MyDNS reference manual

for version 1.1.0, 4 September 2006

This is the manual for MyDNS (version 1.1.0, 4 September 2006)

Table of Contents

1 Introduction to MyDNS

MyDNS is an Internet domain name server. It is unique among DNS servers in that it was designed explicitly to work with an external SQL database.

At the moment, two popular open-source databases are supported: MySQL and PostgreSQL.

The primary goals of the MyDNS package are stability, security, interoperability, simplicity, and speed. But not necessarily in that order.

This manual assumes that the reader has a working understanding of basic DNS concepts and their SQL server.

2 Installation

2.1 Compiling the source

First, uncompress and unpack the distribution to a location of your choosing. The distribution will be extracted into a directory named '`mydns-1.1.0`'.

Change directory into the distribution directory, then run the `configure` script to configure the package for your system.

The '`INSTALL`' file has full details on how the `configure` script works. Run `configure --help` to output a summary of all options.

You'll probably be able to simply run the `configure` command with no additional arguments.

```
$ ./configure
```

If you have both MySQL and PostgreSQL installed on your system, MyDNS chooses MySQL by default. To tell MyDNS to use PostgreSQL instead of MySQL, run `configure --without-mysql`.

After the configuration process is complete, build the package.

```
$ make
```

Then, if the build completed successfully, install the package.

```
# make install
```

The '`mydns`' binary should now be installed in the '`sbin`' dir beneath the '`prefix`' provided to the `configure` script, if any.

By default, '`mydns`' is installed as '`/usr/local/sbin/mydns`'.

2.2 Creating the database

Now that you have installed MyDNS, you'll need to set up a database and access permissions.

To create a database called '`mydns`' on your database server:

MySQL:

```
$ mysqladmin -h host -u username -p create mydns
```

PostgreSQL:

```
$ createdb mydns
```

2.3 Creating the tables

Next, create the tables in your database that will hold the DNS data.

Running `mydns --create-tables` will cause MyDNS to output `CREATE TABLE` statements appropriate for your database.

MySQL:

```
$ mydns --create-tables | mysql -h host -u username -p mydns
```

PostgreSQL:

```
$ mydns --create-tables | psql mydns
```

After you have created the tables, you should have two tables in your '`mydns`' database, called '`soa`' (see Section 3.1 [soa table], page 4) and '`rr`' (see Section 3.2 [rr table], page 5).

2.4 Database access

Next, create a user that the MyDNS server can use to access the '`mydns`' database:

MySQL:

```
$ mysql -h host -u username -p mydns
mysql> GRANT SELECT ON mydns.* TO user@localhost IDENTIFIED BY 'password';
```

PostgreSQL:

```
$ psql mydns
mydns=# CREATE USER user WITH PASSWORD 'password';
mydns=# GRANT SELECT ON soa,soa_id_seq TO user;
mydns=# GRANT SELECT ON rr,rr_id_seq TO user;
```

2.5 Creating '`mydns.conf`'

MyDNS probably won't start properly if it can't find its configuration file. By default, the configuration file is called '`/etc/mydns.conf`'.

Running `mydns --dump-config` will output a set of configuration options suitable for using as your configuration file. See Appendix B [Configuration], page 19.

So, for a fresh installation, this command will create your initial configuration file:

```
# mydns --dump-config > /etc/mydns.conf
```

You should now edit your '`mydns.conf`' file. Most defaults should be fine for most sites.

The values you are most likely to want to modify are the values for '`db-host`' and '`db-user`', which should contain the database username and password that you created at the end of step 2.4. See Section 2.4 [Database access], page 3.

3 Database

The default database name is '`mydns`'.

To specify a different name, edit the '`database`' variable in your '`mydns.conf`'.

You can freely add columns to the '`mydns`' database. You can also modify the columns that MyDNS uses, as long as you don't change their names.

The table layouts described here are for the tables created on a MySQL database. If you're using PostgreSQL, the fields are pretty much the same; however, the field types are slightly different. You can run `mydns --create-tables` to see the exact table structures.

3.1 The 'soa' table

The '`soa`' table contains one row for each zone for which the server is authoritative.

The default values for the various timer fields are from RFC 1537.

'`id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY`' (*MySQL*)
'`id SERIAL NOT NULL PRIMARY KEY`' (*PostgreSQL*)
: A unique number identifying this zone.

'`origin CHAR(255) NOT NULL`' (*MySQL*)
'`origin VARCHAR(255) NOT NULL`' (*PostgreSQL*)
: The name of this zone. (*Unique key*)
ex: `example.com.`

'`ns CHAR(255) NOT NULL`' (*MySQL*)
'`ns VARCHAR(255) NOT NULL`' (*PostgreSQL*)
: The name of the name server that was the original or primary source of data for this zone. *(meaningless to MyDNS)*
ex: `primary.example.com.`

'`mbox CHAR(255) NOT NULL`' (*MySQL*)
'`mbox VARCHAR(255) NOT NULL`' (*PostgreSQL*)
: A name which specifies the mailbox of the person responsible for this zone. This should be specified in the mailbox-as-domain-name format where the '`@`' character is replaced with a dot. *(meaningless to MyDNS)*
ex: `postmaster.example.com.`

'`serial INT UNSIGNED NOT NULL DEFAULT '1'`' (*MySQL*)
'`serial INTEGER NOT NULL DEFAULT 1`' (*PostgreSQL*)
: A "version number" for this zone. DNS servers that rely on AXFR for zone transfers use this to determine when updates have occurred. Popular values to use are the Unix timestamp or a date in the form YYYYMMDD. (see Section 4.4 [Zone transfers], page 12).
ex: `20020529`

'`refresh INT UNSIGNED NOT NULL DEFAULT '28800'`' (*MySQL*)
'`refresh INTEGER NOT NULL DEFAULT 28800`' (*PostgreSQL*)
: The number of seconds after which slave nameservers should check to see if this zone has been changed. If the zone's serial number has changed, the slave

nameserver initiates a zone transfer. *(meaningless to MyDNS)*
ex: `10800`

'`retry INT UNSIGNED NOT NULL DEFAULT '7200''` (*MySQL*)
'`retry INTEGER NOT NULL DEFAULT 7200`' (*PostgreSQL*)
This specifies the number of seconds a slave nameserver should wait before retrying if it attmepts to transfer this zone but fails. *(meaningless to MyDNS)*
ex: `3600`

'`expire INT UNSIGNED NOT NULL DEFAULT '604800''` (*MySQL*)
'`expire INTEGER NOT NULL DEFAULT 604800`' (*PostgreSQL*)
If for *expire* seconds the primary server cannot be reached, all information about the zone is invalidated on the secondary servers (i.e., they are no longer authoritative for that zone). *(meaningless to MyDNS)*
ex: `60400`

'`minimum INT UNSIGNED NOT NULL DEFAULT '86400''` (*MySQL*)
'`minimum INTEGER NOT NULL DEFAULT 86400`' (*PostgreSQL*)
The minimum TTL field that should be exported with any RR from this zone. If any RR in the database has a lower TTL, this TTL is sent instead.
ex: `86400`

'`ttl INT UNSIGNED NOT NULL DEFAULT '86400''` (*MySQL*)
'`ttl INTEGER NOT NULL DEFAULT 86400`' (*PostgreSQL*)
The number of seconds that this zone may be cached before the source of the information should again be consulted. Zero values are interpreted to mean that the zone should not be cached.
ex: `86400`

3.2 The 'rr' table

The '`rr`' table contains all non-`SOA` resource record types.

It has a unique key on the combination of *zone*, *name*, *type*, and *data*.

'`id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY`' (*MySQL*)
'`id SERIAL NOT NULL PRIMARY KEY`' (*PostgreSQL*)
A unique number identifying this record.

'`zone INT UNSIGNED NOT NULL`' (*MySQL*)
'`zone INTEGER NOT NULL`' (*PostgreSQL*)
The ID of the zone (from the '`soa`' table) to which this resource record belongs. (see Section 3.1 [soa table], page 4).

For PostgreSQL databases, this column is also created with '`FOREIGN KEY (zone) REFERENCES soa (id) ON DELETE CASCADE`'.

'`name CHAR(64) NOT NULL`' (*MySQL*)
'`name VARCHAR(64) NOT NULL`' (*PostgreSQL*)
The name that this RR describes. Wildcard values such as '`*`' or '`*.sub`' are supported, and this field can contain a FQDN or just a hostname. It may contain out-of-zone data if this is a glue record.

ex: `foo`
ex: `foo.example.com.`

‘`type ENUM('A','AAAA','CNAME','HINFO','MX','NAPTR','NS','PTR','RP','SRV','TXT') NOT NULL`’ (*MySQL*)
‘`type VARCHAR(5) NOT NULL CHECK (type='A' OR type='AAAA' OR type='CNAME' OR type='HINFO' OR type='MX' OR type='NAPTR' OR type='NS' OR type='PTR' OR type='RP' OR type='SRV' OR type='TXT')`’ (*PostgreSQL*)
The type of resource record. (see Section 3.3 [Supported RR types], page 6).

‘`data CHAR(128) NOT NULL`’ (*MySQL*)
‘`data VARCHAR(128) NOT NULL`’ (*PostgreSQL*)
The data associated with this resource record. See Section 3.3 [Supported RR types], page 6, for specifications and examples of the type of data each record type should contain.

‘`aux INT UNSIGNED NOT NULL`’ (*MySQL*)
‘`aux INTEGER NOT NULL default 0`’ (*PostgreSQL*)
An auxillary numeric value in addition to *data*. For ‘`MX`’ records, this field specifies the preference. For ‘`SRV`’ records, this field specifies the priority.

‘`ttl INT UNSIGNED NOT NULL DEFAULT '86400'`’ (*MySQL*)
‘`ttl INTEGER NOT NULL default 86400`’ (*PostgreSQL*)
The time interval that this resource record may be cached before the source of the information should again be consulted. Zero values are interpreted to mean that the RR can only be used for the transaction in progress, and should not be cached.

3.3 Supported RR types

The ‘`type`’ column in the ‘`rr`’ table may contain any of the following supported resource record types:

‘`A`’ A host address. The ‘`data`’ column should contain the IP address (in numbers-and-dots format) associated with the ‘`name`’.

example: ‘`192.168.1.88`’

‘`AAAA`’ An IPv6 host address. The ‘`data`’ column should contain the IPv6 address associated with the ‘`name`’.

example: ‘`3ffe:b00:c18:3::a`’

‘`ALIAS`’ A server side alias. An alias is like a `CNAME`, only it is handled entirely by the server. The ‘`data`’ column should contain the hostname aliased by ‘`name`’. Aliases can be used in place of `A` records. The client will only see `A` records and will not be able to tell that aliases are involved. The hostname specified by ‘`data`’ must exist in the database.

It can be useful to use aliases for everything. Use `A` records for the canonical name of the machine and use aliases for any additional names. This is especially useful when combined with automatic `PTR` records. If a single IP address is only

used for one `A` record, then there will never be any confusion over what the `PTR` record should be.

In order for server-side aliases to work, MyDNS must have been compiled with `configure --enable-alias`.

example: '`albuquerque.example.com.`' (FQDN)
example: '`albuquerque`' (hostname only)

'`CNAME`' The canonical name for an alias. The '`data`' column should contain the real name of the machine specified by '`name`'. '`data`' may be a hostname or an FQDN.

example: '`porcini.example.com.`' (FQDN)
example: '`porcini`' (hostname only)

'`HINFO`' Host information. The '`data`' column should contain two strings which provide information about the host specified by '`name`'. The first string specifies the CPU type, and the second string describes the operating system type. The two strings should be separated by a space. If either string needs to contain a space, enclose it in quotation marks.

Well-known CPU and operating system types that are most often used are listed in RFC 1700.

example: '`"Pentium Pro" Linux`'

'`MX`' Mail exchange. The '`data`' column should contain the hostname or FQDN of a mail server which will accept mail for the host specified by '`name`'. The '`aux`' column should contain a preference for this mail server. Mail transfer agents prefer MX records with lower values in '`aux`'.

example: '`ns0.example.com.`' (FQDN)
example: '`ns0`' (hostname only)

'`NAPTR`' Naming authority pointer. The '`data`' column should contain six fields, separated by whitespace, which describe a regular expression based rewrite rule as described in RFC 2915 for the name specified by '`name`'. The first field is the order (a number) in which the record must be processed with other similar records. The second field is the preference (a number) in which similar records with equal order values should be processed. The third field (a string) describe processing flags used while rewriting. The fourth field (a string) specifies the services available down this rewrite path. The fifth field (a string) contains a regular expression to use while rewriting. The last field (a string) contains the next name to query along the rewrite path. If a string contains spaces, it may be enclosed in quotation marks. If a string needs to contain a literal quotation mark, precede it with a backslash character.

example: '`100 90 "" "" "!http://([^/:]+)!\1!i" .`'

'`NS`' An authoritative nameserver. The '`data`' column should contain the hostname or FQDN of a server which should be considered authoritative for the zone listed in '`name`'.

example: '`france.example.com.`' (FQDN)
example: '`france`' (hostname only)

'PTR' A domain name pointer. These records, used only with *IN-ADDR.ARPA* zones, should contain the canonical hostname of the machine referred to by '`name`' in '`data`'.

example: '`webserver.example.com.`'

'RP' A responsible person. The '`data`' column should contain the DNS-encoded email address of the person responsible for the name requested, then a space, then a hostname that should return a TXT record containing additional information about the responsible person. If there is no such TXT record, the second value should contain a dot ('`.`').

For more information, see RFC 1183.

example: '`webmaster.example.com. contactinfo.example.com.`'

'SRV' Server location. Specifies the location of the server(s) for a specific protocol and domain. The '`data`' column must contain three space-separated values. The first value is a number specifying the *weight* for this entry. The second field is a number specifying the *port* on the target host of this service. The last field is a name specifying the *target* host. The '`aux`' column should contain the *priority* of this target host. Targets with a lower priority are preferred.

For well-known services, a reserved universal symbolic name may be listed in RFC 1700.

For more information, see RFC 2782.

example: '`0 9 server.example.com.`' (FQDN)
example: '`0 9 server`' (hostname only)

'TXT' A text string. The '`data`' column contains a text string that is returned only when a *TXT* query is issued for the host specified by '`name`'.

example: '`This is a string.`'

3.4 Optional columns

Each of these columns is optional.

If these columns exist, MyDNS will notice this and enable additional code specific to each optional field.

If you add any of these fields to your database, you must signal MyDNS to rescan the tables by sending it a SIGHUP signal (see Section 4.2 [Signals], page 12).

3.4.1 any.active

Both the '`soa`' table and the '`rr`' table may contain a column called '`active`'.

If this column exists, it should contain a boolean value. This could be `0/1` (an integer), `'Y'/'N'`, `'1'/'0'`, or `'Active'/'Inactive'`. For MySQL databases, an `ENUM` value is recommended.

If the `active` column is present, whenever records are retrieved from that table, the `active` column will be honored. If the row is inactive, it will be as if the row did not exist at all.

To create an '`active`' column on your '`soa`' table, for example, you might issue SQL statements like this:

MySQL:

```
mysql> ALTER TABLE mydns.soa ADD COLUMN active ENUM('Y','N') NOT NULL;
mysql> ALTER TABLE mydns.soa ADD INDEX (active);
```

PostgreSQL:

```
mydns=# ALTER TABLE soa ADD COLUMN active INT;
mydns=# UPDATE soa SET active=1;
mydns=# ALTER TABLE soa ALTER COLUMN active SET NOT NULL;
mydns=# ALTER TABLE soa ALTER COLUMN active SET DEFAULT 1;
```

3.4.2 soa.xfer

If the '`soa`' table contains a column named '`xfer`' and DNS-based zone transfers are enabled (see Section 4.4 [Zone transfers], page 12), the '`xfer`' column will be examined whenever a DNS-based zone transfer request is received.

The '`xfer`' column should contain one or more IP addresses separated by commas. These IP addresses will be allowed to transfer the zone.

If the '`xfer`' column is empty, no DNS-based zone transfers will be allowed for that zone.

The IP addresses in '`xfer`' may contain standard wildcard characters. Thus, if you want to grant zone transfer access for a particular zone to any IP address, you would set '`xfer`' to '`*`'.

Addresses may also be specified in CIDR notation (i.e. `192.168.1.1/24`) or in network/netmask notation (i.e. `192.168.1.1/255.255.0.0`).

The '`xfer`' column may be any size you want, and whatever size you think will be adequate for the IP address lists you intend to use.

To create an '`xfer`' column on your '`soa`' table, for example, you might issue SQL statements like this:

MySQL:

```
mysql> ALTER TABLE mydns.soa ADD COLUMN xfer CHAR(255) NOT NULL;
```

PostgreSQL:

```
mydns=# ALTER TABLE soa ADD COLUMN xfer VARCHAR(255);
mydns=# UPDATE soa SET xfer='';
mydns=# ALTER TABLE soa ALTER COLUMN xfer SET NOT NULL;
mydns=# ALTER TABLE soa ALTER COLUMN xfer SET DEFAULT '';
```

3.4.3 soa.update_acl

If the '`soa`' table contains a column named '`update_acl`' and dynamic DNS updates are enabled (see Section 4.5 [DNS UPDATE], page 12), the '`update`' column will be examined whenever a DNS UPDATE request is received.

The '`update_acl`' column should contain one or more IP addresses separated by commas. These IP addresses will be allowed to update the zone.

If the '`update_acl`' column is empty, no dynamic DNS updates will be allowed for that zone.

The IP addresses in '`update_acl`' may contain standard wildcard characters. Thus, if you want to grant access for a particular zone to any IP address, you would set '`update_acl`' to '`*`'.

Addresses may also be specified in CIDR notation (i.e. `192.168.1.1/24`) or in network/netmask notation (i.e. `192.168.1.1/255.255.0.0`).

The '`update_acl`' column may be any size you want, and whatever size you think will be adequate for the IP address lists you intend to use.

To create an '`update_acl`' column on your '`soa`' table, for example, you might issue SQL statements like this:

MySQL:

```
mysql> ALTER TABLE mydns.soa ADD COLUMN update_acl CHAR(255) NOT NULL;
```

PostgreSQL:

```
mydns=# ALTER TABLE soa ADD COLUMN update_acl VARCHAR(255);
mydns=# UPDATE soa SET update_acl='';
mydns=# ALTER TABLE soa ALTER COLUMN update_acl SET NOT NULL;
mydns=# ALTER TABLE soa ALTER COLUMN update_acl SET DEFAULT '';
```

4 Server

4.1 Caching

MyDNS uses a lightweight internal cache to speed up question resolution. When the DNS server receives a question, it descends through each label in the name, looking for the first label that has any associated resource records (see RFC 1034).

This means that a request for a name with lots of labels may require many database queries, most of which are likely to return no rows.

MyDNS stores positive results in its zone cache. The size of the zone cache is determined by the `zone-cache-size` variable in '`mydns.conf`'. The zone-cache-size specifies the *number of entries* the zone cache may contain at any given time. If the zone-cache-size is set to zero, the zone cache will be completely disabled, and the database will be queried every time. Typically, the bigger your cache, the better MyDNS will perform. Large sites may consider a cache around 32768 entries. The default size is 8192 entries.

The `zone-cache-expire` variable in '`mydns.conf`' specifies the number of seconds after which zone cache data expires. Most installations will want to set this value fairly low, maybe 60 seconds or so. This way, the DNS data being served by MyDNS will never be more than 60 seconds behind what is actually stored in the database. If your database changes infrequently, set this value much higher.

If any RR stored in the zone data cache has a TTL that is shorter than the value of `zone-cache-expire`, the cached data will expire when the TTL expires.

Once a complete reply has been constructed for a specific request (for example, `IN A foo.example.com.`), the completed reply will be stored in the reply cache. The size of the reply cache is determined by the `reply-cache-size` variable in '`mydns.conf`'. Entries in the reply cache expire after `reply-cache-expire` seconds.

The reply cache is especially useful because if a match is found for a request in the reply cache, MyDNS will not need to perform any database queries or even very much internal computation in order to return the reply.

A good way to check your cache configuration is to send `SIGUSR2` to your server:

```
# kill -USR2 `cat /var/run/mydns.pid`
```

The server will then output its cache status. For example

```
mydns: zone cache 47% useful (31385 hits, 15894 misses),
       2143 collisions (5%), 100% full (8192 records),
       12711624 bytes, avg life 27 sec
mydns: reply cache 84% useful (55200 hits, 10718 misses),
       5707 collisions (14%), 100% full (8192 records),
       3357269 bytes, avg life 38 sec
```

This tells you that MyDNS has been able to find the answer to a question in the reply cache (avoiding all database queries) 84 percent of the time, and that the other 16 percent of the time, it was able to find the data needed in the zone cache 47 percent of the time.

When tweaking your cache sizes, the best clue in this output is the "avg life". This is the average amount of time an entry remains in the cache, between the time it was first inserted

and the time it was removed due to either expiration or because it was removed to make room for other, more commonly-requested entries.

If your "avg life" is extremely short (just a second or two) you should consider increasing your cache size. Of course, if the average life is very short because your zone data has extremely short TTL values, this is to be expected.

A very long `zone-cache-expire/reply-cache-expire` time means that the results returned by MyDNS are more likely to be out-of-date, especially if your database is constantly being updated. Most DNS data is not.

4.2 Signals

If you send '`SIGHUP`' to MyDNS, it empties its cache.

MyDNS responds to '`SIGUSR1`' by outputting some brief server statistics.

MyDNS responds to '`SIGUSR2`' by outputting cache statistics.

4.3 TCP support

MyDNS will process all TCP requests it receives if the configuration option '`allow-tcp`' is true. This is not usually necessary or recommended. TCP support will make the server run a little slower, and a denial-of-service attack is easier if TCP is allowed.

Some very large sites may require TCP support, however. If a response set would exceed the UDP message size limit (512 bytes), MyDNS will set the TC (truncated) flag on its answer. Some clients will then fall back to TCP, which can handle such large answers. If TCP support is enabled, those clients can get their responses. Also, TCP support is required to perform DNS-based zone transfers.

4.4 Zone transfers

MyDNS will allow zone transfers (via AXFR) if the configuration option '`allow-axfr`' is true. This is recommended only if you have an absolute need for DNS-based zone transfers, such as if your secondary name server is running BIND.

MyDNS does *not* support incremental zone transfers (`IXFR`).

If you need to support DNS-based zone transfers, you have to enable '`allow-tcp`'. (This is not true for BIND 9.)

You can specify IP access rules for DNS-based zone transfers by using an optional column called '`xfer`' in the `soa` table. See Section 3.4.2 [soa.xfer], page 9.

4.5 DNS UPDATE

MyDNS will allow dynamic DNS updates (described in RFC 2136) if the configuration option '`allow-update`' is true.

You can specify IP access rules for DNS UPDATE by using an optional column called '`update_acl`' in the `soa` table. See Section 3.4.3 [soa.update_acl], page 9.

If the '`update_acl`' column does not exist in the `soa` table, DNS UPDATE requests will be allowed only from localhost.

In order for dynamic DNS updates to work, the '`db-user`' specified in the MyDNS configuration file must have permissions to insert and update on the `rr` table.

If MyDNS receives multiple UPDATE requests in one packet, they must all complete successfully, or the UPDATE must fail. Therefore, your database must have transactional capabilities if you enable DNS UPDATE.

For more information, see RFC 2136.

4.6 Round robin

If your `rr` table contains more than one address record for the same name (but with different addresses, of course), MyDNS will serve them up in a random order each time.

Round robin is used only if all the address records found have an `aux` value of '`0`'. If any of the records have an `aux` value that is non-zero, load balancing will be used instead. (See Section 4.7 [Load balancing], page 13.)

Note that MyDNS will also return multiple same-preference MX records in random order, to help equalize the load among same-preference MX hosts.

4.7 Load balancing

If your `rr` table contains more than one address record for the same name, and one or more of the records has an `aux` value greater than zero, MyDNS will weight its response using the value in `aux`.

MyDNS uses the value in `aux` to determine the order in which addresses are listed. Clients usually start with the first address and work their way down, so addresses that are usually listed first will bear the heaviest client load.

A low value in `aux` makes an address record more likely to be listed first. The balancing algorithm causes servers with a lower `aux` to be selected more frequently than those with higher values, although all servers will still be listed first occasionally, as the algorithm is partially random.

Records where `aux` is 0 (zero) will be listed first almost every time. Records where `aux` is 50,000 or greater will always be listed last.

Here's an example of how hosts were distributed on a 100,000 query test against ten hosts with aux values 10-100. The number shown is the number of times that host was listed first:

```
aux 10   51,211
aux 20   21,881
aux 30   10.983
aux 40    6,209
aux 50    3,661
aux 60    2,311
aux 70    1,526
```

```
aux 80    1,032
aux 90      675
aux 100     511
```

4.8 Logging queries

If MyDNS is started with the `--verbose` (`-v`) option, each query the server receives will be output via the logging mechanism specified in your configuration file (see Section B.4 [Misc options], page 20).

Each log line consists of the program name (and possibly the PID) followed by a colon, then seventeen fields separated by spaces. For example:

```
mydns: 25-Jul-2003 01:50:11+659583 #1 3987 UDP 192.168.1.1 IN ANY
bboy.net. NOERROR - 1 11 0 5 LOG Y QUERY
```

or

```
mydns: 25-Jul-2003 01:50:44+720684 #2 33848 UDP 192.168.1.1 IN ANY
bogus.example.com. NXDOMAIN No_matching_resource_records 1 0 1 0 LOG N QUERY
```

In order, here's what these fields mean:

1. The date the query was received.
2. The time the query was received, then a plus sign ('`+`'), then the number of microseconds after the second the query was received.
3. A pound sign ('`#`') followed by the server's internal ID number for this query. The internal ID numbers begin at 0 and advance sequentially.
4. The query ID provided by the client. This is usually a seemingly-random 16-bit number used by the client to make sure the answer it receives matches the question it asked.
5. The transport used, always either `TCP` or `UDP`.
6. The client IP address, in dotted-decimal notation.
7. The query class, always `IN`.
8. The query type, such as `A`, `MX`, `NS`, etc. (see Section 3.3 [Supported RR types], page 6).
9. The name being requested.
10. The result of the query. The following values are possible:

    NOERROR
    : No error; the query was successful.

    FORMERR
    : The server was unable to interpret the query.

    SERVFAIL
    : MyDNS experienced an internal error, usually the result of some malformed data in the database.

    NXDOMAIN
    : No resource records (of any type) exist matching the domain name requested.

    NOTIMP
    : The requested type of query is not implemented.

REFUSED
: The query was refused due to server policy. This usually happens because the client attempted to `AXFR` a zone that they were not allowed to transfer, or because the client requested a name within a zone for which the server is not authoritative.

11. If the previous field was anything but `NOERROR`, this is a human-readable reason why the query failed, with any space characters in the string converted into underscore ('`_`') characters. If the previous field was `NOERROR`, this field contains a dash ('`-`').
12. The number of resource records included in the *question* section of the reply.
13. The number of resource records included in the *answer* section of the reply.
14. The number of resource records included in the *authority* section of the reply.
15. The number of resource records included in the *additional* section of the reply.
16. The word `LOG`.
17. The character '`Y`' if this was a cached reply, '`N`' if it was not.
18. The opcode for this query – '`QUERY`' or '`UPDATE`'.
19. If the previous field was '`UPDATE`', this is a description of the update performed, enclosed in quotation marks. For example, this field might contain '`"test-a.example.com. 3600 IN A 0 1.2.3.4"`', indicating that for the zone specified, an `A` record was created for `test-a.example.com.` with the value `1.2.3.4`.

There is a script in the `contrib/` directory of the source distribution called `stats.php` that provides an example of how a script might read and parse these lines, in case you wanted to accumulate usage information or something.

5 Utilities

MyDNS includes several helpful utilities.

All utilities support the '`--host`', '`--database`', '`--user`', and '`--password`' options.

5.1 mydnscheck

The '`mydnscheck`' program scans one more more zones and reports on syntax and consistency problems in the zone data. When used without any zone arguments, '`mydnscheck`' checks all zones by default.

'`mydnscheck`' outputs lines of tab-delimited data. This is so that it will hopefully be easier for experienced users to write scripts to automate fixups, in the event that they have created a new database that has many problems. Each line contains seven fields:

1. A brief, human-readable string describing the error found.
2. The zone ID, or '`-`' if no zone ID is applicable.
3. The resource record ID, or '`-`' if no resource record ID is applicable.
4. The name, or '`-`' if no name is applicable.
5. The ttl (time-to-live) value, or '`-`' if no ttl is applicable.
6. The resource record type, or '`-`' if no type is applicable.
7. The data value, or '`-`' if no data value is applicable.

The most useful way for an administrator to use '`mydnscheck`' is without any arguments (indicating a scan of all zones) and with the database consistency check option enabled. This will perform a thorough analysis of your database. To perform this type of check, you would run:

```
# mydnscheck --consistency
```

You can also run '`mydnscheck`' on a single zone only. This might be useful if invoked from a CGI script, to offer customers or clients the ability to check their zone:

```
$ mydnscheck -uUSER -pPASS example.com
```

For an explanation of all available options, please see the '`mydnscheck`'(8) man page.

5.2 mydnsexport

The '`mydnsexport`' program outputs zone data in various formats understood by DNS servers other than MyDNS.

By default, '`mydnsexport`' exports a single zone specified on the command line in BIND format. The following command would output the `example.com` zone in BIND zone file format:

```
# mydnsexport example.com
```

'`mydnsexport`' can also output *tinydns-data* style data files, as used by the `tinydns` name server, by specifying the '`-t, --tinydns-data`' option. If this output format is specified, and no zone names are provided on the command line, '`mydnsexport`' will output all zones.

For an explanation of all available options, please see the '`mydnsexport`'(8) man page.

5.3 mydnsimport

The '`mydnsimport`' program can be used to import data into your MyDNS database from external sources. This is the simplest way to seed your database when migrating from another name server to MyDNS.

The only import option supported at this time is the '`-a, --axfr`' option. Pretty much every name server on the market supports DNS-based zone transfers via `AXFR`. Make sure your MyDNS server has permission to request a zone transfer for the zone you wish to import, then specify the host name and zone name with the '`--axfr`' option.

If you want to test permissions, you can use the '`dig`' command, like:

```
# dig @bind.example.com axfr example.com
```

Let's say you have a BIND server located at `bind.example.com` and you are going to get rid of it and switch to MyDNS. Great! You want to import the zones `example.com` and `example.net`, as well as the PTR records from `1.168.192.in-addr.arpa`. You would issue the following commands:

```
# mydnsimport --axfr=bind.example.com example.com example.net
# mydnsimport --axfr=bind.example.com 1.168.192.in-addr.arpa
```

For an explanation of all available options, please see the '`mydnsimport(8)`' man page.

Appendix A Troubleshooting

Of the problems you may encounter while running MyDNS, the vast majority will be caused by *inappropriate* data in your tables. MyDNS does not know what your intentions are, and will serve the data as you have specified it. The best way to make sure your data seems reasonable is to run the provided data validation utility. (See Section 5.1 [mydnscheck], page 16.)

If you give the '`--enable-debug`' option to the '`configure`' script, MyDNS will be compiled with built-in debug messages. You can then run MyDNS with the '`-d, --debug`' flag, and it will output copious amounts of debugging information. If you are trying to debug, do not run MyDNS as a daemon, as the debugging information will not be output.

Appendix B Configuration

The '`mydns.conf`' file has a simple, familiar format. It consists of lines that may contain variables and values, in the format

```
variable = value
```

Blank lines are allowed. The '`#`' character begins comments, which are ignored.

The '`--dump-config`' option of the `mydns` program will output all possible variables in '`mydns.conf`' format. (See Section 2.5 [Creating mydns.conf], page 3.)

Boolean values can be '`yes`', '`no`', '`1`', '`0`', '`on`', or '`off`'.

B.1 Database configuration

db-host *(string)* The hostname where your database server is located. May be overridden by the '`-h`' ('`--host`') command-line option.

db-user *(string)* The username to provide to the database server during authentication. May be overridden by the '`-u`' ('`--user`') command-line option.

db-password
(string) The password to provide to the database server during authentication. May be overridden by the '`-p`' ('`--password`') command-line option.

database *(string)* The name of the database containing DNS data.

B.2 Name daemon configuration

user *(string)* Run with the permissions of this user.

group *(string)* Run with the permissions of this group.

listen *(string)* Listen and accept requests on this address only. If this is '`*`', the server will accept connections on all addresses. This must be an IP address in numbers-and-dots format, or '`*`'. Multiple addresses may be specified, as a comma-delimited list of addresses or on separate '`listen`' lines. To specify a port other than port 53, append '`:port`' to the address.

no-listen *(string)* Do *not* listen on this address. This must be an IP address in numbers-and-dots format, or '`*`'. Multiple addresses may be specified, as a comma-delimited list of addresses or on separate '`no-listen`' lines. To specify a port other than port 53, append '`:port`' to the address. This option may be used to easily tell MyDNS not to listen on the address '`127.0.0.1`', on which you are running a recursive name server.

B.3 Cache configuration

zone-cache-size
: *(integer)* The number of items stored in the DNS server's internal zone data cache. Set this to '`0`' to disable the zone data cache entirely. (See Section 4.1 [Caching], page 11.)

zone-cache-expire
: *(integer)* Number of seconds after which cached items expire. If this is '`0`', the zone data cache is not used. The TTL value for any RR may override this value if it is a shorter amount of time. (See Section 4.1 [Caching], page 11.)

reply-cache-size
: *(integer)* The number of items stored in the DNS server's internal reply cache. Set this to '`0`' to disable the reply cache entirely. (See Section 4.1 [Caching], page 11.)

reply-cache-expire
: *(integer)* Number of seconds after which cached replies expire. If this is '`0`', the reply cache is not used. (See Section 4.1 [Caching], page 11.)

B.4 Miscellaneous configuration options.

log *(string)* The name daemon should log via the syslog facility specified, which may be '`LOG_DAEMON`' or any of '`LOG_LOCAL0`' through '`LOG_LOCAL7`'. If the argument is '`stderr`' or '`stdout`', program output will go to that stream only. If the argument is a filename, program output will go to that file.

pidfile *(string)* The `mydns` program will write its PID to this file on startup.

timeout *(integer)* Number of seconds after which queries should time out.

multicpu *(integer)* Number of processors in your system.

allow-axfr *(boolean)* Should DNS-based zone transfers be enabled?

allow-tcp *(boolean)* Should TCP queries be allowed? Use of this option is usually not recommended. However, TCP queries should be enabled if you think your server will be serving out answers larger than 512 bytes.

allow-update
: *(boolean)* Should RFC 2136 DNS UPDATE queries be allowed? (See Section 4.5 [DNS UPDATE], page 12.)

ignore-minimum
: *(boolean)* Should MyDNS ignore the minimum TTL specified in the SOA record for each zone?

soa-table *(string)* Name of the table containing SOA records.

rr-table *(string)* Name of the table containing resource record data.

soa-where *(string)* An additional SQL '`WHERE`' clause to use when retrieving records from the `soa` table (see Section 3.1 [soa table], page 4).

rr-where *(string)* An additional SQL '`WHERE`' clause to use when retrieving records from the `rr` table (see Section 3.2 [rr table], page 5).

recursive *(string)* Forward recursive requests to a DNS server at this address and return its response to the client.

Appendix C References

RFC 1034 Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, November 1987.

RFC 1035 Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987.

RFC 1183 Everhart, C., et. al., "New DNS RR Definitions", RFC 1183, October 1990.

RFC 1537 Beertema, P., "Common DNS Data File Configuration Errors", RFC 1537, October 1993.

RFC 1700 Reynolds, J. and Postel, J., "Assigned Numbers", STD 2, RFC 1700, October 1994.

RFC 2136 Vixie, P., et. al., "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 1537, April 1997.

RFC 2317 Eidnes, H., et. al., "Classless IN-ADDR.ARPA delegation", BCP 20, RFC 1537, March 1998.

RFC 2782 Gulbrandsen, et al., "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

RFC 2915 Mealling, M. and Daniel, R., "The Naming Authority Pointer (NAPTR) DNS Resource Record", RFC 2915, September 2000.

Appendix D GNU Free Documentation License

Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgments" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant

Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgments", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this

License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

D.0.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) *year your name*.
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.1
> or any later version published by the Free Software Foundation;
> with the Invariant Sections being *list their titles*, with the
> Front-Cover Texts being *list*, and with the Back-Cover Texts being *list*.
> A copy of the license is included in the section entitled ‘‘GNU
> Free Documentation License’’.

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being *list*"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

R

S

T

U

V

W

X

Z