NAME

ldap_table − Postfix LDAP client configuration

SYNOPSIS

postmap −q "*string***" ldap:/etc/postfix/***filename*

postmap −q − ldap:/etc/postfix/*filename* **<***inputfile*

DESCRIPTION

The Postfix mail system uses optional tables for address rewriting or mail routing. These tables are usually in **dbm** or **db** format.

Alternatively, lookup tables can be specified as LDAP databases.

In order to use LDAP lookups, define an LDAP source as a lookup table in main.cf, for example:

```
alias_maps = ldap:/etc/postfix/ldap−aliases.cf
```

The file /etc/postfix/ldap−aliases.cf has the same format as the Postfix main.cf file, and can specify the parameters described below. An example is given at the end of this manual.

This configuration method is available with Postfix version 2.1 and later. See the section "BACKWARDS COMPATIBILITY" below for older Postfix versions.

For details about LDAP SSL and STARTTLS, see the section on SSL and STARTTLS below.

BACKWARDS COMPATIBILITY

For backwards compatibility with Postfix version 2.0 and earlier, LDAP parameters can also be defined in main.cf. Specify as LDAP source a name that doesn't begin with a slash or a dot. The LDAP parameters will then be accessible as the name you've given the source in its definition, an underscore, and the name of the parameter. For example, if the map is specified as "ldap:*ldapsource*", the "server_host" parameter below would be defined in main.cf as "*ldapsource*_server_host".

Note: with this form, the passwords for the LDAP sources are written in main.cf, which is normally world−readable. Support for this form will be removed in a future Postfix version.

For backwards compatibility with the pre 2.2 LDAP clients, **result_filter** can for now be used instead of **result_format**, when the latter parameter is not also set. The new name better reflects the function of the parameter. This compatibility interface may be removed in a future release.

LIST MEMBERSHIP

When using LDAP to store lists such as $mynetworks, $mydestination, $relay_domains, $local_recipient_maps, etc., it is important to understand that the table must store each list member as a separate key. The table lookup verifies the *existence* of the key. See "Postfix lists versus tables" in the DATABASE_README document for a discussion.

Do NOT create tables that return the full list of domains in $mydestination or $relay_domains etc., or IP addresses in $mynetworks.

DO create tables with each matching item as a key and with an arbitrary value. With LDAP databases it is not uncommon to return the key itself.

For example, NEVER do this in a map defining $mydestination:

```
query_filter = domain=*
result_attribute = domain
```

Do this instead:

```
query_filter = domain=%s
result_attribute = domain
```

GENERAL LDAP PARAMETERS

In the text below, default values are given in parentheses. Note: don't use quotes in these variables; at least, not until the Postfix configuration routines understand how to deal with quoted strings.

server_host (default: localhost)

The name of the host running the LDAP server, e.g.

```
server_host = ldap.example.com
```

Depending on the LDAP client library you're using, it should be possible to specify multiple servers here, with the library trying them in order should the first one fail. It should also be possible to give each server in the list a different port (overriding **server_port** below), by naming them like

```
server_host = ldap.example.com:1444
```

With OpenLDAP, a (list of) LDAP URLs can be used to specify both the hostname(s) and the port(s):

```
server_host = ldap://ldap.example.com:1444
        ldap://ldap2.example.com:1444
```

All LDAP URLs accepted by the OpenLDAP library are supported, including connections over UNIX domain sockets, and LDAP SSL (the last one provided that OpenLDAP was compiled with support for SSL):

```
server_host = ldapi://%2Fsome%2Fpath
        ldaps://ldap.example.com:636
```

server_port (default: 389)

The port the LDAP server listens on, e.g.

```
server_port = 778
```

timeout (default: 10 seconds)

The number of seconds a search can take before timing out, e.g.

```
timeout = 5
```

search_base (No default; you must configure this)

The RFC2253 base DN at which to conduct the search, e.g.

```
search_base = dc=your, dc=com
```

With Postfix 2.2 and later this parameter supports the following '%' expansions:

%% This is replaced by a literal '%' character.

%s This is replaced by the input key. RFC 2253 quoting is used to make sure that the input key does not add unexpected metacharacters.

%u When the input key is an address of the form user@domain, **%u** is replaced by the (RFC 2253) quoted local part of the address. Otherwise, **%u** is replaced by the entire search string. If the localpart is empty, the search is suppressed and returns no results.

%d When the input key is an address of the form user@domain, **%d** is replaced by the (RFC 2253) quoted domain part of the address. Otherwise, the search is suppressed and returns no results.

%[SUD]

For the **search_base** parameter, the upper−case equivalents of the above expansions behave identically to their lower−case counter−parts. With the **result_format** parameter (previously called **result_filter** see the COMPATIBILITY section and below), they expand to the corresponding components of input key rather than the result value.

%[1−9]

The patterns %1, %2, ... %9 are replaced by the corresponding most significant component of the input key's domain. If the input key is *user@mail.example.com*, then %1 is **com**, %2 is **example** and %3 is **mail**. If the input key is unqualified or does not have enough domain components to satisfy all the specified patterns, the search is suppressed and returns no results.

query_filter (default: mailacceptinggeneralid=%s)

The RFC2254 filter used to search the directory, where **%s** is a substitute for the address Postfix is trying to resolve, e.g.

```
query_filter = (&(mail=%s)(paid_up=true))
```

This parameter supports the following '%' expansions:

%% This is replaced by a literal '%' character. (Postfix 2.2 and later).

%s This is replaced by the input key. RFC 2254 quoting is used to make sure that the input key does not add unexpected metacharacters.

%u When the input key is an address of the form user@domain, **%u** is replaced by the (RFC 2254) quoted local part of the address. Otherwise, **%u** is replaced by the entire search string. If the localpart is empty, the search is suppressed and returns no results.

%d When the input key is an address of the form user@domain, **%d** is replaced by the (RFC 2254) quoted domain part of the address. Otherwise, the search is suppressed and returns no results.

%[SUD]

The upper−case equivalents of the above expansions behave in the **query_filter** parameter identically to their lower−case counter−parts. With the **result_format** parameter (previously called **result_filter** see the COMPATIBILITY section and below), they expand to the corresponding components of input key rather than the result value.

The above %S, %U and %D expansions are available with Postfix 2.2 and later.

%[1−9]

The patterns %1, %2, ... %9 are replaced by the corresponding most significant component of the input key's domain. If the input key is *user@mail.example.com*, then %1 is **com**, %2 is **example** and %3 is **mail**. If the input key is unqualified or does not have enough domain components to satisfy all the specified patterns, the search is suppressed and returns no results.

The above %1, ..., %9 expansions are available with Postfix 2.2 and later.

The "domain" parameter described below limits the input keys to addresses in matching domains. When the "domain" parameter is non−empty, LDAP queries for unqualified addresses or addresses in non−matching domains are suppressed and return no results.

NOTE: DO NOT put quotes around the **query_filter** parameter.

result_format (default: %s)

Called **result_filter** in Postfix releases prior to 2.2. Format template applied to result attributes. Most commonly used to append (or prepend) text to the result. This parameter supports the following '%' expansions:

%% This is replaced by a literal '%' character. (Postfix 2.2 and later).

%s This is replaced by the value of the result attribute. When result is empty it is skipped.

%u When the result attribute value is an address of the form user@domain, **%u** is replaced by the local part of the address. When the result has an empty localpart it is skipped.

%d When a result attribute value is an address of the form user@domain, **%d** is replaced by the domain part of the attribute value. When the result is unqualified it is skipped.

%[SUD1−9]

The upper−case and decimal digit expansions interpolate the parts of the input key rather than the result. Their behavior is identical to that described with **query_filter**, and in fact because the input key is known in advance, lookups whose key does not contain all the information specified in the result template are suppressed and return no results.

The above %S, %U, %D and %1, ..., %9 expansions are available with Postfix 2.2 and later.

For example, using "result_format = smtp:[%s]" allows one to use a mailHost attribute as the basis of a transport(5) table. After applying the result format, multiple values are concatenated as comma separated strings. The expansion_limit and size_limit parameters explained below allow one to restrict the number of values in the result, which is especially useful for maps that should return a single value.

The default value **%s** specifies that each attribute value should be used as is.

This parameter was called **result_filter** in Postfix releases prior to 2.2. If no "result_format" is specified, the value of "result_filter" will be used instead before resorting to the default value. This provides compatibility with old configuration files.

NOTE: DO NOT put quotes around the result format!

domain (default: no domain list)

This is a list of domain names, paths to files, or dictionaries. When specified, only fully qualified search keys with a *non−empty* localpart and a matching domain are eligible for lookup: 'user' lookups, bare domain lookups and "@domain" lookups are not performed. This can significantly reduce the query load on the LDAP server.

```
domain = postfix.org, hash:/etc/postfix/searchdomains
```

It is best not to use LDAP to store the domains eligible for LDAP lookups.

NOTE: DO NOT define this parameter for local(8) aliases.

This feature is available in Postfix 1.0 and later.

result_attribute (default: maildrop)

The attribute(s) Postfix will read from any directory entries returned by the lookup, to be resolved to an email address.

```
result_attribute = mailbox, maildrop
```

Don't rely on the default value ("maildrop"). Set the result_attribute explicitly in all ldap table configuration files. This is particularly relevant when no result_attribute is applicable, e.g. cases in which leaf_result_attribute and/or terminal_result_attribute are used instead. The default value is harmless if "maildrop" is also listed as a leaf or terminal result attribute, but it is best to not leave this to chance.

special_result_attribute (default: empty)

The attribute(s) of directory entries that can contain DNs or RFC 2255 LDAP URLs. If found, a recursive search is performed to retrieve the entry referenced by the DN, or the entries matched by the URL query.

```
special_result_attribute = memberdn
```

DN recursion retrieves the same result_attributes as the main query, including the special attributes for further recursion.

URL processing retrieves only those attributes that are included in both the URL definition and as result attributes (ordinary, special, leaf or terminal) in the Postfix table definition. If the URL lists any of the table's special result attributes, these are retrieved and used recursively. A URL that does not specify any attribute selection, is equivalent (RFC 2255) to a URL that selects all attributes, in which case the selected attributes will be the full set of result attributes in the Postfix table.

If an LDAP URL attribute−descriptor or the corresponding Postfix LDAP table result attribute (but not both) uses RFC 2255 sub−type options ("attr;option"), the attribute requested from the LDAP server will include the sub−type option. In all other cases, the URL attribute and the table attribute must match exactly. Attributes with options in both the URL and the Postfix table are requested only when the options are identical. LDAP attribute−descriptor options are very rarely used, most LDAP users will not need to concern themselves with this level of nuanced detail.

terminal_result_attribute (default: empty)

When one or more terminal result attributes are found in an LDAP entry, all other result attributes are ignored and only the terminal result attributes are returned. This is useful for delegating expansion of group members to a particular host, by using an optional "maildrop" attribute on selected groups to route the group to a specific host, where the group is expanded, possibly via mailing−list manager or other special processing.

```
result_attribute =
terminal_result_attribute = maildrop
```

When using terminal and/or leaf result attributes, the result_attribute is best set to an empty value when it is not used, or else explicitly set to the desired value, even if it is the default value "maildrop".

This feature is available with Postfix 2.4 or later.

leaf_result_attribute (default: empty)

When one or more special result attributes are found in a non−terminal (see above) LDAP entry, leaf result attributes are excluded from the expansion of that entry. This is useful when expanding groups and the desired mail address attribute(s) of the member objects obtained via DN or URI recursion are also present in the group object. To only return the attribute values from the leaf objects and not the containing group, add the attribute to the leaf_result_attribute list, and not the result_attribute list, which is always expanded. Note, the default value of "result_attribute" is not empty, you may want to set it explicitly empty when using "leaf_result_attribute" to expand the group to a list of member DN addresses. If groups have both member DN references AND attributes that hold multiple string valued rfc822 addresses, then the string attributes go in "result_attribute". The attributes that represent the email addresses of objects referenced via a DN (or LDAP URI) go in "leaf_result_attribute".

```
result_attribute = memberaddr
special_result_attribute = memberdn
terminal_result_attribute = maildrop
leaf_result_attribute = mail
```

When using terminal and/or leaf result attributes, the result_attribute is best set to an empty value when it is not used, or else explicitly set to the desired value, even if it is the default value "maildrop".

This feature is available with Postfix 2.4 or later.

scope (default: sub)

The LDAP search scope: **sub**, **base**, or **one**. These translate into LDAP_SCOPE_SUBTREE, LDAP_SCOPE_BASE, and LDAP_SCOPE_ONELEVEL.

bind (default: yes)

Whether or how to bind to the LDAP server. Newer LDAP implementations don't require clients to bind, which saves time. Example:

```
# Don't bind
bind = no
# Use SIMPLE bind
bind = yes
# Use SASL bind
bind = sasl
```

Postfix versions prior to 2.8 only support "bind = no" which means don't bind, and "bind = yes" which means do a SIMPLE bind. Postfix 2.8 and later also supports "bind = SASL" when compiled with LDAP SASL support as described in LDAP_README, it also adds the synonyms "bind = none" and "bind = simple" for "bind = no" and "bind = yes" respectively. See the SASL section below for additional parameters available with "bind = sasl".

If you do need to bind, you might consider configuring Postfix to connect to the local machine on a port that's an SSL tunnel to your LDAP server. If your LDAP server doesn't natively support SSL, put a tunnel (wrapper, proxy, whatever you want to call it) on that system too. This should prevent the password from traversing the network in the clear.

bind_dn (default: empty)

If you do have to bind, do it with this distinguished name. Example:

```
bind_dn = uid=postfix, dc=your, dc=com
```

With "bind = sasl" (see above) the DN may be optional for some SASL mechanisms, don't specify a DN if not needed.

bind_pw (default: empty)

The password for the distinguished name above. If you have to use this, you probably want to make the map configuration file readable only by the Postfix user. When using the obsolete ldap:ldapsource syntax, with map parameters in main.cf, it is not possible to securely store the bind password. This is because main.cf needs to be world readable to allow local accounts to submit mail via the sendmail command. Example:

```
bind_pw = postfixpw
```

With "bind = sasl" (see above) the password may be optional for some SASL mechanisms, don't specify a password if not needed.

cache (IGNORED with a warning)

cache_expiry (IGNORED with a warning)

cache_size (IGNORED with a warning)

The above parameters are NO LONGER SUPPORTED by Postfix. Cache support has been dropped from OpenLDAP as of release 2.1.13.

recursion_limit (default: 1000)

A limit on the nesting depth of DN and URL special result attribute evaluation. The limit must be a non−zero positive number.

expansion_limit (default: 0)

A limit on the total number of result elements returned (as a comma separated list) by a lookup against the map. A setting of zero disables the limit. Lookups fail with a temporary error if the limit is exceeded. Setting the limit to 1 ensures that lookups do not return multiple values.

size_limit (default: $expansion_limit)

A limit on the number of LDAP entries returned by any single LDAP search performed as part of the lookup. A setting of 0 disables the limit. Expansion of DN and URL references involves nested LDAP queries, each of which is separately subjected to this limit.

Note: even a single LDAP entry can generate multiple lookup results, via multiple result attributes and/or multi−valued result attributes. This limit caps the per search resource utilization on the LDAP server, not the final multiplicity of the lookup result. It is analogous to the "−z" option of "ldapsearch".

dereference (default: 0)

When to dereference LDAP aliases. (Note that this has nothing do with Postfix aliases.) The permitted values are those legal for the OpenLDAP/UM LDAP implementations:

0	never
1	when searching
2	when locating the base object for the search
3	always

See ldap.h or the ldap_open(3) or ldapsearch(1) man pages for more information. And if you're using an LDAP package that has other possible values, please bring it to the attention of the postfix−users@postfix.org mailing list.

chase_referrals (default: 0)

Sets (or clears) LDAP_OPT_REFERRALS (requires LDAP version 3 support).

version (default: 2)

Specifies the LDAP protocol version to use.

debuglevel (default: 0)

What level to set for debugging in the OpenLDAP libraries.

LDAP SASL PARAMETERS

If you're using the OpenLDAP libraries compiled with SASL support, Postfix 2.8 and later built with LDAP SASL support as described in LDAP_README can authenticate to LDAP servers via SASL.

This enables authentication to the LDAP server via mechanisms other than a simple password. The added flexibility has a cost: it is no longer practical to set an explicit timeout on the duration of an LDAP bind operation. Under adverse conditions, whether a SASL bind times out, or if it does, the duration of the timeout is determined by the LDAP and SASL libraries.

It is best to use tables that use SASL binds via proxymap(8), this way the requesting process can time−out the proxymap request. This also lets you tailer the process environment by overriding the proxymap(8) import_environment setting in master.cf(5). Special environment settings may be needed to configure GSSAPI credential caches or other SASL mechanism specific options. The GSSAPI credentials used for LDAP lookups may need to be different than say those used for the Postfix SMTP client to authenticate to remote servers.

Using SASL mechanisms requires LDAP protocol version 3, the default protocol version is 2 for backwards compatibility. You must set "version = 3" in addition to "bind = sasl".

The following parameters are relevant to using LDAP with SASL

sasl_mechs (default: empty)

Space separated list of SASL mechanism(s) to try.

sasl_realm (default: empty)

SASL Realm to use, if applicable.

sasl_authz_id (default: empty)

The SASL authorization identity to assert, if applicable.

sasl_minssf (default: 0)

The minimum required sasl security factor required to establish a connection.

LDAP SSL AND STARTTLS PARAMETERS

If you're using the OpenLDAP libraries compiled with SSL support, Postfix can connect to LDAP SSL servers and can issue the STARTTLS command.

LDAP SSL service can be requested by using a LDAP SSL URL in the server_host parameter:

```
server_host = ldaps://ldap.example.com:636
```

STARTTLS can be turned on with the start_tls parameter:

```
start_tls = yes
```

Both forms require LDAP protocol version 3, which has to be set explicitly with:

```
version = 3
```

If any of the Postfix programs querying the map is configured in master.cf to run chrooted, all the certificates and keys involved have to be copied to the chroot jail. Of course, the private keys should only be readable by the user "postfix".

The following parameters are relevant to LDAP SSL and STARTTLS:

start_tls (default: no)

Whether or not to issue STARTTLS upon connection to the server. Don't set this with LDAP SSL (the SSL session is setup automatically when the TCP connection is opened).

tls_ca_cert_dir (No default; set either this or tls_ca_cert_file)

Directory containing X509 Certification Authority certificates in PEM format which are to be recognized by the client in SSL/TLS connections. The files each contain one CA certificate. The files are looked up by the CA subject name hash value, which must hence be available. If more than one CA certificate with the same name hash value exist, the extension must be different (e.g. 9d66eef0.0, 9d66eef0.1 etc). The search is performed in the ordering of the extension number, regardless of other properties of the certificates. Use the c_rehash utility (from the OpenSSL distribution) to create the necessary links.

tls_ca_cert_file (No default; set either this or tls_ca_cert_dir)

File containing the X509 Certification Authority certificates in PEM format which are to be recognized by the client in SSL/TLS connections. This setting takes precedence over tls_ca_cert_dir.

tls_cert (No default; you must set this)

File containing client's X509 certificate to be used by the client in SSL/ TLS connections.

tls_key (No default; you must set this)

File containing the private key corresponding to the above tls_cert.

tls_require_cert (default: no)

Whether or not to request server's X509 certificate and check its validity when establishing SSL/TLS connections. The supported values are **no** and **yes**.

With **no**, the server certificate trust chain is not checked, but with OpenLDAP prior to 2.1.13, the name in the server certificate must still match the LDAP server name. With OpenLDAP 2.0.0 to 2.0.11 the server name is not necessarily what you specified, rather it is determined (by reverse lookup) from the IP address of the LDAP server connection. With OpenLDAP prior to 2.0.13, subjectAlternativeName extensions in the LDAP server certificate are ignored: the server name must match the subject CommonName. The **no** setting corresponds to the **never** value of **TLS_REQCERT** in LDAP client configuration files.

Don't use TLS with OpenLDAP 2.0.x (and especially with x <= 11) if you can avoid it.

With **yes**, the server certificate must be issued by a trusted CA, and not be expired. The LDAP server name must match one of the name(s) found in the certificate (see above for OpenLDAP library version dependent behavior). The **yes** setting corresponds to the **demand** value of **TLS_REQCERT** in LDAP client configuration files.

The "try" and "allow" values of **TLS_REQCERT** have no equivalents here. They are not available with OpenLDAP 2.0, and in any case have questionable security properties. Either you want TLS verified LDAP connections, or you don't.

The **yes** value only works correctly with Postfix 2.5 and later, or with OpenLDAP 2.0. Earlier Postfix releases or later OpenLDAP releases don't work together with this setting. Support for LDAP over TLS was added to Postfix based on the OpenLDAP 2.0 API.

tls_random_file (No default)

Path of a file to obtain random bits from when /dev/[u]random is not available, to be used by the client in SSL/TLS connections.

tls_cipher_suite (No default)

Cipher suite to use in SSL/TLS negotiations.

EXAMPLE

Here's a basic example for using LDAP to look up local(8) aliases. Assume that in main.cf, you have:

```
alias_maps = hash:/etc/aliases,
        ldap:/etc/postfix/ldap−aliases.cf
```

and in ldap:/etc/postfix/ldap−aliases.cf you have:

```
server_host = ldap.example.com
search_base = dc=example, dc=com
```

Upon receiving mail for a local address "ldapuser" that isn't found in the /etc/aliases database, Postfix will search the LDAP server listening at port 389 on ldap.example.com. It will bind anonymously, search for any directory entries whose mailacceptinggeneralid attribute is "ldapuser", read the "maildrop" attributes of those found, and build a list of their maildrops, which will be treated as RFC822 addresses to which the message will be delivered.

SEE ALSO

postmap(1), Postfix lookup table manager
postconf(5), configuration parameters
mysql_table(5), MySQL lookup tables
pgsql_table(5), PostgreSQL lookup tables

README FILES

Use "**postconf readme_directory**" or "**postconf html_directory**" to locate this information.
DATABASE_README, Postfix lookup table overview
LDAP_README, Postfix LDAP client guide

LICENSE

The Secure Mailer license must be distributed with this software.

AUTHOR(S)

Carsten Hoeger, Hery Rakotoarisoa, John Hensley, Keith Stevenson, LaMont Jones, Liviu Daia, Manuel Guesdon, Mike Mattice, Prabhat K Singh, Sami Haahtinen, Samuel Tardieu, Victor Duchovni, and many others.