NSORDER overrides the host settings in the */etc/netsvc.conf* and */etc/irs.conf* files.

Format:
```
NSORDER=value[,value]
```

Example:
```
# export NSORDER=bind,nis,local
```

In this example, the resolver will try **bind** (**dns**) first. If it fails, the resolver will try **nis**. If **nis** fails, it will try local */etc/hosts* file to resolve the name.

2.2 /etc/netsvc.conf

The */etc/netsvc.conf* file specifies the sequential order for resolving hostnames and aliases. The */etc/netsvc.conf* file overrides the default order and the order specified in the */etc/irs.conf* file.

Format:
```
hosts = value [, value]
aliases = value [, value]
```

Example:
```
hosts = nis = auth, bind, local
aliases = nis, files
```

In this example, the search order is **nis** and then **bind** and lastly **local**. *nis = auth* specifies that **nis** is authoritative - i.e., the resolver will only use **nis** for naming resolution. The **bind** service is used only when the resolver is unable to contact the **nis** service. The local */etc/hosts* file is used if **bind** fails to resolve the host. The **auth** option can only be used for the hosts keyword.

The sendmail command uses the local */etc/aliases* file by default, and uses **nis** if it is specified

for resolving aliases. The above example overrides the default.

2.3 /etc/irs.conf

The */etc/irs.conf* file controls the search order for network related data, including hosts, networks, services, protocols, and netgroups. The default order for hosts and networks is **dns (bind)**, **nisp**, **nis**, and **local**. The default order for services, protocols, and netgroups is **nis**, **local**. The order defined in /etc/irs.conf will override the default values.

Format:
```
map_type    mechanism    [option]
```

Example:
```
hosts dns continue
hosts nis continue
hosts local
networks dns continue
networks nis continue
networks local
services nis continue
services local
protocols nis continue
protocols local
netgroups nis continue
netgroups local
```

2.4 /etc/rpc.conf

The */etc/rpc.conf* file controls the order of rpc services. The default order is **nis** and **local**. The order defined in */etc/rpc.conf* will override the default values.

Format:
```
map_type    mechanism
```

Example:
```
rpc   nis_ldap
```

3. LDAP Naming Service

AIX supports a **LDAP** mechanism for user and group naming service. For information regarding use of this mechanism for user authentication and user/group management, please refer to the other two papers[1,2]. The **nis_ldap** mechanism is implemented for naming service on network information data other than users and groups. The term "naming service" used in this paper will refer only to hosts, networks, protocols, servers, netgroups, rpc, etc., but not users and groups, unless specified otherwise. These are summarized in table 1.

Table 1. LDAP naming service

Entity	Mechanism	Controlled by
Users	LDAP	User's **registry** attribute in /etc/security/user file
Groups	LDAP	Group's **registry** attribute in /etc/security/group file
Hosts	nis_ldap	NSORDER variable, /etc/netsvc.conf and /etc/irs.conf
Networks	nis_ldap	/etc/netsvc.conf and /etc/irs.conf
Protocols	nis_ldap	/etc/irs.conf
Services	nis_ldap	/etc/irs.conf
Netgrousps	nis_ldap	/etc/irs.conf
Rpc	nis_ldap	/etc/rpc.conf

AIX offers two LDAP naming services, **ldap** and **nis_ldap**. The **ldap** naming service uses the IBM specific schema and supports host name resolution only. The **nis_ldap** naming service implemented in AIX 5L v5.2 uses the RFC 2307 schema and supports name resolution of hosts, services, networks, protocols, and netgroups. This paper will only focus on the **nis_ldap** mechanism.

Of the API listed in the RFC 2307, the following are **nis_ldap** enabled:

- getservbyname ()
- getservbyport ()
- getrpcbyname ()
- gerrpcbynumber ()
- gerrpcent ()
- getprotobyname()
- getprotobynumber ()
- gethostbyname ()
- gethostbyaddr ()
- getnetbyname ()
- getentbyaddr ()
- setnetgrent ()

Many of the getxxxent() calls are not suitable for the LDAP environment, and as a result they are not **nis_ldap** enabled even though they are listed in the RFC2307 APIs:

- getpwent ()
- getspnam ()
- getspent ()
- getgrent ()
- getservent ()
- getprotoent ()

- gethostent ()
- getnetent ()

The following APIs are LDAP enabled, but are not under the control of the **nis_ldap** mechanism, rather they are under the control of the **LDAP** loadable authentication module[2]:

- getpwnam ()
- getpwuid ()
- getgrnam ()
- getgrgid ()

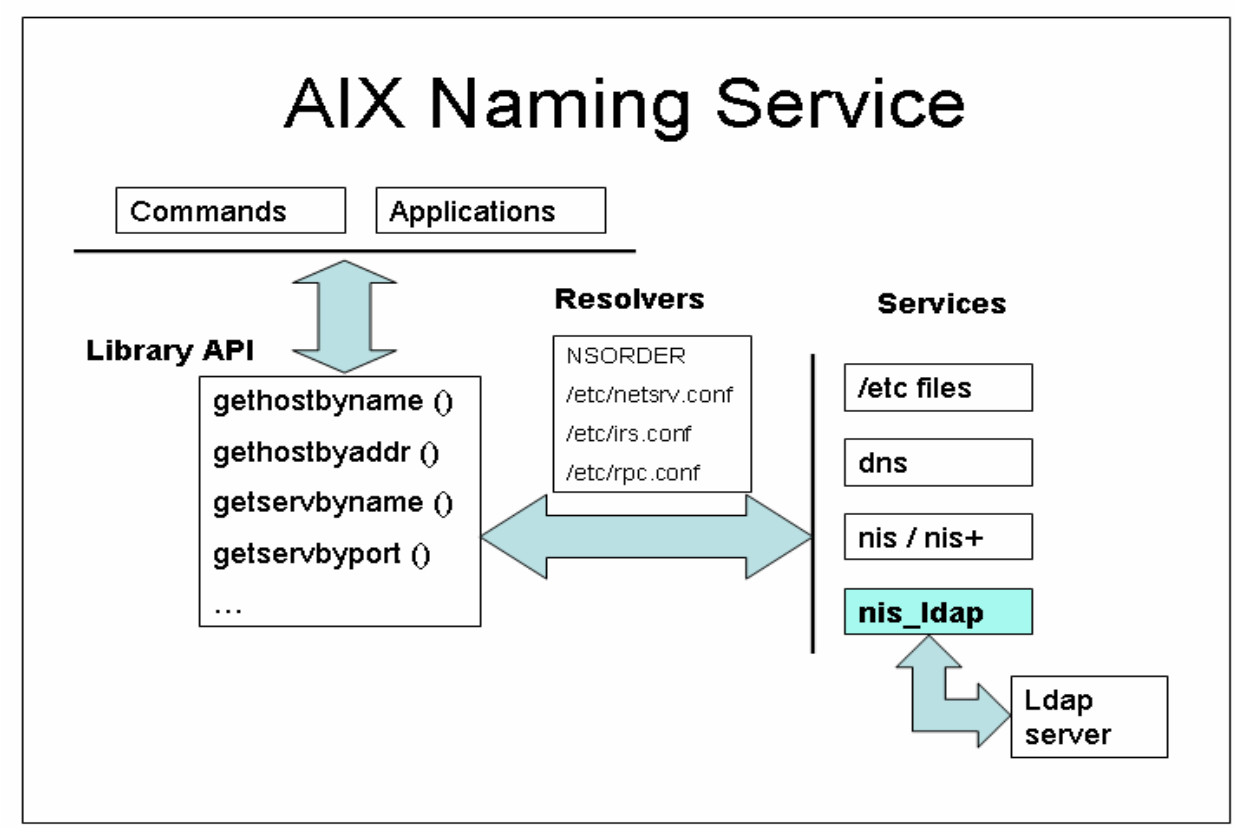

Figure 1. AIX Naming Service

4. Configuring an LDAP Server

To use the nis_ldap mechanism for naming lookup, one needs to configure an IBM Directory Server (IDS) LDAP server and populate the server with network information data. This section will address some of the issues in server configuration and data migration.

4.1 Schema

nis_ldap mechanism requires RFC 2307 schema which is also referred to as nisSchema. The nisSchema is not available by default in IBM Directory Server version 4.1 and earlier. However, a schema file containing the nisSchema is shipped as part of security component on AIX 5L v5.2 base CDs, and the file is installed automatically as part of AIX 5L v5.2 base installation. The nisSchema file is in ldif format and it can be found at */etc/security/ldap/nisSchema.ldif*. IBM Directory Server version 5.1 and later contains nisSchema by default.

The **mksecldap** command automatically updates the directory schema using the */etc/security/ldap/nisSchema.ldif* file at IBM Directory Server configuration time if **mksecldap** can not find the nisSchema in the installed directory schema. This update will only take place for IBM Directory of version 4.1 and earlier.

For server configuration with **ldapcfg** command or through the web utility, one has to add the RFC 2307 schema manually. After configuring the server and making sure the server is running, run the following command to update the schema:

```
# ldapmodify -D adminDN -w pwd -c -f /etc/security/ldap/nisSchema.ldif
```

where `adminDN` is the LDAP server administrator DN and `pwd` the password.

One can also use the ldif file to update the schema of an IBM Directory Server running on AIX 5L v5.1 or earlier which does not ship the file. To do so, run the following command from an AIX 5L v5.2 system:

```
# ldapmodify -h host -D adminDN -w pwd -c -f /etc/security/ldap/nisSchema.ldif
```

where `host` is the IBM Directory Server which is running on AIX 5L v5.1 or earlier, `adminDN` the server administrator, and `pwd` the password.

The user and group part of the nisSchema covers only a subset of the attributes needed by AIX. To support full AIX functionality, two new AIX specific objectclasses are created in AIX 5L v5.2 - **AIXAuxAccount** for users and **AIXAuxGroup** for groups. These two objectclasses contain all the attributes which are needed by AIX but not covered by the user and group part of the nisSchema.

As a result, there are two schema types for users and groups:

- **RFC2307** - posixAccount and shadowAccount for users; posixGroup for groups. This is the typical nisSchema for users and groups.
- **RFC2307AIX** - posixAccount, shadowAccount, and AIXAuxAccount for users; posixGroup and AIXAuxGroup for groups.

When specifying the schema types for the **mksecldap** command and the **nistoldif** command, only user and group entries are affected. The **RFC2307AIX** schema type supports full AIX user authentication and authorization functionalities, and also maintains RFC 2307 compatibility.

Therefore, use of **RFC2307AIX** is highly recommended. See section 4.2 for details of schema selection during IBM Directory Server configuration and data migration.

4.2 Directory Information Tree (DIT)

The default DIT root is *cn=aixdata*. The **mksecldap** command will generate a default *cn=aixdata* suffix during LDAP server setup. This default can be overridden by supplying an optional base DN to the -d option of the **mksecldap** command. The new suffix will be set to *cn=aixdata,<user-supplied-baseDN>*.

The following table shows the default DIT structure under the *cn=aixdata* parent RDN. There are two RDNs below the DIT root. The *cn=aixsecdb* RDN is generated by the **mksecldap** command, and it is the base DN for users and groups. The *cn=nisdata* RDN is generated by the **nistoldif** command, and it is the base DN for the rest of the NIS data.

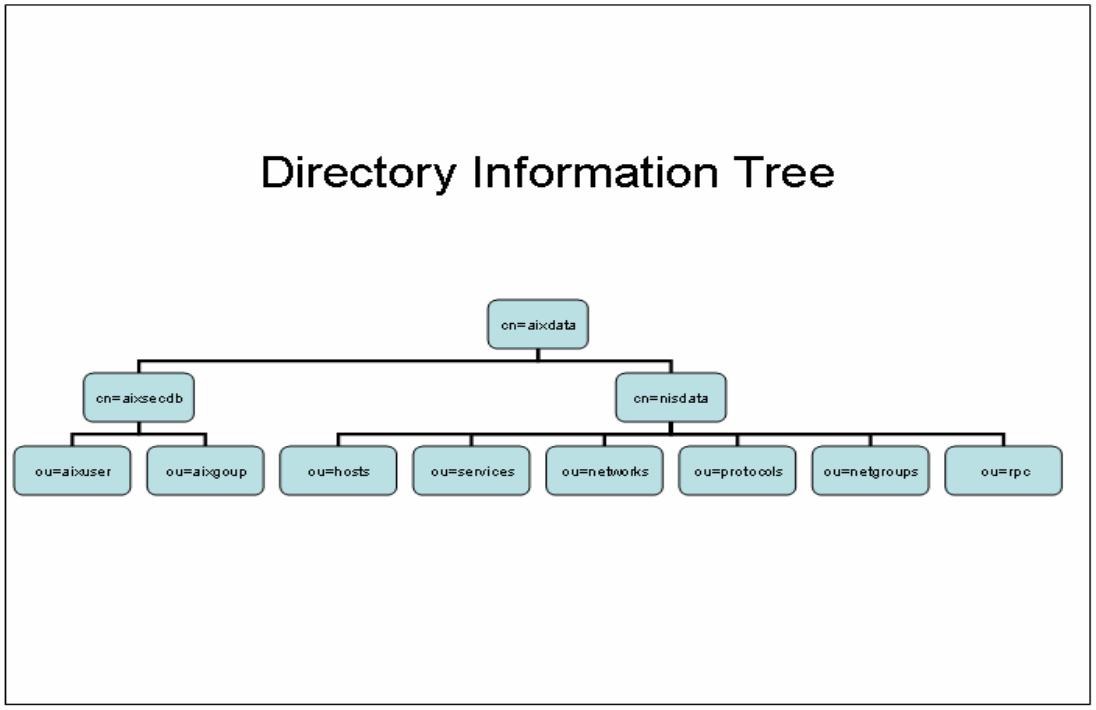

Figure 2. Directory information tree in AIX 5L v5.2

4.3 Software Requirement

AIX provides the **mksecldap** command for setting up an IBM Directory Server. The **mksecldap** command requires the *ldap.server* fileset to be installed. This fileset can be found from the base AIX CDs. The *ldap.client* fileset and the backend DB2® software are automatically installed when installing *ldap.server*. The *ldap.server* fileset is the IBM Directory Server software. The *ldap.client* fileset is the client library and utilities, and DB2 is the backend database software for IBM Directory Server software.

For SSL communication, the crypto version of the IBM Directory Server software and the GSKit software are required. The crypto filesets are *ldap.max_crypto_server* and *ldap.max_crypto_client*. IDS v4.1 requires GSKit version 5 (fileset is *gskkm.rte*), and IDS v5.1 requires GSKit v6 (fileset is *gskak.rte*). The IDS and GSKit filesets are available from the AIX expansion pack.

Table 2. Related filesets

Fileset	Description
ldap.server	Fileset for IBM Directory Server software.
ldap.client	Fileset for IBM Directory client library, header files, utilities.
ldap.max_crypto_server	Fileset for IBM Directory Server software, encryption version, required for SSL setup.
ldap.max_crypto_client	Fileset for IBM Directory client software, encryption version, required for SSL setup.
gskkm.rte	Fileset for IBM GSKit software, required for IDS v4 SSL setup.
gskak.rte	Fileset for IBM GSKit software, required for IDS v5 SSL setup.

4.4 IBM Directory Server Setup

4.4.1 Server Setup With Default Suffix

Once the IBM Directory Server software is installed, run **mksecldap** to setup the server. The **mksecldap** command creates the backend database, sets up the adminDN and password, sets up the suffix, and generates the base entries of the DIT.

```
# mksecldap -s -a cn=admin -p pwd -S rfc2307aix
```

This sets up a server with `cn=admin` as the administrator DN, `pwd` as the password, and *cn=aixdata* as the default suffix. Locally defined users and groups in /etc/ files will be exported to the server with **RFC2307AIX** type schema. The export process will not modify the local user and group data. After a successful setup, **mksecldap** starts the **slapd** server process.

Note, **mksecldap** command does not migrate any NIS data.

4.4.2 Server Setup With Given Suffix

To have **mksecldap** setup a different suffix, use the -d option.

```
# mksecldap -s -a cn=admin -p pwd -S rfc2307aix -d o=mycompany
```

mksecldap will generate the "*cn=aixdata,o=mycompany*" suffix, and migrate users and groups under it.

4.4.3 Server Setup Without Local User and Group Export

In many cases, one may not want to export users and groups from the local host to the server, since doing so could potentially "pollute" the user and group data from a NIS server. To prevent **mksecldap** migrating local users and groups to the server database, use the "-u NONE" option:

```
# mksecldap -s -a cn=admin -p pwd -S rfc2307aix -u NONE
```

This sets up a server similar to the example in section 4.4.1, with the exception that there is no user and group migration.

4.4.4 Server Setup Without "cn=aixdata"

mksecldap always creates a suffix with the "*cn=aixdata*" prefix. For example, if one wants to setup a "*dc=ibm.com*" suffix, mksecldap will prefix it with "*cn=aixdata*", and set the actual suffix as "*cn=aixdata,dc=ibm.com*".

If one wants a "*dc=ibm.com*" like suffix, it must be created manually. To do so, one would first run **mksecldap** to create the server with the default suffix, and remember to use "-u NONE" option to suppress local user and group migration:

```
# mksecldap -s -a cn=admin -p pwd -S rfc2307aix -u NONE
```

The above command will create a default "*cn=aixdata*" which is a container entry. Once the configuration is successfully done, edit the */etc/slapd32.conf* server configuration file to add the new preferred suffix. Find the line:

```
ibm-slapdSuffix: cn=aixdata
```

and insert the following line below:

```
ibm-slapdSuffix: dc=ibm.com
```

Save the file. The new suffix requires that the slapd server process be restarted to take effect.

```
# kill `ps -e | awk '{ print $1 }'`
# /usr/bin/slapd
```

4.5 Data Migration

4.5.1 The nistoldif Tool

The **nistoldif** command reads the passwd, group, host, service, protocol, rpc, network, and netgroup maps from a NIS server and exports the data to the LDAP server (requires that the *ldap.client* fileset be installed locally). It can optionally export the maps to stdout in ldif format, which can be redirected to a file, examined, and added to the LDAP server with the **ldapadd**

command. If a map is not available, **nistoldif** uses the corresponding file in the /etc directory instead. With the -s flag, **nistoldif** can be used to migrate selected NIS maps or files.

AIX supports a variety of user and group attributes which are not covered by the nisSchema. To exploit these attributes, use of the **RFC2307AIX** schema type is highly recommended. However, the **nistoldif** tool only exports basic user and group attributes, i.e., the nisSchema attributes. Use of the **RFC2307AIX** schema type simply adds the AIX extended schema objectclass to user and group entries, which allows later definition of the AIX specific attributes. The schema type only affects export of user and group entries, not other NIS entities.

The syntax of the command is:

```
nistoldif -d BaseDN [ -a BindDN -h Host -p BindPasswd [-n Port ] ] [ -f
      Directory ] [ -y Domain ] [ -S Schema ] [ -k SSLKeyPath -w SSLKeyPasswd
      ] [ -s Maps ]
```

Flags:

-a BindDN	Specifies the administrative bind DN used to connect to the LDAP server.
-d BaseDN	Specifies the suffix that the data should be added under.
-f Directory	Specifies the directory to look for flat files in. If this flag is not used, nistoldif will look for flat files in /etc.
-h Host	Specifies the host name which is running the LDAP server. If this flag is used, -a and -p must also be used, and data will be written directly to the LDAP server.
-k SSLKeyPath	Specifies the SSL key path.
-n Port	Specifies the port to connect to the LDAP server on.
-p BindPasswd	Specifies the password used to connect to the LDAP server
-s Maps	Specifies a set of maps to be exported. This flag requires a list of letters representing the maps that should be migrated. If this flag is not used, all maps will be migrated. The letters are: **e** for netgroup, **g** for group, **h** for hosts, **n** for networks, **p** for protocols, **r** for rpc, **s** for services, and **u** for passwd.
-S Schema	Specifies the LDAP schema to use for users and groups. This can be either RFC2307 or RFC2307AIX; RFC2307AIX gives extended AIX schema support. Default is RFC2307.
-w SSLKeyPasswd	Specifies the SSL password.
-y Domain	Specifies the NIS domain to read maps from. If this flag is not used, the default domain will be used.

Examples

1. To export data of the default domain under the “cn=aixdata” base DN and use RFC2307AIX schema type for users and groups, and redirect the output to the nis.ldif file, type:

```
nistoldif -d cn=aixdata –S rfc2307aix > nis.ldif
```

2. To export the NIS maps of the domain "austin.ibm.com" under the "cn=aixdata" base DN, type:

   ```
   nistoldif -d cn=aixdata -y austin.ibm.com –S rfc2307aix > nis.ldif
   ```

3. To export the NIS data from /tmp/etc files under the "cn=aixdata" base DN, type:

   ```
   nistoldif -d cn=aixdata –S rfc2307aix -f /tmp/etc > nis.ldif
   ```

4. To export only the hosts and services maps from the default domain and add the data directly to a directory server under the "cn=aixdata" base DN, type:

   ```
   nistoldif -s hs -d cn=aixdata -h ldap.austin.ibm.com -a cn=root -p
   secret
   ```

   where `ldap.austin.ibm.com` is the directory host, `cn=root` the bind DN, and `secret` the bind password.

4.5.2 Indirect Migration

Once the server is setup, the next step is the NIS data migration. The **nistoldif** command provided by AIX is for this purpose. The **nistoldif** command reads the NIS maps or files and converts the data into ldif format. One can save the ldif output to a file, and then add the data using the **ldapadd** command. For example,

```
# nistoldif -d cn=aixdata -S rfc2307aix > /etc/security/ldap/nis.ldif
# ldapadd -D cn=admin -w pwd -f /etc/security/ldap/nis.ldif
```

To add the data to a directory server running on another host:

```
# ldapadd -h serverhost -D cn=admin -w pwd -f /etc/security/ldap/nis.ldif
```

Where `serverhost` is the hostname of the LDAP server, `cn=admin` is the LDAP administrator DN, and `pwd` the password. Use of SSL is recommended for better security during the data transfer over the network.

When redirecting the ldif output to a file, make sure to save the output file to a secure location. Since the data contains sensitive information like user password, no one except privileged users should read the file.

Since users and groups defined in NIS maps have only attributes which are covered by the RFC2307 schema, use of the **RFC2307AIX** schema type by **nistoldif** simply adds the **aixAuxAccount** objectclass for user entries and **AIXAuxGroup** objectclass for group entries. Additional attributes stored in other files, e.g., */etc/security/user*, are not exported. However, the additional AIX schema allows addition of extra AIX specific attributes in the future.

4.5.3 Direct Migration

The **nistoldif** tool also supports direct data migration to the LDAP server. The mode requires that the *ldap.client* fileset be installed. To do so, simply supply the hostname of the LDAP server, the admin DN, and the password to the command line:

```
# nistoldif -h ldapserver.mycompany.com -a cn=admin -p pwd -d cn=aixdata -S
      RFC2307AIX
```

In this case, **nistoldif** migrates all NIS data from the local host and directly adds the data to the directory server. Note, the directory server must be configured and running.

With direct migration from multiple sources or hosts, the **nistoldif** command checks the numeric id for users and groups. If a conflict is found during data loading, e.g., user foo from host A and user bar from host B have the same numeric id, **nistoldif** will not load the second entry to the LDAP server.

When loading ldif data with indirect migration, however, the **ldif2db** and **ldapadd** commands do not do integration checks for the numeric user id and group id. When loading data from multiple hosts, there is a high chance that id conflicts exist, and it is recommended one use direct migration in such case.

The indirect migration mechanism does provide a way to manipulate the data. One can save the ldif output to a file, view the data structure, modify any data of the file, and then add the data to the LDAP server.

Please note that future releases of IBM Directory Server may support a unique attribute feature which solves exactly the kind of problem that was discussed above on duplicate ids. With the unique attribute feature, one can add additional users and groups without worrying about duplicating ids, for the server will reject these entries for you.

4.5.4. Migrating Data from NIS+

The **nistoldif** command cannot directly export data from NIS+ server. The data must first be extracted from the tables using the **nisaddent** command:

```
# /usr/lib/nis/nisaddent -d [ -t table ] tabletype > /mydir/filename
```

where `table` is the name of the NIS table, `tabletype` is one of passwd, passwd, group, hosts, networks, protocols, services, rpc, or netgroup. The `table` is always <tabletype>.org_dir, except shadow which is passwd.org_dir.

These files must be placed under the same directory and they must have the same name as the files in the */etc* directory, e.g., the data from the *hosts.org_dir* file must be dumped into a file named *hosts*. Run the **nistoldif** command on the directory to export all the data to a file which can be added to the LDAP server.

```
# nistoldif -d cn=aixdata –S rfc2307aix -f /mydir > nis.ldif
```

For more information, please refer to the AIX documentation on NIS/NIS+ [3].

4.5.5 Multiple Domains

Multiple domains can be dealt with two ways. The first is one-to-one migration - one NIS domain per set of LDAP servers (master and the replicas, or peers). The second is many-to-one migration - multiple domains to a single set of LDAP servers. The one-to-one model works best for organizations whose units are geographically separated. While the many-to-one model suits smaller organizations where data flow is less extensive. Which model to choose depends on an organization's geographic distribution and intensity of data traffic. The general role of thumb is to use your current NIS model to set up domains in LDAP directory.

One-to-one

The procedures described earlier in this paper apply to one-to-one setup. One needs to configure a LDAP server for each domain and migrate data from each domain to the corresponding LDAP server. Setup replica servers afterwards.

Many-to-one

Create the first domain, e.g., the cn=aixdata,ou=humanResouce,o=mycompany,c=us, as described earlier, and then follow these steps:

1. Edit the */etc/slapd32.conf* file. Find the first domain suffix, and append additional domain suffixes which are to be migrated

   ```
   ibm-slapdSuffix : cn=aixdata,ou=humanResouce,o=mycompany,c=us
   ibm-slapdSuffix : cn=aixdata,ou=finance,o=mycompany,c=us
   ```

   Save the file.
2. Restart the LDAP server.
3. From the second NIS domain host, run the following command to export the NIS data to the LDAP server:

   ```
   # nistoldif -h ldapserver -a adminDN -p pwd -d
       cn=aixdata,ou=finance,o=mycompany,c=us -S rfc2307aix
   ```

   where ldapserver is the host name of the LDAP server, adminDN the LDAP server administrator DN, pwd the administrator bind password, cn=aixdata,ou=finance,o=mycompany,c=us the new domain to be created, and rfc2307aix the schema type used.

Repeat this step for each additional domain.

It is also possible to create one suffix, and multiple subtrees under the suffix. Each subtree will hold data for a domain. Run step 3 above for each domain to migrate the data.

5. Configuring An AIX Client to Use NIS_LDAP Mechanism

5.1 Client Setup

The LDAP naming service in AIX 5L v5.2 is **nis_ldap**. **nis_ldap** provides services for naming resolution on hosts, networks, services, protocols, rpc, and netgroups. AIX offers a **ldap** service for host name resolution only, and its use is deprecated. Use of the newer **nis_ldap** service is recommended.

Configuring an AIX client for LDAP naming resolution can be done with the **mksecldap** command. This command was created to configure a client for user authentication through LDAP in earlier versions of AIX, and it is expanded in AIX 5L v5.2 for configuring LDAP naming resolution also.

For example:

```
# mksecldap -c -h serverhost -a cn=admin -p pwd
```

This makes **mksecldap** to setup a system as a LDAP client to the `serverhost` LDAP server. **mksecldap** saves the admin DN, the password, and the searched baseDNs to *the /etc/security/ldap/ldap.cfg* file; finds the right base DNs for users, groups, hosts, services, protocols, networks, netgroups, and rpc objects; sets the naming resolver to **nis_ldap** in the */etc/netsvc.conf* and */etc/irs.conf* files; and finally starts the **secldapclntd** client daemon.

If the LDAP server contains data from multiple domains under different subtrees, one must use the -d option to instruct **mksecldap** to find the base DNs from the right subtree. Please see section 5.2 for more details.

This configuration step will work with IBM Directory Server as well as other LDAP servers whose NIS data conform to the RFC 2307 standard.

Many features regarding the **secldapclntd** client daemon have been discussed in the LDAP client configuration paper[1] and they are not repeated here. These features include multiple server support, failover mechanism, and other system behaviors. Please read the paper for more information on these features. In the following, we only address what is relevant to **nis_ldap** naming service and what is not covered by that paper.

5.1.1 Base DNs

The **mksecldap** command searches the server database during client configuration for the base DNs of users, groups, hosts, services, networks, protocols, and rpc. The base DNs are then saved to the */etc/security/ldap/ldap.cfg* client configuration file. The search for base DN is done by searching for an entry which contains the following signature objectclasses.

Table 3. NIS entities and the corresponding objectclasses

Entity	Objectclass
users	posixAccount
groups	posixGroup
Hosts	ipHost
Services	ipServices
Networks	ipNetwork
Protocols	ipProtocol
Netgroups	nisNetgroup
Rpc	oncRpc

One inherited behavior of the **mksecldap** command is that client setup fails if neither the user base DN nor the group base DN is found from the server. However, **mksecldap** command does not hard fail if it fails to find base DNs for any other entities. If a base DN of the latter group is not found, **mksecldap** simply ignores it and the setup process continues.

All base DNs found are recorded to the */etc/security/ldap/ldap.cfg* file. One may double check that the DNs are correct by checking the file after **mksecldap** finishes. Included below is part of a ldap.cfg file which contains the base DNs:

```
# Base DN where the user and group data are stored in the LDAP server.
# e.g., if user foo's DN is: username=foo,ou=aixuser,cn=aixsecdb
# then the user base DN is: ou=aixuser,cn=aixsecdb
userbasedn:ou=aixuser,cn=aixsecdb,cn=aixdata
groupbasedn:ou=aixgroup,cn=aixsecdb,cn=aixdata
idbasedn:cn=aixid,ou=system,cn=aixsecdb,cn=aixdata
hostbasedn:ou=hosts,cn=nisdata,cn=aixdata
servicebasedn:ou=services,cn=nisdata,cn=aixdata
protocolbasedn:ou=protocols,cn=nisdata,cn=aixdata
networkbasedn:ou=networks,cn=nisdata,cn=aixdata
netgroupbasedn:ou=netgroup,cn=nisdata,cn=aixdata
rpcbasedn:ou=rpc,cn=nisdata,cn=aixdata
```

5.1.2 Resolver Configuration

For every base DN found, **mksecldap** makes **nis_ldap** the preferred mechanism for the corresponding naming service. For example, if **mksecldap** finds the base DN for hosts, it will make **nis_ldap** the preferred host naming resolution mechanism, and save the resolving order in the */etc/netsvc.conf* file (for hosts), the */etc/irs.conf* file (for services, networks, protocols, and netgroups), and */etc/rpc.conf* (for rpc).

The following bullets describe the behavior when **mksecldap** updates the */etc/netsvc.conf*, */etc/irs.conf*, and the */etc/rpc.conf* files as part of client configuration:

- If a configuration files does not exist, **mksecldap** creates the file, and make **nis_ldap** the preferred resolving mechanism followed by the default order of mechanisms. For example, if */etc/irs.conf* does not exist before running **mksecldap**, **mksecldap** creates this file during client configuration, and adds the found service to the file. For example:

  ```
  protocols nis_ldap continue
  protocols nis continue
  protocols local
  ```

- If a configuration file does exist, **mksecldap** makes **nis_ldap** the first mechanism followed by the original ordering. For example, the */etc/netsvc.conf* file has the following before running mksecldap:

  ```
  hosts = local, nis
  ```

  mksecldap inserts **nis_ldap** at the beginning of the list to make **nis_ldap** the first host naming resolution mechanism:

  ```
  hosts = nis_ldap, local, nis
  ```

- **mksecldap** updates a resolution ordering only when it finds the base DN for that mechanism from the ldap server. For example, if **mksecldap** command finds only the base DN for hosts and networks from the ldap server, it only updates the hosts and networks entries in the */etc/netsvc.conf* and */etc/irs.conf* files while leaving other entries (protocols, rpc, services, etc) untouched.

If you decide not to use the **nis_ldap** mechanism for a service, edit the corresponding resolver file and remove the **nis_ldap** from it for that service.

Users and groups are different in that their resolution is not controlled by the */etc/netsvc.conf* and */etc/irs.conf* file, rather it is controlled by the /etc/security/user file. To resolve user "foo" through LDAP, the administrator needs to run:

```
# chuser -R LDAP SYSTEM=LDAP registry=LDAP foo
```

This will enable user foo to authenticate through LDAP, and make foo's account information be resolved through the **LDAP** mechanism. For more information in this respect, readers are recommended to read the LDAP client configuration paper[1].

5.1.3 Known Issues

5.1.3.1 Deadlock: Attempting to use nis_ldap on the LDAP server

There is a known problem in enabling **nis_ldap** for host naming resolution on the LDAP server that may cause the **slapd** server process and the **secldapclntd** client daemon process to deadlock. The **secldapclntd** daemon sends a request to the server for host name resolution, the server, in responding to the request, needs to do a host lookup, which in turn is done through the

secldapclntd daemon. This deadlock occurs only on a system where the IDS server is running and which itself is configured to use **nis_ldap** mechanism for host name resolution.

To avoid the deadlock, **mksecldap** will not enable **nis_ldap** for hosts naming resolution on the LDAP server. However, one may want to check the */etc/netsvc.conf* and */etc/irs.conf* to make sure that this is not the cause for a possible server hang.

This deadlock occurs in all current versions of IDS (version 5.1 and earlier). A formal fix to this deadlock problem is expected to be resolved through IDS efixes.

5.1.3.2 TCP/IP Could Not Start With nis_ldap For Service

With **nis_ldap** mechanism for service lookup, TCP/IP daemons may not start at system boot, and users, including root, may not be able to login to the system. A simple workaround is to comment out **nis_ldap** from the services entry of the */etc/irs.conf* file. Applying APAR IY48219 shall fix the problem.

5.2 Multiple Domains

If a LDAP server hosts multiple domains, setting up a client system using **mksecldap** will require the -d option for a specific domain. Otherwise, the result is unspecified - the client system can end up talking to any one of the domains. For example, the LDAP server has two domains, with each domain in a different subtree:

```
ou=humanResource,o=mycompany
ou=finance,o=mycompany
```

To configure a system to be a client of `the ou=humanRecource,c=mycompany` domain, run the following command:

```
# mksecldap -c -h server -a cn=admin -p pwd -d ou=humanResource,c=mycompany
```

To configure a system to be client of the `ou=finance,c=mycompany` domain, run the following command:

```
# mksecldap -c -h server -a cn=admin -p pwd -d ou=finance,c=mycompany
```

One can only configure an AIX system to be a member of one domain. AIX does not support concurrent multiple domains.

5.3 Caching

Client caching is done by the **secldapclntd** daemon for user and group entities as of AIX 5L v5.2. Caching for other entries may be done in a later release. Cache size and cache TTL (time to live) can be configured by modifying the corresponding entries in the */etc/security/ldap/ldap.cfg* file. The default size is 1000 entries for user and100 for group, and default TTL is 300 seconds.

```
# Number of user cache entries. Valid value is 100 - 10000 entries.
# Default is 1000.
#usercachesize: 1000

# Number of group cache entries. Valid value is 10 - 1000 entries.
# Default is 100.
 #groupcachesize: 100

# Cache timeout value in seconds. Valid value is 60 - 60*60
# seconds. Default is 300. Set to 0 to disable caching
#cachetimeout: 300
```

5.4 Enabling nis_ldap Naming Resolution

Configuring a client with the **mksecldap** command updates resolver files with nis_ldap service and automatically enables nis_ldap for name lookups.

User and groups require additional configuration, as the system administrator has to enable user authentication and lookup through LDAP using the user's **SYSTEM** and **registry** attribute. For more information, please refer to the LDAP client configuration paper[1].

5.6 User Application

Applications which make calls to the **nis_ldap** enabled API (see section 3), will get the network information from LDAP as long as the resolver is set to **nis_ldap**

User and group lookup is not as straight forward. As has been described, user and group lookup through LDAP is controlled by the */etc/security/user* and the */etc/security/group* file[1]. Such control is at per user/group level using the **registry** attribute. While AIX commands will correctly interpret a user's **registry** attribute and work with the correct user registry, a direct call by an user application to an API like *getpwnam()* may not find the correct user. If the user is unique throughout all registries (e.g., local, LDAP, DCE, NIS, etc.) enabled for the system, *getpwnam()* will find the user. However, if the user is not unique, there is no guarantee that information returned by *getpwnam()* will be from LDAP.

Applications should call *getuserattr()* routine for a user's registry first and then call *setauthdb()* to set the registry, so that subsequent library calls can be routed to the correct user registry. The *setauthdb()* call will control user lookup as well as group lookup. Reset the value by calling *setauthdb()* again using the saved value from the previous call to *setauthdb()*. The following library routines are affected:

```
authenticate
ckuseracct
ckuserID
chpass
cuserid
getpwnam
getpwnam_r
getpwuid
getpwuid_r
```

```
getgrgid
getgrgid_r
getgrnam
getgrnam_r
getgrset
getpcred
getuserattr
getuserpw
getgroupattr
IDtouser
IDtogroup
initgroups
loginrestrictions
newpass
passwdexpired
putuserattr
putgroupattr
putuserpw
putuserpwhist
setpcred
```

5.6 Adding/Deleting/Modifying Entries

AIX system commands are enabled for managing LDAP users and groups. Those commands include **mkuser**, **chuser**, **lsuser**, **passwd**, **mkgroup**, **chgroup**, **lsgroup**, etc. All of these commands support a "-R LDAP" option to work on LDAP users and groups. For example, to create a LDAP user with the **mkuser** command:

```
# mkuser -R LDAP foo
```

For detailed information on managing LDAP users and groups, refer to the LDAP client configuration paper[1].

There are no other corresponding AIX commands for managing other NIS entries in LDAP. To add other NIS map entries, follow these steps:

1. Create a working directory, e.g., */tmp/nis*.
2. Create a file with the appropriate name for your entry type:

   /tmp/nis/hosts for host entries

   /tmp/nis/services for service entries

   /tmp/nis/protocols for protocol entries

   /tmp/nis/networks for network entries

   /tmp/nis/netgroups for netgroup entries

   /tmp/nis/rpc for rpc entries
3. Edit the file, adding your new entries. Refer to the corresponding files in the /etc directory for proper format. Save the file.

4. Repeat step 2 and 3 for any new entry types.
5. Run the **nistoldif** command to add the new entries to the server.

```
# nistoldif -h server -a bindDN -p pwd -d cn=aixdata -f /tmp/nis
```

Depending on the suffix that your server uses, you may need to replace the default suffix "*cn=aixdata*" with the actual suffix in step 5, e.g., "*cn=aixdata,<o=mycompany,c=us>*". Otherwise, the new entries may not end up in the correct place of your DIT.

Run **ldapsearch** command to check if the entries are added to the server successfully. Once confirmed, delete the no-longer needed files in the /tmp/nis directory.

One can also create entries in LDIF format, and add the data file using the **ldapadd** command. Refer to Appendix for example entries. Other LDAP commands can also help in managing your entries. These include **ldapmodify**, **ldapsearch**, and **ldapdelete**. Another option is to use the IBM Directory Web Administration tool for management tasks. For more information, please consult the IBM Directory Administration Guide.

Appendix

Entry Examples

```
cn=pecten.ibm.com+ipHostNumber=9.53.84.58,ou=hosts,cn=nisdata,cn=aixdata
objectclass=top
objectclass=ipHost
objectclass=device
iphostnumber=9.53.84.58
cn=pecten.ibm.com

dn:
cn=tcpmux+ipServicePort=1+ipServiceProtocol=tcp,ou=services,cn=nisdata,cn=ai
xdata
objectClass: top
objectClass: ipService
ipServicePort: 1
ipServiceProtocol: tcp
cn: tcpmux

dn: cn=portmapper,ou=rpc,cn=nisdata,cn=aixdata
cn: portmapper
cn: portmap
cn: sunrpc
objectClass: top
objectClass: oncRpc
oncRpcNumber: 100000
description: description

dn: cn=ip,ou=protocols,cn=nisdata,cn=aixdata
cn: ip
```

```
cn: IP
objectClass: top
objectClass: ipProtocol
ipProtocolNumber: 0
description: description

dn: cn=net5+ipNetworkNumber=10,ou=networks,cn=nisdata,cn=aixdata
cn: net5
cn: net5aliase
objectClass: top
objectClass: ipNetworks
ipNetworkNumber: 10

dn: cn=group3,ou=netgroup,cn=nisdata,cn=aixdata
cn: group3
objectClass: top
objectClass: nisNetgroup
nisNetgrupTriple: (-,foo,mydomain),(-,bar,mydomain)
```

References

1. Configuring an AIX Client System for User Authentication and Management Through LDAP. *http://www-1.ibm.com/servers/aix/whitepapers/ldap_client.html*
2. Configuring an IBM Directory Server for User Authentication and Management in AIX. *http://www-1.ibm.com/servers/aix/whitepapers/ldap_server.html*
3. Appendix B. Migrating from NIS and NIS+ to RFC 2307-compliant LDAP services, Network Information Services (NIS and NIS+) Guide.
4. AIX v4.3 Documentation: System Management Guide: Operating System and Devices: LDAP Exploitation of the Security Subsystem. *http://publib.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/baseadmn/toc.htm*
5. Configuring the AIX security subsystem to use IBM's SecureWay Directory (LDAP). *http://www.ibm.com/servers/aix/products/aixos/whitepapers/ldap.html*
6. RFC 2307: An approach for using LDAP as a network information service. *http://www.ietf.org/rfc/rfc2307.txt*
7. IBM Directory Server Version 4.1: Administration Guide. *http://www.ibm.com/software/network/directory/library/#4*
8. AIX 5L Version 5.2 Security Guide: LDAP exploitation of the Security Subsystem. *http://publib16.boulder.ibm.com/pseries/en_US/aixbman/security/securitytfrm.htm*
9. Planning and Configuring NIS_LDAP Name Resolution (RFC 2307 schema), System Management Guide: Communications and Networks.

IBM Corporation
Marketing Communications
Systems Group
Route 100
Somers, New York 10589

Produced in the United States of America
09-03