NAME

host − query nameserver about domain names and zones

SYNOPSIS

host [**−v**] [**−a**] [**−t** *querytype*] [*options*] *name* [*server*]
host [**−v**] [**−a**] [**−t** *querytype*] [*options*] **−l** *zone* [*server*]
host [**−v**] [*options*] **−H** [**−D**] [**−E**] [**−G**] *zone*
host [**−v**] [*options*] **−C** *zone*
host [**−v**] [*options*] **−A** *host*

host [*options*] **−x** [*name* ...]
host [*options*] **−X** *server* [*name* ...]

OPTION SYNTAX

Besides the traditional short options (one letter with single dash, and an optional value as separate argument), there are now also long options in the format **−−keyword**[=*value*]. Many (but not all) short options have a long equivalent. There are several long options without a short equivalent. The long options are not yet documented in this manual page, but a summary of the existing long options, and the mapping to their short alternative, is available via the command **host −−help**.

DESCRIPTION

host looks for information about Internet hosts and domain names. It gets this information from a set of interconnected servers that are spread across the world. The information is stored in the form of "resource records" belonging to hierarchically organized "zones".

By default, the program simply converts between host names and Internet addresses. However, with the **−t**, **−a** and **−v** options, it can be used to find all of the information about domain names that is maintained by the domain nameserver system. The information printed consists of various fields of the associated resource records that were retrieved.

The arguments can be either host names (domain names) or numeric Internet addresses.

A numeric Internet address consists of four decimal numbers separated by dots, e.g. **192.16.199.1**, representing the four bytes of the 32-bit address.
The default action is to look up the associated host name.

A host name or domain name consists of component names (labels) separated by dots, e.g.
nikhefh.nikhef.nl
The default action is to look up all of its Internet addresses.

For single names without a trailing dot, the local domain is automatically tacked on the end. Thus a user in domain "nikhef.nl" can say "host nikhapo", and it will actually look up "nikhapo.nikhef.nl". In all other cases, the name is tried unchanged. Single names with trailing dot are considered top-level domain specifications, e.g. "nl."

Note that the usual lookup convention for any name that does not end with a trailing dot is to try first with the local domain appended, and possibly other search domains. (As of BIND 4.9, names that have embedded dots but no trailing dot are first tried ‘‘as is’’ before appending search domains) This convention is not used by this program.

The actual suffix to tack on the end is usually the local domain as specified in the **/etc/resolv.conf** file, but this can be overridden. See below for a description of how to customize the host name lookup.

ARGUMENTS

The first argument is normally the host name (domain name) for which you want to look up the requested information. If the first argument is an Internet address, a query is done on the special "reverse mapping" domain to look up its associated host name.

If the **−l** option is given, the first argument is a domain zone name for which a complete listing is given. The program enters a special zone listing mode which has several variants (see below).

The second argument is optional. It allows you to specify a particular server to query. If you don’t specify

this argument, default servers are used, as defined by the **/etc/resolv.conf** file.

EXTENDED SYNTAX

If the **−x** option is given, it extends the syntax in the sense that multiple arguments are allowed on the command line. An optional explicit server must now be specified using the **−X** option as it cannot be given as an ordinary argument any more. The **−X** option implies **−x**.

The extended syntax allows no arguments at all, in which case the arguments will be read from standard input. This can be a pipe, redirection from a file, or an interactive terminal. Note that these arguments are the names to be queried, and not command options. Everything that appears after a '#' or ';' on an input line will be skipped. Multiple arguments per line are allowed.

OPTIONS

There are a number of options that can be used before the specified arguments. Some of these options are meaningful only to the people who maintain the domain database zones. The first options are the regularly used ones.

−v causes printout to be in a "verbose" format. All resource record fields are printed. Without this option, the ttl and class fields are not shown. Also the contents of the "additional information" and "authority information" sections in the answer from the nameserver are printed, if present. Normally these sections are not shown. In addition, the verbose option prints extra information about the various actions that are taken by the program. Note that **−vv** is "very verbose". This generates a lot of output.

−t *querytype*
allows you to specify a particular type of resource record information to be looked up. Supported types are listed below. The wildcard may be written as either **ANY** or *****. Types may be given in upper or lower case. The default is type **A** for regular lookups, and **A**, **NS**, and **PTR** for zone listings.

−a is equivalent to **−t ANY**. Note that this gives you "anything available" (currently cached) and not "all defined data" if a non-authoritative server is queried.

SPECIAL MODES

The following options put the program in a special mode.

−l *zone*
generates the listing of an entire zone.

E.g. the command
host −l nikhef.nl
will give a listing of all hosts in the "nikhef.nl" zone. The **−t** option is used to filter what information is extracted, as you would expect. The default is address information from A records, supplemented with data from PTR and NS records.

The command
host −Z −a −l nikhef.nl
will give a complete download of the zone data for "nikhef.nl", in the official master file format.

−H can be specified instead of the **−l** option. It will print the count of the unique hostnames (names with an A record) encountered within the zone. It will not count pseudo names like "localhost", nor addresses associated with the zone name itself. Neither are counted the "glue records" that are necessary to define nameservers for the zone and its delegated zones.

By default, this option will not print any resource records.

Combined with the **−S** option, it will give a complete statistics survey of the zone.

The host count may be affected by duplicate hosts (see below). To compute the most realistic value, subtract the duplicate host count from the total host count.

−G implies **−H**, but lists the names of gateway hosts. These are the hosts that have more than one address. Gateway hosts are not checked for duplicate addresses.

−E implies **−H**, but lists the names of extrazone hosts. An extrazone host in zone "foo.bar" is of the form "host.xxx.foo.bar" where "xxx.foo.bar" is not defined as a delegated zone with an NS record. This may be intentional, but also may be an error.

−D implies **−H**, but lists the names of duplicate hosts. These are hosts with only one address, which is known to have been defined also for another host with a different name, possibly even in a different zone. This may be intentional, but also may be an error.

−C can be specified instead of the **−l** option. It causes the SOA records for the specified zone to be compared as found at each of the authoritative nameservers for the zone (as listed in the NS records). Nameserver recursion is turned off, and it will be checked whether the answers are really authoritative. If a server cannot provide an authoritative SOA record, a lame delegation of the zone to that server is reported. Discrepancies between the records are reported. Various sanity checks are performed.

−A enters a special address check mode.

If the first argument is a host name, its addresses will be retrieved, and for each of the addresses it will be checked whether they map back to the given host.

If the first argument is a dotted quad Internet address, its name will be retrieved, and it will be checked whether the given address is listed among the known addresses belonging to that host.

If the **−A** flag is specified along with any zone listing option, a reverse lookup of the address in each encountered A record is performed, and it is checked whether it is registered and maps back to the name of the A record. This applies to forward zones. For reverse in-addr.arpa zones, it is checked whether the target in PTR records maps to a canonical host name.

LISTING OPTIONS

The following options apply only to the special zone listing modes.

−L *level*
Recursively generate zone listings up to this level deep. Level 1 traverses the parent zone and all of its delegated zones. Each additional level descends into another layer of delegated zones.

−S prints statistics about the various types of resource records found during zone listings, the number of various host classifications, the number of delegated zones, and some total statistics after recursive listings.

−p causes only the primary nameserver of a zone to be contacted for zone transfers during zone listings. Normally, zone transfers are obtained from any one of the authoritative servers that responds. The primary nameserver is obtained from the SOA record of the zone. If a specific server is given on the command line, this option will query that server for the desired nameservers of the zone. This can be used for testing purposes in case the zone has not been registered yet.

−P *prefserver*
gives priority for zone transfers to preferred servers residing in domains given by the comma-separated list *prefserver*. The more domain component labels match, the higher the priority. If this option is not present, priority is given to servers within your own domain or parent domains. The order in which NS records are issued may be unfavorable if they are subject to BIND 4.9 round-robin reshuffling.

−N *skipzone*
prohibits zone transfers for the zones given by the comma-separated list *skipzone*. This may be used during recursive zone listings when certain zones are known to contain bogus information which should be excluded from further processing.

COMMON OPTIONS

The following options can be used in both normal mode and domain listing mode.

−d turns on debugging. Nameserver transactions are shown in detail. Note that **−dd** prints even more debugging output.

−f *filename*
writes the resource record output to the given logfile as well as to standard output.

−F *filename*
same as **−f**, but exchange the role of stdout and logfile. All stdout output (including verbose and debug printout) goes to the logfile, and stdout gets only the extra resource record output (so that it can be used in pipes).

−I *chars*
suppresses warning messages about illegal domain names containing invalid characters, by specifying such characters in the string *chars*. The underscore is a good candidate.

−i constructs a query for the "reverse mapping" **in-addr.arpa** domain in case a numeric (dotted quad) address was specified. Useful primarily for zone listing mode, since for numeric regular lookups such query is done anyway (but with −i you see the actual PTR resource record outcome).

−n constructs a query for the "reverse mapping" **nsap.int** domain in case an nsap address was specified. This can be used to look up the names associated with nsap addresses, or to list reverse nsap zones. An nsap address consists of an even number of hexadecimal digits, with a maximum of 40, optionally separated by interspersed dots. An optional prefix "0x" is skipped. If this option is used, all reverse nsap.int names are by default printed in forward notation, only to improve readability. The **−Z** option forces the output to be in the official zone file format.

−q be quiet and suppress various warning messages (the ones preceded by " !!! "). Serious error messages (preceded by " *** ") are never suppressed.

−Q selects quick mode, in which several potentially time consuming special checks are not carried out, and statistics gathering is skipped if not explicitly selected.

−T prints the time-to-live values during non-verbose output. By default the ttl is shown only in verbose mode.

−Z prints the selected resource record output in full zone file format, including trailing dot in domain names, plus ttl value and class name.

OTHER OPTIONS

The following options are used only in special circumstances.

−c *class*
allows you to specify a particular resource record class. Supported are **IN**, **INTERNET**, **CS**, **CSNET**, **CH**, **CHAOS**, **HS**, **HESIOD**, and the wildcard **ANY** or *****. The default class is **IN**.

−e excludes information about names that are not residing within the given zone during zone listings, such as some glue records. For regular queries, it suppresses the printing of the "additional information" and "authority information" sections in the answer from the nameserver.

−m is equivalent to **−t MAILB**, which filters any of types **MB**, **MR**, **MG**, or **MINFO**. In addition, **MR** and **MG** records will be recursively expanded into **MB** records.

−o suppresses the resource record output to stdout. Can be used in combination with the **−f** option to separate the resource record output from verbose and debug comments and error messages.

−r causes nameserver recursion to be turned off in the request. This means that the contacted nameserver will return only data it has currently cached in its own database. It will not ask other servers to retrieve the information. Note that nameserver recursion is always turned off when checking SOA records using the **−C** option. Authoritative servers should have all relevant information available.

−R Normally querynames are assumed to be fully qualified and are tried as such, unless it is a single name, which is always tried (and only once) in the default domain. This option simulates the default BIND behavior by qualifying any specified name by repeatedly adding search domains, with the exception that the search terminates immediately if the name exists but does not have the desired

querytype. The default search domains are constructed from the default domain by repeatedly peeling off the first component, until a final domain with only one dot remains.

−s *seconds*
specifies a new nameserver timeout value. The program will wait for a nameserver reply in two attempts of this number of seconds. Normally it does 2 attempts of 5 seconds per nameserver address tried. The actual timeout algorithm is slightly more complicated, extending the timeout value dynamically depending on the number of tries and the number of nameserver addresses.

−u forces the use of virtual circuits (TCP) instead of datagrams (UDP) when issuing nameserver queries. This is slower, but potentially more reliable. Note that a virtual circuit is automatically chosen in case a query exceeds the maximum datagram packet size. Also if a datagram answer turns out to be truncated, the query is retried using virtual circuit. A zone transfer is always done via a virtual circuit.

−w causes the program to retry forever if the response to a regular query times out. Normally it will time out after some 10 seconds per nameserver address tried.

−V prints just the version number of the **host** program, and exits.

SPECIAL OPTIONS

The following options are used only in special circumstances.

−O *srcaddr*
Define an explicit source IP address for sending nameserver queries. This may be necessary for multihomed hosts with asymmetric routing policy.

−j *minport* **−J** *maxport*
Define a range of explicit port numbers to be assigned to the source IP address of the client socket for sending the nameserver queries and receiving the replies. Normally the kernel chooses a random free port number. This may be an inappropriate number if you are behind a firewall that filters random port numbers on incoming traffic.
If only one of **−j** or **−J** is given, a single explicit port number is defined. This is ok for UDP queries, but may not be sufficient for TCP queries.

DEFAULT OPTIONS

Default options and parameters can be preset in an environment variable **HOST_DEFAULTS** using the same syntax as on the command line. They will be evaluated before the command line arguments.

QUERYTYPES

The following querytypes (resource record types) are supported. Indicated within parentheses are the various kinds of data fields.

A	Host address (dotted quad)
NS	Authoritative nameserver (domain name)
MD	Mail destination (domain name)
MF	Mail forwarder (domain name)
CNAME	Canonical name for an alias (domain name)
SOA	Marks the start of a zone of authority (domain name of primary, domain name of hostmaster, serial, refresh, retry, expiration, default ttl)
MB	Mailbox domain name (domain name)
MG	Mail group member (domain name)
MR	Mail rename domain name (domain name)
NULL	Null resource record (no format or data)
WKS	Well-known service description (dotted quad, protocol name, list of services)
PTR	Domain name pointer (domain name)

HINFO Host information (CPU type string, OS type string)

MINFO Mailbox or mail list information (request domain name, error domain name)

MX Mail exchanger (preference value, domain name)

TXT Descriptive text (one or more strings)

UINFO User information (string)

UID User identification (number)

GID Group identification (number)

UNSPEC Unspecified binary data (data)

ANY Matches information of any type available.

MAILB Matches any of types **MB**, **MR**, **MG**, or **MINFO**.

MAILA Matches any of types **MD**, or **MF**.

The following types have been defined in RFC 1183, but are not yet in general use. They are recognized by this program.

RP Responsible person (domain name for MB, domain name for TXT)

AFSDB AFS database location (type, domain name)

X25 X25 address (address string)

ISDN ISDN address (address string, optional subaddress string)

RT Route through host (preference value, domain name)

The following types have been defined in RFC 1348, but are not yet in general use. They are recognized by this program. RFC 1348 has already been obsoleted by RFC 1637 and RFC 1706, which defines a new experimental usage of NSAP records. This program has now hooks to manipulate them.

NSAP NSAP address (encoded address)

NSAP-PTR
NSAP pointer (domain name)

The following are new types as per RFC 1664 and RFC 1712. Note that the GPOS type has been withdrawn already, and has been superseded by the LOC type.

PX X400 to RFC822 mapping (preference value, rfc822 domain, x400 domain)

GPOS Geographical position (longitude string, latitude string, altitude string)

The following types have been reserved in RFC 1700, and are defined in RFC 2065 and revised per RFC 2035.

SIG Security signature

KEY Security key

NXT Next valid record

The IP v6 address architecture and DNS extensions are defined in RFC 1884 and RFC 1886.

AAAA IP v6 address (address spec with colons)

The following type is documented in RFC 1876.

LOC Geographical location (latitude, longitude, altitude, precision)

The following types have been proposed, but are still in draft.

EID Endpoint identifier

NIMLOC Nimrod locator

ATMA ATM address

The following type is defined per RFC 2168.

NAPTR Naming authority URN

The following type is proposed in RFC 2052, updated by RFC 2782.

SRV Internet service information

The following type is proposed in RFC 2230.

KX Key exchanger (preference value, domain name)

The following type is defined in RFC 2538.

CERT

The following types have been proposed, but are still in draft.

A6

DNAME

SINK

The following type is defined in RFC 2671.

OPT

EXAMPLES

A very good summary and validation of an entire zone can be obtained with the following command:

host −G −S −C −A −L 1 *zone*

DIAGNOSTICS

FAILURE MESSAGES

The following messages are printed to show the reason of failure for a particular query. The name of an explicit server, if specified, may be included. If a special class was requested, it is also shown.

Nameserver [*server*] not running
The contacted server host does not have a nameserver running.

Nameserver [*server*] not responding
The nameserver at the contacted server host did not give a reply within the specified time frame.

Nameserver [*server*] not reachable
The network route to the intended server host is blocked.

name does not exist [at *server*] (Authoritative answer)
The queryname does definitely not exist at all.

name does not exist [at *server*], try again
The queryname does not exist, but the answer was not authoritative, so it is still undecided.

name has no *type* record [at *server*] (Authoritative answer)
The queryname is valid, but the specified type does not exist. This status is here returned only in case authoritative.

name type record currently not present [at *server*]
The specified type does not exist, but we don't know whether the queryname is valid or not. The answer was not authoritative. Perhaps recursion was off, and no data was cached locally.

name type record not found [at *server*], try again
Some intermediate failure, e.g. timeout reaching a nameserver.

name type record not found [at *server*], server failure
Some explicit nameserver failure to process the query, due to internal or forwarding errors. This may also be returned if the zone data has expired at a secondary server, of when the server is not authoritative for some class.

name type record not found [at *server*], no recovery
Some irrecoverable format error, or server refusal.

name type record query refused [by *server*]
The contacted nameserver explicitly refused to answer the query. Some nameservers are configured to refuse zone transfer requests that come from arbitrary clients.

name type record not found [at *server*]
The exact reason for failure could not be determined. (This should not happen).

zone has lame delegation to *server*
If we query a supposedly authoritative nameserver for the SOA record of a zone, the information should be available and the answer should be authoritative. If not, a lame delegation is flagged. This is also done if the server turns out not to exist at all. Ditto if we ask for a zone transfer and the server cannot provide it.

No nameservers for *zone* found
It was not possible to retrieve the name of any nameserver for the desired zone, in order to do a zone transfer.

No addresses of nameservers for *zone* found
We got some nameserver names, but it was not possible to retrieve addresses for any of them.

No nameservers for *zone* responded
When trying all nameservers in succession to do a zone transfer, none of them were able or willing to provide it.

WARNING AND ERROR MESSAGES

Miscellaneous warning messages may be generated. They are preceded by " !!! " and indicate some non-fatal condition, usually during the interpretation of the retrieved data. These messages can be suppressed with the −q command line option.

Error messages are preceded by " *** " and indicate a serious problem, such as format errors in the answers to queries, but also major violations of the specifications. Those messages cannot be suppressed.

zone has only one nameserver *server*
When retrieving the nameservers for a zone, it appears that only one single nameserver exists. This is against the recommendations.

zone nameserver *server* is not canonical (*realserver*)
When retrieving the nameservers for a zone, the name of the specified server appears not to be canonical. This may cause serious operational problems. The canonical name is given between parentheses.

empty zone transfer for *zone* from *server*
The zone transfer from the specified server contained no data, perhaps only the SOA record. This could happen if we query the victim of a lame delegation which happens to have the SOA record in its cache.

extraneous NS record for *name* within *zone* from *server*
During a zone transfer, an NS record appears for a name which is not a delegated subzone of the current zone.

extraneous SOA record for *name* within *zone* from *server*
During a zone transfer, an SOA record appears for a name which is not the name of the current zone.

extraneous glue record for *name* within *zone* from *server*
During a zone transfer, a glue record is included for a name which is not part of the zone or its delegated subzones. This is done in some older versions of BIND. It is undesirable since unauthoritative,

or even incorrect, information may be propagated.

incomplete *type* record for *name*
: When decoding the resource record data from the answer to a query, not all required data fields were present. This is frequently the case for HINFO records of which only one of the two data field is encoded.

name has both NS and A records within *zone* from *server*
: An A record has been defined for the delegated zone *name*. This is signalled only during the transfer of the parent *zone*. It is not an error, but the overall hostcount may be wrong, since the A record is counted as a host in the parent zone. This A record is not included in the hostcount of the delegated zone.

name type record has zero ttl
: Resource records with a zero ttl value are special. They are not cached after retrieval from an authoritative nameserver.

name type records have different ttl within *zone* from *server*
: Resource records of the same name/type/class should have the same ttl value in zone listings. This is sometimes not the case, due to the independent definition of glue records or other information in the parent zone, which is not kept in sync with the definition in the delegated zone.

name type record has illegal name
: The name of an A or MX record contains invalid characters. Only alphanumeric characters and hyphen '-' are valid in components (labels) between dots.

name type host *server* has illegal name
: The name of an NS or MX target host contains invalid characters. Only alphanumeric characters and hyphen '-' are valid in components (labels) between dots.

name type host *server* does not exist
: The NS or MX target host *server* does not exist at all. In case of NS, a lame delegation of *name* to *server* is flagged. It also applies to the PTR target host in reverse zones.

name type host *server* has no A record
: The NS or MX target host *server* has no address. In case of NS, a lame delegation of *name* to *server* is flagged. It also applies to the PTR target host in reverse zones.

name type host *server* is not canonical
: The NS or MX target host *server* is not a canonical name. This may cause serious operational problems during domain data retrieval, or electronic mail delivery. It also applies to the PTR target host in reverse zones.

name type target *domain* does not exist
: The CNAME target *domain* does not exist at all.

name type target *domain* has no ANY record
: The CNAME target *domain* does not seem to have any associated resource record, although the name seems to exist.

name address *A.B.C.D* is not registered
: The reverse lookup of the address of an A record failed in an authoritative fashion. It was not present in the corresponding in-addr.arpa zone.

name address *A.B.C.D* maps to *realname*
: The reverse lookup of the address of an A record succeeded, but it did not map back to the name of the A record. There may be A records with different names for the same address. In the reverse in-addr.arpa zone there is usually only one PTR to the ''official'' host name.

name address *A.B.C.D* maps to alias *aliasname*
: In case of multiple PTR records, the first one encountered points to the ''official'' host name. Subsequent ones are returned as alias names via gethostbyaddr() as of BIND 4.9. Note that PTR records are exempt from round-robin reshuffling.

zone SOA record at *server* is not authoritative
When checking the SOA for a zone at one of its supposedly authoritative nameservers, the SOA information turns out to be not authoritative. This could be determined by making a query without nameserver recursion turned on.

zone SOA primary *server* is not advertised via NS
The primary nameserver is not among the list of nameservers retrieved via NS records for the zone. This is not an error per se, since only publicly accessible nameservers may be advertised, and others may be behind a fi rewall.

zone SOA primary *server* has illegal name
The name of the primary nameserver contains invalid characters.

zone SOA hostmaster *mailbox* has illegal mailbox
The name of the hostmaster mailbox contains invalid characters. A common mistake is to use an RFC822 email address with a ‘‘@’’, whereas the at-sign should have been replaced with a dot.

zone SOA serial has high bit set
Although the serial number is an unsigned 32-bit value, overfbw into the high bit can inadvertently occur by making inappropriate use of the dotted decimal notation in the zone fi le. This may lead to synchronization failures between primary and secondary servers.

zone SOA retry exceeds refresh
A failing refresh would be retried after it is time for the next refresh.

zone SOA refresh+retry exceeds expire
The retry after a failing refresh would be done after the data has already expired.

zone SOA expire is less than 1 week
The authoritative data at secondary servers expires after only one week of failing refresh attempts. This is probably a little too early under normal circumstances.

zone SOA expire is more than 6 months
Secondary servers will retry failing refresh attempts for a period of more than 6 months before their authoritative data expires. As BIND 8 concludes: war must have broken out.

server1 and *server2* have different primary for *zone*
If the SOA record is different, the zone data is probably different as well. What you get depends on which server you happen to query.

server1 and *server2* have different hostmaster for *zone*
If the SOA record is different, the zone data is probably different as well. What you get depends on which server you happen to query.

server1 and *server2* have different serial for *zone*
This is usually not an error, but happens during the period after the primary server has updated its zone data, but before a secondary performed a refresh. Nevertheless there could be an error if a mistake has been made in properly adapting the serial number.

server1 and *server2* have different refresh for *zone*
If the SOA record is different, the zone data is probably different as well. What you get depends on which server you happen to query.

server1 and *server2* have different retry for *zone*
If the SOA record is different, the zone data is probably different as well. What you get depends on which server you happen to query.

server1 and *server2* have different expire for *zone*
If the SOA record is different, the zone data is probably different as well. What you get depends on which server you happen to query.

server1 and *server2* have different defttl for *zone*

If the SOA record is different, the zone data is probably different as well. What you get depends on which server you happen to query.

EXIT STATUS

The program returns a zero exit status if the requested information could be retrieved successfully, or in case zone listings or SOA checks were performed without any serious error. Otherwise it returns a non-zero exit status.

ENVIRONMENT

CUSTOMIZING HOST NAME LOOKUP

In general, if the name supplied by the user does not have any dots in it, a default domain is appended to the end. This domain is usually defined in the **/etc/resolv.conf** file. If not, it is derived by taking the local host-name and taking everything after its first dot.

The user can override this, and specify a different default domain, by defining it in the environment variable *LOCALDOMAIN*.

In addition, the user can supply his own single-word abbreviations for host names. They should be in a file consisting of one line per abbreviation. Each line contains an abbreviation, white space, and then the fully qualified host name. The name of this file must be specified in the environment variable *HOSTALIASES*.

SPECIAL CONSIDERATIONS

The complete set of resource record information for a domain name is available from an authoritative nameserver only. Therefore, if you query another server with the "-a" option, only a subset of the data may be presented, since this option asks for any data that the latter server currently knows about, not all data that may possibly exist. Note that the "-v" option shows whether an answer is authoritative or not.

When listing a zone with the "-l" option, information will be fetched from authoritative nameservers for that zone. This is implemented by doing a complete zone transfer and then filtering out the information that you have asked for. Note that direct contact with such nameservers must be possible for this option to work. This option should be used with caution. Servers may be configured to refuse zone transfers if they are fboded with requests.

RELATED DOCUMENTATION

rfc819, Domain naming convention for internet applications
rfc883, Domain names - implementation and specification
rfc920, Domain requirements
rfc952, DOD Internet host table specification
rfc974, Mail routing and the domain system
rfc1032, Domain administrators guide
rfc1033, Domain administrators operations guide
rfc1034, Domain names - concepts and facilities
rfc1035, Domain names - implementation and specification
rfc1101, DNS encoding of network names and other types
rfc1122, Requirements for Internet hosts - comm. layers
rfc1123, Requirements for Internet hosts - application
rfc1183, New DNS RR definitions
rfc1348, DNS NSAP RRs
rfc1535, A security problem and proposed correction
rfc1536, Common DNS implementation errors
rfc1537, Common DNS data file configuration errors
rfc1591, Domain Name System structure and delegation
rfc1597, Address allocation for private internets
rfc1627, Network 10 considered harmful
rfc1637, DNS NSAP resource records
rfc1664, Using DNS to distribute X.400 address mappings
rfc1700, Assigned numbers
rfc1706, DNS NSAP resource records

rfc1712, DNS encoding of geographical location (GPOS)
rfc1713, Tools for DNS debugging
rfc1794, DNS support for load balancing
rfc1876, Expressing location information in the DNS (LOC)
rfc1884, IP v6 addressing architecture
rfc1886, DNS extensions to support IP v6 (AAAA)
rfc1912, Common DNS operational and confi guration errors
rfc1982, Serial number arithmetic
rfc1995, Incremental zone transfer in DNS (IXFR)
rfc1996, Prompt notifi cation of zone changes
rfc2010, Operational criteria for root nameservers
rfc2052, Specifi cation of location of services (SRV)
rfc2065, DNS security extensions (KEY/SIG/NXT)
rfc2136, Dynamic updates in the DNS
rfc2137, Secure DNS dynamic update
rfc2163, Using DNS to distribute global address mapping (PX)
rfc2168, Resolution of Uniform Resource Identifi ers (NAPTR)
rfc2181, Clarifi cations to the DNS specifi cation
rfc2230, Key exchange delegation record for the DNS (KX)
rfc2308, Negative cacheing of DNS queries
rfc2317, Classless in-addr.arpa delegation
rfc2535, DNS security extensions (KEY/SIG/NXT)
rfc2538, Storing certifi cates in the DNS (CERT)
rfc2541, DNS security operational considerations
rfc2671, Extension mechanisms for DNS (OPT)
rfc2782, Specifying the location of services (SRV)

AUTHOR

This program is originally from Rutgers University.
Rewritten by Eric Wassenaar, NIKHEF, <e07@nikhef.nl>

SEE ALSO

named(8), resolv.conf(5), resolver(3)