NAME

nstest – DNS test shell

SYNOPSIS

nstest [**–d**] [**–i**] [**–r**] [**–v**] [**–p** *port*] [*inet_addr* [*logfile*]]

DESCRIPTION

nstest is an interactive DNS test program. Queries are formed and sent by user command; any reply received is printed on the standard output. *inet_addr* is the Internet address of the DNS resolver to which **nstest** should send its queries. If *inet_addr* is not included, **nstest** first tries to contact a DNS server on the local host; if that fails, it tries the servers listed in the **/etc/resolv.conf** file. If a *logfile* is supplied, **nstest** uses it to log the queries sent and replies received.

OPTIONS

–d Causes **nstest** to create a file named **ns_packet.dump** (if it does not exist) and write into it a raw (binary) copy of each packet sent. If **ns_packet.dump** does exist, **nstest** will truncate it.

–i Sets the **RES_IGNTC** flag on the queries it makes. See **resolver**(3N) for a description of the **RES_IGNTC** flag.

–r Turns off the **RES_RECURSE** flag on the queries it makes. See **resolver**(3N) for a description of the **RES_RECURSE** flag.

–v Turns on the **RES_USEVC** and **RES_STAYOPEN** flags on the **res_send ( )** calls made. See **resolver**(3N) for a description of the **RES_USEVC** and **RES_STAYOPEN** flags.

–p Causes **nstest** to use the supplied *port* instead of the default name server port.

USAGE

When **nstest** starts, it prints a prompt (">") and waits for user input. DNS queries are formed by typing a *key letter* followed by the appropriate *argument*. Each *key letter* results in a call to **res_mkquery ( )** with *op* set to either **IQUERY** or **QUERY** and *type* set to one of the type values (defined in **<arpa/nameser.h>**). (Any other **key letter** than those listed below causes **nstest** to print a summary of the following table.)

Key Letter & Argument	Op	Type
a*host*	QUERY	T_A
A*addr*	IQUERY	T_A
B*user*	QUERY	T_MG
b*user*	QUERY	T_MB
c*host*	QUERY	T_CNAME
f*host*	QUERY	T_UINFO
G*gid*	IQUERY	T_GID
g*host*	QUERY	T_GID
h*host*	QUERY	T_HINFO
i*host*	QUERY	T_MINFO
M*host*	QUERY	T_MAILB
m*host*	QUERY	T_MX

n*host*	**QUERY**	**T_NS**
p*host*	**QUERY**	**T_PTR**
r*host*	**QUERY**	**T_MR**
s*host*	**QUERY**	**T_SOA**
T*host*	**QUERY**	**T_TXT**
U*uid*	**IQUERY**	**T_UID**
u*host*	**QUERY**	**T_UID**
w*host*	**QUERY**	**T_WKS**
x*host*	**QUERY**	**T_AXFR**

After the query is successfully formed, **res_send ( )** is called to send it and wait for a reply. **nstest** then prints the following on the standard output:

- a summary of the request and reply packets, including the **HEADER** structure (defined in **<arpa/nameser.h>**) used in the request
- the question being asked of the name server
- an enumeration of the name server(s) being polled
- a summary of the **HEADER** structure received in the reply
- the question the name server answered
- the answer itself

EXAMPLES

To fetch the address of host **playground.sun.com** from the Sun name server, the user would enter:

```
$ nstest 192.9.5.1
> aplayground.sun.com
```

nstest would return the following:

```
res_mkquery(0, playground.sun.com, 1, 1)
res_send()
HEADER:
    opcode = QUERY, id = 1, rcode = NOERROR
    header flags:  rd
    qdcount = 1, ancount = 0, nscount = 0, arcount = 0

QUESTIONS:
    playground.sun.com, type = A, class = IN

Querying server (# 1) address = 192.9.5.1
got answer:
HEADER:
    opcode = QUERY, id = 1, rcode = NOERROR
    header flags:  qr aa rd ra
    qdcount = 1, ancount = 1, nscount = 0, arcount = 0
```

```
QUESTIONS:
      playground.sun.com, type = A, class = IN

ANSWERS:
      playground.sun.com
      type = A, class = IN, ttl = 1 day, dlen = 4
      internet address = 192.9.5.5
```

To look up a PTR record, enter:

```
$ nstest 192.9.5.1
> p5.5.9.192.in-addr.arpa
```

nstest would return the following:

```
res_mkquery(0, 5.5.9.192.in-addr.arpa, 1, 12)
res_send()
HEADER:
      opcode = QUERY, id = 2, rcode = NOERROR
      header flags:  rd
      qdcount = 1, ancount = 0, nscount = 0, arcount = 0

QUESTIONS:
      5.5.9.192.in-addr.arpa, type = PTR, class = IN

Querying server (# 1) address = 192.9.5.1
got answer:
HEADER:
      opcode = QUERY, id = 2, rcode = NOERROR
      header flags:  qr aa rd ra
      qdcount = 1, ancount = 1, nscount = 0, arcount = 0

QUESTIONS:
      5.5.9.192.in-addr.arpa, type = PTR, class = IN

ANSWERS:
      5.5.9.192.in-addr.arpa
      type = PTR, class = IN, ttl = 7 hours 47 mins 2 secs, dlen = 23
      domain name = playground.sun.com
```

FILES

/usr/include/arpa/nameser.h	include file for implementation of DNS protocol
/usr/include/resolv.h	include file for the resolver daemon (in.named)

SEE ALSO

nslookup(1M), **resolver**(3N)