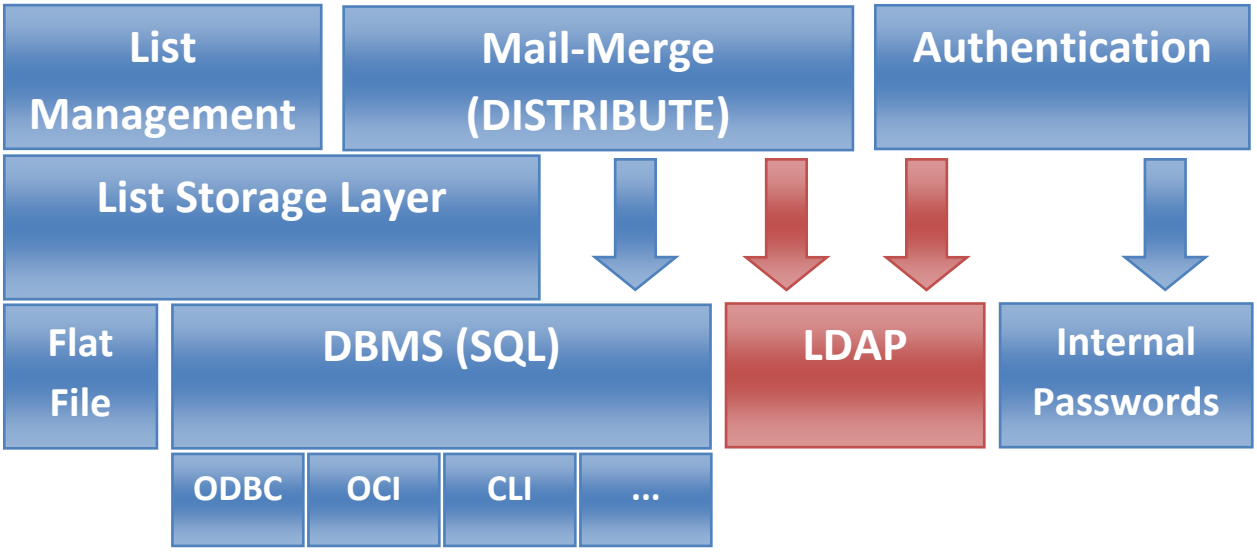
LISTSERV LDAP Documentation

28 November 2007

Overview

LISTSERV version 15.5 can interface to LDAP servers to authenticate user logins, to insert LDAP attributes in mail-merge distributions as well as to implement *Dynamic Queries,* a new feature described in a separate document.

The following diagram shows the LISTSERV 15.5 LDAP architecture, in relation to other components:



For clarity, *Dynamic Query* functions have been omitted from the diagram, but they also interface with the new LDAP functionality.

The LDAP interface is at the same level as the DBMS interface – *not* at the level of the vendor-specific SQL drivers. Quite simply, LDAP servers do not "speak" SQL. To support LDAP, we had to teach the mail-merge and authentication modules to "speak" LDAP. This is why there is a new syntax for every LDAP-related function.

At this point, LISTSERV only queries LDAP directories. It will never try to make any changes, so it should not be given write access to the directory.

Configuring LDAP in LISTSERV

The first step in using LDAP with LISTSERV is to add one or more LDAP servers in the LISTSERV site configuration. This can be done via the LISTSERV web administration interface (the preferred method), or alternately by adding the entries manually to SITE.CFG or 'go'.

Each LDAP server is given a nickname in the LISTSERV configuration, similarly to DBMS data sources. You can also configure one unnamed LDAP server, again like with DBMS data sources, but it is probably less confusing to assign a nickname to every LDAP server.

Three configuration variables must be defined for every LDAP server:

- **LDAP_SERVER_*nickname*=**hostname[:port]
  The hostname and optional port of the LDAP server. The exact format depends on your operating system and LDAP library; LISTSERV passes this string to the LDAP library as it is. On unix, SSL encryption is requested by prepending 'ldaps://' to the hostname. On Windows, the 'ldaps://' prefix is not available, but setting the port to 636 automatically requests SSL.
- **LDAP_UID_*nickname*=**userid
  LDAP_AUTH_*nickname*=password
  The userid and password that LISTSERV should use in order to login to the LDAP server. The exact format of the userid depends on your LDAP server. LISTSERV does not attempt to parse or reformat these variables. If the password is the empty string, most LDAP servers will perform an anonymous login. If both userid and password are the empty string, LISTSERV will attempt a default login, as defined by the LDAP library for your operating system. Under Windows, LISTSERV will be logged in with its current domain credentials (assuming it is connecting to an Active Directory server), and this usually provides sufficient access – try it before configuring a userid and password.

If the LDAP server is to be used to authenticate LISTSERV users, the following variables must also be defined:

- **LDAP_PW_BASE_*nickname*=**DN
  The 'distinguished name' that should be the 'base' for searches when LISTSERV looks for a user account (see below for an explanation of the authentication process). This can be used to restrict LISTSERV access to a particular organizational unit within the enterprise. If omitted, LISTSERV tries to guess the DN that will admit any Active Directory Windows account, but this is a difficult guess to make, and of course you may not even be connecting to Active Directory.
- **LDAP_PW_FILTER_*nickname*=**filter
  The LDAP 'filter' that should be used when looking up user accounts (if this filter returns at least one entry, LISTSERV allows the user to try and log in; otherwise, the login is rejected, even if the user would otherwise be able to log in to the LDAP server with the supplied credentials). Any occurrences of '%s' are replaced with the user's full e-mail address, while '%u' expands to just the userid and '%h' expands to the hostname. If omitted, LISTSERV uses a filter that is suitable for most Active Directory installations.

In addition, the following optional variables can be defined:

- **LDAP_DEFAULT_EMAIL_*nickname*=**attribute
  The name of the attribute that ordinarily specifies a user's e-mail address in this directory. This is used as a default value in searches and can be overridden. If omitted, it defaults to 'mail' (suitable for Active Directory).
- **LDAP_DEFAULT_NAME_*nickname*=**attribute
  The name of the attribute that ordinarily contains the user's full name. Defaults to 'name'.

Using LDAP for mail-merge

Because of its complex, machine-friendly syntax, LDAP is primarily suited for scripting. While it is relatively easy for a programmer to write a script that sends a weekly notice to every member of a particular department, it is not realistic to expect ordinary list owners or end-users to understand the intricacies of LDAP and devise working search filters. For instance, to select all users in an Exchange database, one would have to use the following filter:

(&(!(Alias=$null))(|(&(ObjectCategory=person)(ObjectClass=user)(Database=$null)(ServerLegacyDN=$null))(&(ObjectCategory=person)(ObjectClass=user)(!(Database=$null))(!(ServerLegacyDN=$null)))))

L-Soft expects that LDAP-based distributions will be created by customer-developed scripts – either intranet web scripts or traditional 'cron' jobs or scheduled tasks. At this point, there are no plans to provide a web interface page into which raw LDAP search filters could be entered.

To create an LDAP-based distribution, a script uses the DISTRIBUTE command and specifies an LDAP keyword as follows:

DISTRIBUTE … LDAP=YES(SERVER=nickname,E-MAIL=attribute,PARTS=attribute)

The syntax of this keyword is essentially the same as for SQL-based distributions ("DBMS="):

- **SERVER=*nickname*** identifies the LDAP server to be queried. If omitted, the default (unnamed) LDAP server is used.
- **E-MAIL=*attribute*** identifies the name of the directory attribute containing the recipient's e-mail address. If omitted, the value of LDAP_DEFAULT_EMAIL_*nickname* is used.
- **PARTS=*attribute*** is the name of an optional directory attribute containing a list of message parts that the recipient subscribes to. Although this mail-merge feature is unlikely to be used with LDAP, it is available if desired.

Similarly to SQL-based distributions, the 'TO' DD contains a list of LDAP search statements, rather than a list of actual recipients. Each line in the 'TO' DD can be one of the following statements:

- **BASE** DN
  The 'distinguished name' of the 'base' of the LDAP search. Mandatory.

- **FILTER** search_filter
  The LDAP search filter for the search. Mandatory.
- **ATTRS** attr1 [attr2 […]]
  A list of directory attributes of interest (used in the mail-merge). If omitted, all directory attributes are made available. Attribute names are not case-sensitive. The main purpose of this statement is to improve search performance if there are many irrelevant attributes in the directory. Note that the E-MAIL and (if enabled) PARTS attributes must be specified or the distribution will fail.
- **SCOPE** BASE|ONELEVEL|SUBTREE
  Optionally changes the scope of the search from the default (SUBTREE).
- **SEARCH**
  Starts the search. This command allows multiple LDAP searches to be performed in the same distribution. If there is only one search, this command is optional – LISTSERV automatically starts the search when it reaches the end of the 'TO' DD.

For instance, this search will select all Windows users in the EXAMPLE.COM domain with a valid e-mail address:

BASE CN=Users,DC=EXAMPLE,DC=COM
FILTER (&(objectcategory=person)(objectclass=user))
ATTRS Name Mail Phone
SEARCH

Using LDAP for authentication

LISTSERV can be configured to use one or several LDAP servers for authentication (user login). You can choose to allow users without an LDAP account to log in with an internal LISTSERV password, or to restrict access to users with an LDAP account.

LDAP authentication is enabled by defining the following configuration variables:

- **LDAP_PW_SERVERS=**nickname1 [nickname2 […]]
  The list of LDAP servers to be queried (in the specified order) for user accounts. Be sure to enter server nicknames, not hostnames.
- **LDAP_PW_ONLY=**0 or 1 (default: 0)
  If set to 1, only users with an LDAP account are allowed to log in to LISTSERV; other users will only be able to access LISTSERV anonymously. ***Make sure to test your LDAP settings before enabling this option, or you will not be able to undo it from the web interface!*** Enabling this option on a server that previously had external users is likely to result in significant confusion for the external users, whose passwords will no longer work.
- **LDAP_PW_REQUIRE_SSL=**0 or 1 (default: 1)
  Whether or not LISTSERV should accept LDAP passwords transmitted to the web interface in plain text. By default, LISTSERV will only attempt to verify passwords transmitted over SSL. Note

that this option does *not* control LISTSERV's own use of SSL when communicating with the LDAP server. See the LDAP_SERVER_*nickname* variable.

- **SIGNUP_REQUIRE_SSL=**0 or 1 (default: 0)
  Similar to the above, but affects all LISTSERV passwords, whether LDAP or internal. Can be used without enabling LDAP authentication.

The LDAP authentication process

When LDAP is enabled, LISTSERV goes through the following steps to log in a user:

1. The servers listed in LDAP_PW_SERVERS are examined in turn, in the order in which they were listed. For each server, LISTSERV executes the search configured with the LDAP_PW_BASE_*nickname* and LDAP_PW_FILTER_*nickname* variables. LISTSERV stops at the first successful search, or when there are no more LDAP servers to query.
2. If none of the searches were successful (no LDAP account exists for this user), LISTSERV:
    a. Rejects the login if LDAP_PW_ONLY=1.
    b. Switches to internal (non-LDAP) login if LDAP_PW_ONLY=0. The login will be validated against the user's internal LISTSERV password, if any, or the user will be prompted to create a LISTSERV password.
3. If an LDAP account was found for this user, LISTSERV:
    a. Rejects the login if LDAP_PW_REQUIRE_SSL=1 and the login request did not come over an SSL session. In this case, LISTSERV does not even try to verify the password.
    b. Verifies the password against the LDAP server where the account was found, and accepts or rejects the login as appropriate.

A note on the "require SSL" option

The purpose of the "require SSL" option is to prevent ordinary, non-malicious users from jeopardizing their login credentials for their personal convenience, for instance by typing clear-text passwords in e-mail requests because it is faster than waiting for a confirmation 'cookie' at the particular Internet café where they are reading their mail. The "require SSL" option effectively disables these login attempts and forces users to log in using the web interface and SSL.

As LISTSERV does not directly process SSL sessions, it has no first-hand knowledge as to whether SSL was used to encrypt the login session or not. It is the web server that handles the SSL session with the user's browser, notifies the LISTSERV web interface that SSL was used, and the web interface script in turn notifies LISTSERV that the password was not sent in clear text. LISTSERV has no way to verify this representation or guarantee that SSL was in fact used to transmit the password. This being said, there is no advantage for a malicious user in logging in to LISTSERV with his own credentials over an unencrypted connection. The malicious user's interest is for *other*, non-malicious users to expose their passwords by sending them in clear text, so that the malicious user may gather them.

Dynamic Query feature

Although *Dynamic Queries* are primarily based on the LDAP interface, they are described in a separate document as they support both LDAP and DBMS data stores.

Known issues and restrictions

The following known issues and restrictions exist:

- **Some unix systems not supported.** At this point, LDAP functionality is not available for Tru64 or HP-UX.
- **OpenLDAP library required to re-link on unix.** Customers wishing to re-link 'lsv' on unix will have to install the **OPENLDAP** library (except on Tru64 and HP-UX), even if they do not want to use LDAP.
- **Static library support not tested on all unix brands.** Our goal is for unix builds with LDAP functionality to work on target systems that do not have the dynamic LDAP library, but we have not tested this on every system.