Running Multiple Solaris™ Operating Environment Naming Services on a Client

By Tom Bialaski - Enterprise Engineering

Sun BluePrints™ OnLine - May 2001

http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 816-0713-10
Revision 01, 05/08/01
Edition: May 2001

Running Multiple Solaris™ Operating Environment Naming Services on a Client

Since Solaris™ Operating Environment (Solaris OE) native LDAP is intended to replace your current NIS or NIS+ naming service, the client installation assumes you will not be running them together. The reason is that both LDAP directories and NIS maps are used to store identical system related data. Maintaining this data in two unrelated data stores is not a recommended practice because the data can become inconsistent if the data stores are not synchronized.

However, during your transition from NIS to LDAP, you may choose to run both naming services in parallel. While this is possible, the procedure for doing so is not well documented. There are also several items to watch out for when running multiple Solaris OE naming services concurrently.

This article provides a procedure for running NIS and LDAP clients on the same system and highlights best practices for doing so. In addition, procedures for backing out either naming service and restoring a previous one are provided.

What Makes a Client an LDAP Client

Solaris 8 OE includes all the software necessary to run a system as an LDAP client. The activation of this software is really quite simple. All that is required is the creation of two ASCII configuration files and a modification to the `/etc/nsswitch.conf` file. The two configuration files are:

- `/var/ldap/ldap_client_file`
- `/var/ldap/ldap_client_cred`

These two files are created automatically when the `ldapclient` command, used to initialize a native LDAP client, is executed. The modifications to `nsswitch.conf`, which includes support for LDAP as a naming service, are contained in the `nsswitch.ldap` template file. This template file automatically replaces your current `nsswitch.conf` file during client initialization. To complete the installation, a system reboot is required which starts the LDAP cache manager, (`ldap_cachemgr`) with the `/etc/rc2.d/S71ldap.client` script. The cache manager is not required, but is recommended to boost performance.

While running `ldapclient` is the recommended procedure for initialization, a Solaris OE LDAP client can be configured manually if you already have created `ldap_client_file` and `ldap_client_cred` configuration files. However, there are some caveats to be aware of if `ldapclient` is not used. To understand these caveats, the next section walks you through the steps performed behind the scenes by the `ldapclient` command.

LDAP Client Initialization

An LDAP client can be initialized either at installation time or after a client is already configured and running another naming service. A future article will discuss the mechanics for using the JumpStart™ technology to automatically initialize a Solaris OE LDAP client. In this article, the assumption is that the client is already running NIS.

The `ldapclient` Command

The recommended procedure for initializing an LDAP client is to execute the `ldapclient` command on the Solaris OE client. The command is typically run with a client *profile* specified as a command line argument. With this method, a profile containing several configuration parameters is first created using the `ldap_gen_profile` command, then the profile is installed on the LDAP directory server supporting the clients. The alternative is to supply all the necessary configuration parameters on the command line when the `ldapclient` command is run. The manual page for `ldapclient` lists all the available options.

The advantage of using profiles is that fewer parameters need to be entered on the command line and the configuration information contained in the profile can be cached on the client and later updated automatically as the data changes on the LDAP directory server. The cache manager running on the client periodically checks the directory server to see if the profile data has been updated. If it has, a new version of the profile is cached and stored, and the configuration files residing in `/var/ldap` are updated.

The following sequence of events occurs when a client is initialized with the profile download mechanism.

1. The `ldapclient` command performs a search on the LDAP directory server for an entry containing a `nisdomain` attribute matching the client's domain name which is obtained either from the value returned by the `domainname` command or from the `-d` argument if supplied.
2. If the search is successful, the `ldapclient` command performs a search for a profile object that matches the name specified with the `-P` argument on the command line.
3. If the `ldapclient` command finds the specified profile object, the client downloads it.
4. The client parses the information in the profile, then uses it to create the `ldap_client_file` and `ldap_client_cred` files, which reside in the `/var/ldap` directory.
5. The client backs up the following NIS related files:
   a. `/etc/nsswitch.conf` -> `/etc/nsswitch.orig`
   b. `/etc/defaultdomain` -> `/etc/defaultdomain.orig`
   c. `/var/yp/binding/domain` -> `/var/yp/binding/domain.orig`
6. The client halts the name service cache daemon (`nscd`).
7. The client halts the `ypbind` daemon.
8. The client copies the `/etc/nsswitch.ldap` template file to `/etc/nsswitch.conf`.
9. The client creates a new `/etc/defaultdomain` file if a new domain name is specified.
10. The client requests the user to reboot the system.
11. Upon reboot, the client starts `/usr/lib/ldap/ldap_cachemgr` with the `S71ldap.client` script located in `/etc/rc2.d`.

The previous sequence assumes you are using the `pam_unix` module for authentication. Since `pam_unix` is the default authentication method, no additional configuration is required. If `pam_ldap` is used for authentication, then the `/etc/pam.conf` file needs to be modified as described later in this article.

LDAP Client Initialization Files

This section examines the files created or referenced during initialization. These files are:

1. *myprofile*.ldif – output from the ldap_gen_profile command.
2. ldap_client_file – configuration file generated from information residing in the client profile specified with the ldapclient -P argument or from command line arguments if the -i option is used.
3. ldap_client_cred – configuration file that contains the bindDN and password the client uses to bind to the directory server with. This information is gathered from the client profile or command line arguments.
4. nsswitch.ldap – preconfigured version of the nsswitch.conf file that specifies LDAP as the primary naming service.

Client Profile

The ldap_gen_profile command produces an LDIF file based on command line arguments which can be imported into your LDAP directory server. In the next example, the following arguments are specified:

- -P This is the name given to the profile you are creating.
- -b The search base the client uses to start looking for naming service data.
- -D The Distinguished Name (DN) with which the client binds to the directory server.

- `-w` The password associated with the `bindDN`.

```
# ldap_gen_profile -P myprofile -b dc=bprus,dc=com -D
"cn=proxyagent,ou=profile,dc=bprus,dc=com" -w mysecret
129.148.181.130 > myprofile.ldif
# cat myprofile.ldif

dn: cn=myprofile,ou=profile,dc=bprus,dc=com
SolarisBindDN: cn=proxyagent,ou=profile,dc=bprus,dc=com
SolarisBindPassword: {NS1}c2ab886b43e612a6
SolarisLDAPServers: 129.148.181.130
SolarisSearchBaseDN: dc=bprus,dc=com
SolarisAuthMethod: NS_LDAP_AUTH_NONE
SolarisTransportSecurity: NS_LDAP_SEC_NONE
SolarisSearchReferral: NS_LDAP_FOLLOWREF
SolarisSearchScope: NS_LDAP_SCOPE_ONELEVEL
SolarisSearchTimeLimit: 30
SolarisCacheTTL: 43200
cn: myprofile
ObjectClass: top
ObjectClass: SolarisNamingProfile
```

The output of the command is redirected to a file called *myprofile*`.ldif`, which is imported into the directory server. After a client is initialized with the `ldapclient` command using the `-P` option, the following two files are created:

- `/var/ldap/ldap_client_file`
- `/var/ldap/ldap_client_cred`

ldap_client_file

The following is an example of the resulting `ldap_client_file` file.

```
#
# Do not edit this file manually; your changes will be lost. Please
use ldapclient (1M) instead.
#
NS_LDAP_FILE_VERSION= 1.0
NS_LDAP_SERVERS= 129.148.181.130
NS_LDAP_SEARCH_BASEDN= dc=blueprints,dc=com
NS_LDAP_AUTH= NS_LDAP_AUTH_SIMPLE
NS_LDAP_TRANSPORT_SEC= NS_LDAP_SEC_NONE
NS_LDAP_SEARCH_REF= NS_LDAP_FOLLOWREF
NS_LDAP_DOMAIN= blueprints.com
NS_LDAP_EXP= 984609458
NS_LDAP_SEARCH_SCOPE= NS_LDAP_SCOPE_ONELEVEL
NS_LDAP_SEARCH_TIME= 30
NS_LDAP_PROFILE= myprofile
```

Most of the information contained in the profile is placed here. The two missing pieces of information are placed in the `ldap_client_cred` file as shown in the next example.

ldap_client_cred

The following is an example of the resulting `ldap_client_cred` file.

```
#
# Do not edit this file manually; your changes will be lost. Please
use ldapclient (1M) instead.
#
NS_LDAP_BINDDN= cn=proxyagent,ou=profile,dc=blueprints,dc=com
NS_LDAP_BINDPASSWD= {NS1}4a3788e8c053424f
```

Notice that the password is stored in `{NS1}` hashing format to prevent unauthorized people from reading it. The `{NS1}` string is produced by a two way hash algorithm and is decoded into clear text before it is sent over the wire.

A warning appears in both the `ldap_client_file` and `ldap_client_cred` files that states that you should not manually edit these files. The reason for the warning is that the LDAP cache manager will overwrite any changes if profiles are implemented. If you do not use profiles, the data will not be overwritten.

nsswitch.ldap

A template file is provided for use with LDAP specified as your primary naming service. The following is an example of the `nsswitch.ldap` template file.

```
bpsrus# more nsswitch.ldap
passwd:     files ldap
group:      files ldap

# consult /etc "files" only if ldap is down.
hosts:      ldap [NOTFOUND=return] files
ipnodes:    files

networks:   ldap [NOTFOUND=return] files
protocols:  ldap [NOTFOUND=return] files
rpc:        ldap [NOTFOUND=return] files
ethers:     ldap [NOTFOUND=return] files
netmasks:   ldap [NOTFOUND=return] files
bootparams: ldap [NOTFOUND=return] files
publickey:  ldap [NOTFOUND=return] files

netgroup:   ldap

automount:  files ldap
aliases:    files ldap

# for efficient getservbyname() avoid ldap
services:   files ldap
sendmailvars:   files

# role-based access control
auth_attr: files ldap
exec_attr: files ldap
prof_attr: files ldap
user_attr: files ldap

# audit
audit_user: files ldap
```

This template file is similar to the `nsswitch.nis` file, which is used when NIS is specified, except the `ldap` keyword is used in place of `nis`. For testing purposes you want to remove the `[NOTFOUND=return]`tag on the `hosts` line. The LDAP service is considered unavailable only if the server fails to respond to a bind request from the client. However, if the client cannot attempt a bind due to a client configuration error, the service is not considered down. In this case, `files` would never be searched resulting in the inability of the client to reach important hosts.

Enabling NIS Client Support

When the native LDAP client is initialized, the NIS naming service is disabled by renaming the directory in `/var/yp/binding` that matches your NIS domain name and by killing the `ypbind` process. After the native LDAP client is initialized, NIS can be enabled by reversing that process. The steps to do this are as follows:

1. Rename the directory in `/var/yp/binding` back to the original name.
2. Start the `ypbind` daemon.
3. Modify the `nsswitch.conf` file.

The following is an example of commands to do this.

```
# cd /var/yp/binding
# mv mynisdomain.orig mynisdomain
# cd /usr/lib/netsvc/yp
# ./ypbind -broadcast
# ypwhich
nisserver
# vi /etc/nsswitch.conf (add nis tag to database paths)
```

In this example, the broadcast method of locating NIS servers is used. If you have a `ypservers` file in your `/var/yp/binding/`*`mydomain`* directory, you do not have to specify the `-broadcast` option with `ypbind`. However, the host or hosts specified in `ypservers` must appear in a `hosts` database path specified in your `/etc/nsswitch.conf` file.

Note – If your NIS domain name is different than the `nisdomain` attribute value you are referencing on your directory server, you need to reset your domain name before running `ypbind`. You can still run the LDAP client with a different domain set, but the `ldap_cachemgr` process will generate an error message.

Once `ypbind` is running, you can modify the `/etc/nsswitch.conf` file to include the `nis` keyword. The following is a sample section of a modified `nsswitch.conf` file to illustrate this point.

```
bpsrus# more nsswitch.ldap
passwd:     files ldap nis
group:      files ldap nis

# consult /etc "files" only if ldap is down.
hosts:      files ldap nis
```

You can place the naming service keywords in any order. However, for testing purposes, specify `files` first because it is not dependent on any naming service being operational.

If you are storing passwords in a format other than crypt, you need to include the `pam_ldap` module in the client's `/etc/pam.conf` file. This is because `pam_unix` can only deal with passwords stored in crypt format. You should always stack `pam_ldap` below `pam_unix` as shown in the following code sample.

```
bpsrus# more /etc/pam.conf
#
#ident  "@(#)pam.conf   1.14    99/09/16 SMI"
#
# Copyright (c) 1996-1999, Sun Microsystems, Inc.
# All Rights Reserved.
#
# PAM configuration
#
# Authentication management
#
login auth sufficient /usr/lib/security/pam_unix.so.1
login auth required /usr/lib/security/pam_ldap.so.1 \
try_first_pass
#
```

Doing this allows user accounts stored in `files` and `nis` to be authenticated in addition to those stored in an LDAP directory. This is because `pam_ldap` can only access the LDAP directory for authentication. By specifying `pam_ldap` with the `try_first_pass` option below `pam_unix`, users are not prompted for passwords twice if their password is not stored in crypt format.

Backing Out the LDAP Naming Service

After native LDAP support is configured, it can easily be disabled. This can be performed manually or by running the `ldapclient -u` command. However, the `-u` argument assumes you were previously running a naming service like NIS or NIS+ and attempts to restore it. The `ldapclient -u` command does this by renaming the `*.orig` files, created when the LDAP client was initialized, to the original file names. If the system is configured to run both native LDAP and NIS, do not use the `ldapclient -u` command to back out native LDAP support because you already have NIS enabled.

Note – If you run the `ldapclient` command to initialize a client successive times without uninitialzing it with the `ldapclient -u` command first, the `nsswitch.conf.org` file is overwritten with the LDAP version. If this happens, the `ldapclient -u` command will not be able to restore the original `nsswitch.conf` file. You will have to do that file manually.

To manually disable native LDAP, perform these steps:

1. Rename (or delete) the `ldap_client_file` and `ldap_client_cred` files located in `/var/ldap`.
2. Remove the `ldap` tag in `/etc/nsswitch.conf`.
3. Reboot, or stop `ldap_cachemgr` by issuing the following command:

```
# /etc/init.d/ldap.client stop
```

Testing Tips and Best Practices

When setting up a native LDAP client for testing, observe the following rules:

1. Always specify `pam_unix` in the `pam.conf` authentication management section. If you do not, you cannot authenticate using the local root account.
2. Turn off the name service cache daemon while testing by using the following command:

```
# /etc/init.d/nscd stop
```

If the daemon is running, the client uses cached data instead of obtaining it directly from the LDAP directory. This can cause confusion because you won't see the results you expect.

3. Disable the LDAP cache manager to eliminate distracting error messages with the following command:

```
# /etc/init.d/ldap.client stop
```

If your NIS domain name does not match the name assigned to the `nisdomain` attribute on your LDAP directory server, `ldap_cachemgr` generates error messages that warn you to correct this situation. These messages can be prevented by disabling `ldap_cachemgr`.

Note – With `ldap_cachemgr` disabled, you cannot run the `ldaplist` command (which searches and lists naming information) without logging in as `root`. Without access to the cache, the `ldaplist` command attempts to read the `ldap_client_cred` file which is only readable by `root`.

---

4. Use the `passwd -r` command, specifying either `nis` or `ldap`, to change passwords if both `nis` and `ldap` are specified on the `passwd` line in `/etc/nsswitch.conf`. You cannot use the `passwd` command without the `-r` argument in this configuration.

Conclusion

While not intended to run along side another naming service, there is nothing to prevent native LDAP coexistence with another naming service, like NIS, on the same client. There are two ways to set this up.

1. On a client, already running NIS, configure native LDAP with `ldapclient`, then reactivate NIS manually.
2. Manually set up native LDAP on a client by copying the necessary configuration files to `/var/ldap` and modifying `/etc/nsswitch.conf`.

This article described how to deploy each method. Either one will work although you should initialize at least one client with `ldapclient` to create the required `ldap_client_file` and `ldap_client_cred` files.

Running two naming services, which contain the same data, on a client is generally not a good practice. However, for testing purposes or if you do not want to move all your NIS maps to LDAP at the same time, running NIS and native LDAP together is possible.

Author's Bio: Tom Bialaski

Tom Bialaski is currently a Senior Staff Engineer with the Enterprise Engineering group at Sun Microsystems, and is the author of "Solaris Guide for Windows NT Administrators," and co-author of "Solaris and LDAP: Naming Services." Tom has 20 years of experience with the UNIX® operating system and has been a Sun Engineer since 1984.