Commands

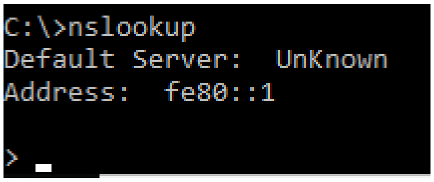
NSLookup

1. What is NSLookup?

   NSlookup is a useful suite of tools for looking at DNS records. While the ping command can only look at A records, the NSlookup command allows you to question your domains nameserver's, and find out much more information regarding your domains DNS.

2. Syntax

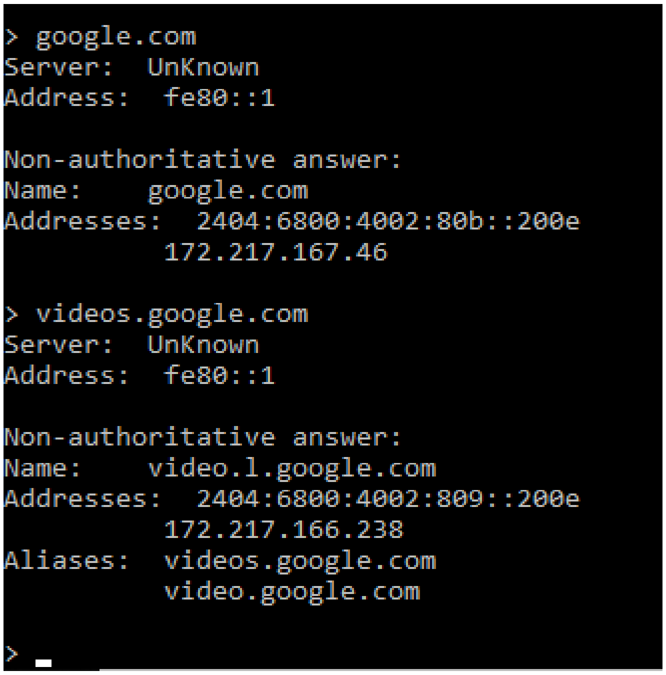
   To open the NSLookup tool, type the following at the command prompt and press <enter>.
   nslookup



```
C:\>nslookup
Default Server:  UnKnown
Address:  fe80::1

>
```

3. Querying a domain name

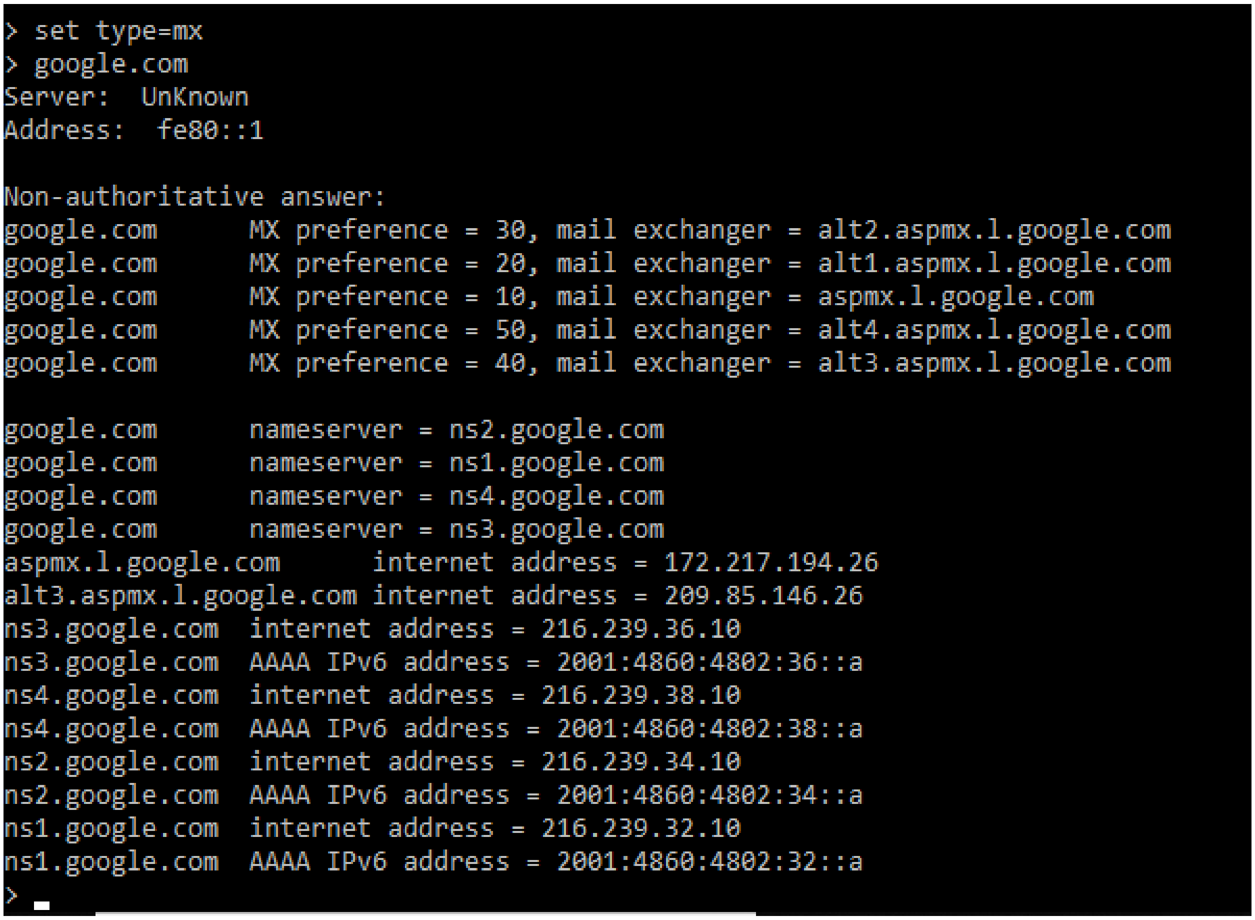
   To perform a DNS lookup, simply enter the domain or subdomain one would like to query and press <enter> on your keyboard.
   For example *google.com or videos.google.com.*



```
> google.com
Server:  UnKnown
Address:  fe80::1

Non-authoritative answer:
Name:    google.com
Addresses:  2404:6800:4002:80b::200e
          172.217.167.46

> videos.google.com
Server:  UnKnown
Address:  fe80::1

Non-authoritative answer:
Name:    video.l.google.com
Addresses:  2404:6800:4002:809::200e
          172.217.166.238
Aliases:  videos.google.com
          video.google.com

>
```

4. Changing the query type

The set type command will let you query a particular type of DNS record.
For example, if one wanted to check the MX (mail) records for a particular domain, he would enter the following command:
set type=mx
One can now perform another nslookup on the domain name. This time only MX records will be returned.
google.com



```
> set type=mx
> google.com
Server:  UnKnown
Address:  fe80::1

Non-authoritative answer:
google.com      MX preference = 30, mail exchanger = alt2.aspmx.l.google.com
google.com      MX preference = 20, mail exchanger = alt1.aspmx.l.google.com
google.com      MX preference = 10, mail exchanger = aspmx.l.google.com
google.com      MX preference = 50, mail exchanger = alt4.aspmx.l.google.com
google.com      MX preference = 40, mail exchanger = alt3.aspmx.l.google.com

google.com      nameserver = ns2.google.com
google.com      nameserver = ns1.google.com
google.com      nameserver = ns4.google.com
google.com      nameserver = ns3.google.com
aspmx.l.google.com      internet address = 172.217.194.26
alt3.aspmx.l.google.com internet address = 209.85.146.26
ns3.google.com  internet address = 216.239.36.10
ns3.google.com  AAAA IPv6 address = 2001:4860:4802:36::a
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  AAAA IPv6 address = 2001:4860:4802:38::a
ns2.google.com  internet address = 216.239.34.10
ns2.google.com  AAAA IPv6 address = 2001:4860:4802:34::a
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  AAAA IPv6 address = 2001:4860:4802:32::a
> _
```

In this example, the MX record for google.com points to
alt1.aspmx.l.google.com,
alt2.aspmx.l.google.com,
alt3.aspmx.l.google.com,
alt4.aspmx.l.google.com,
aspmx.l.google.com mail exchangers.

5. Changing the server

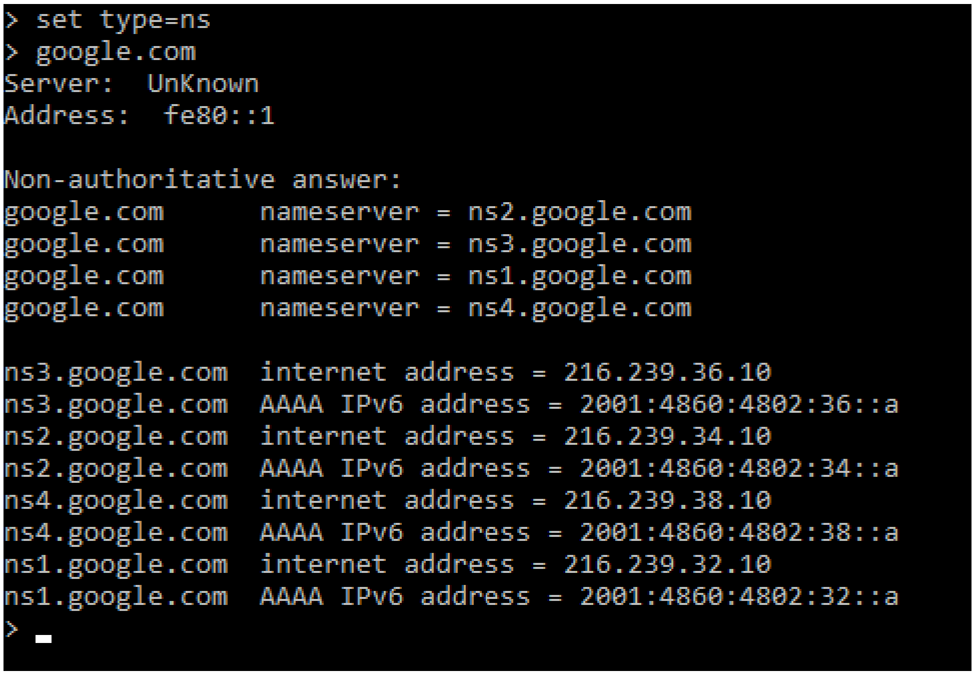
When machine first start NSLookup it will query its local DNS server. This is likely to be its router or Internet Service Provider's DNS servers.

As a result, one may not also receive accurate results, as the server, one is querying may not exist in its local DNS server. NSLookup allows it to change the nameserver, to ensure a nameserver from which one is guaranteed to get an accurate result.

If one query the nameserver listed against the domain name, it will receive an authoritative answer, because the nameserver has authority over the DNS for the domain name.

Start by retrieving the nameservers for the domain name by using the set type command and then querying the domain.

set type=ns
google.com



```
> set type=ns
> google.com
Server:  UnKnown
Address:  fe80::1

Non-authoritative answer:
google.com      nameserver = ns2.google.com
google.com      nameserver = ns3.google.com
google.com      nameserver = ns1.google.com
google.com      nameserver = ns4.google.com

ns3.google.com  internet address = 216.239.36.10
ns3.google.com  AAAA IPv6 address = 2001:4860:4802:36::a
ns2.google.com  internet address = 216.239.34.10
ns2.google.com  AAAA IPv6 address = 2001:4860:4802:34::a
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  AAAA IPv6 address = 2001:4860:4802:38::a
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  AAAA IPv6 address = 2001:4860:4802:32::a
> 
```

The results show that google.com has four nameserver's,
ns1.google.com,
ns2.google.com,
ns3.google.com and
ns4.google.com.

We can now use NSlookup to query one of those authoritative nameserver's for this domain name.

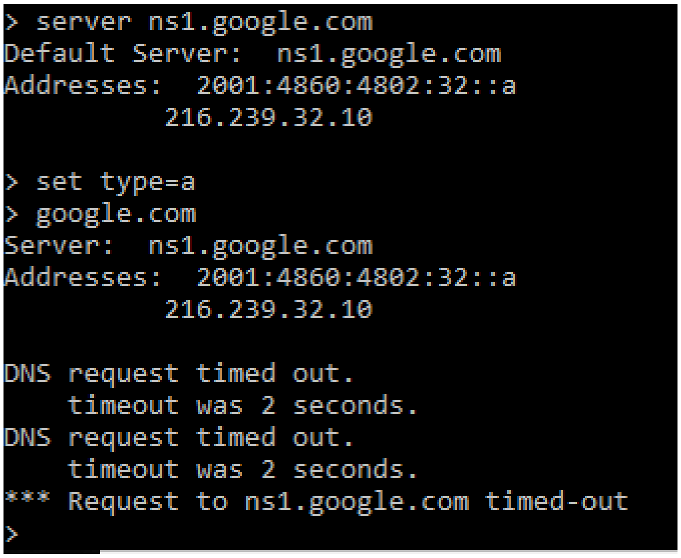
server ns1.google.com

Using set type, change the record type one wants to lookup (for example A record, MX record). In this example we shall retrieve the A records for the domain.

set type=a

And finally, query the domain name.

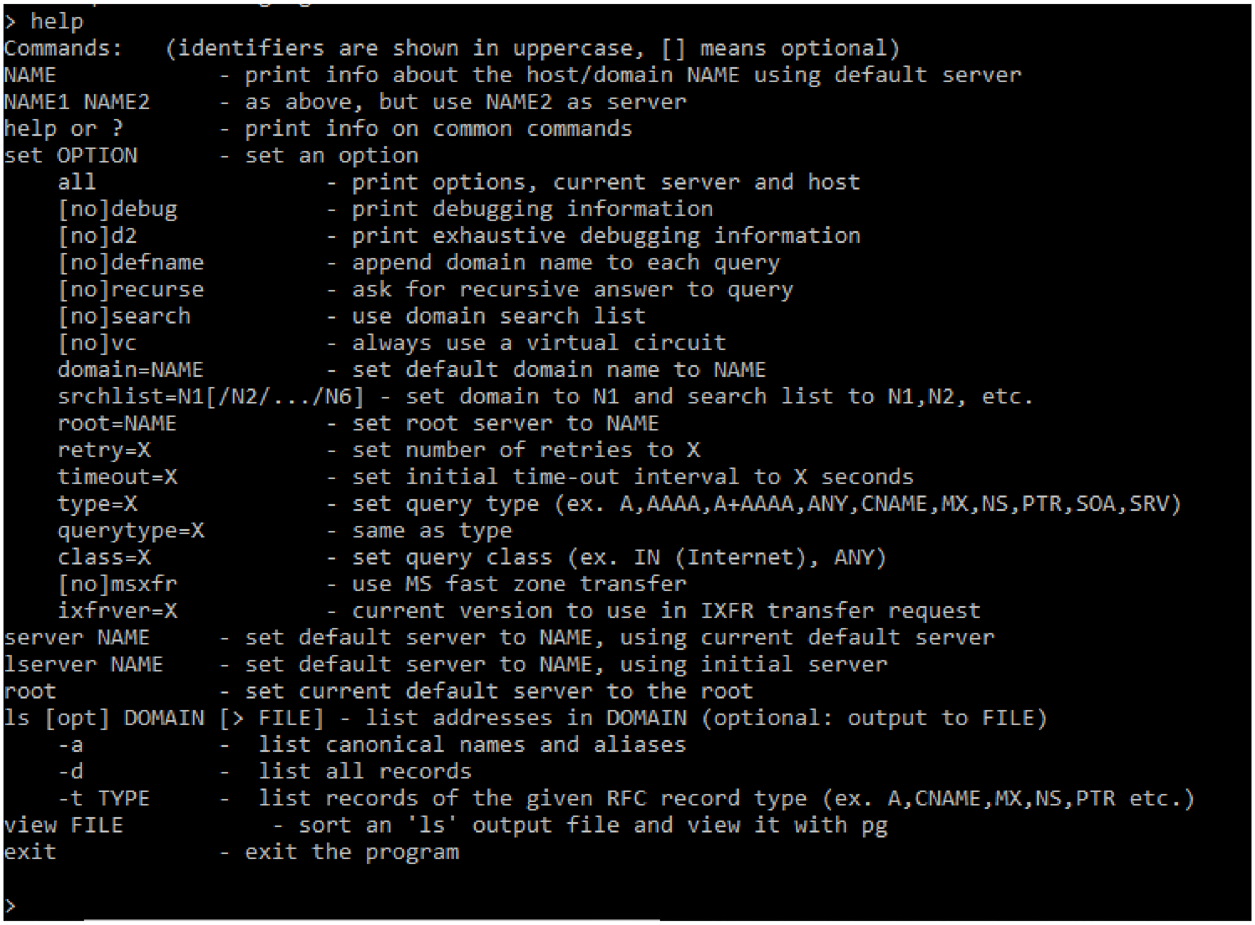
google.com



```
> server ns1.google.com
Default Server:  ns1.google.com
Addresses:  2001:4860:4802:32::a
          216.239.32.10

> set type=a
> google.com
Server:  ns1.google.com
Addresses:  2001:4860:4802:32::a
          216.239.32.10

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to ns1.google.com timed-out
>
```

These results show that the google*.com* A record has an IP address of *216.239.32.10* according to the nameserver *ns1.google.com.*

6. Other (help or ?)



```
> help
Commands:   (identifiers are shown in uppercase, [] means optional)
NAME            - print info about the host/domain NAME using default server
NAME1 NAME2     - as above, but use NAME2 as server
help or ?       - print info on common commands
set OPTION      - set an option
    all                 - print options, current server and host
    [no]debug           - print debugging information
    [no]d2              - print exhaustive debugging information
    [no]defname         - append domain name to each query
    [no]recurse         - ask for recursive answer to query
    [no]search          - use domain search list
    [no]vc              - always use a virtual circuit
    domain=NAME         - set default domain name to NAME
    srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
    root=NAME           - set root server to NAME
    retry=X             - set number of retries to X
    timeout=X           - set initial time-out interval to X seconds
    type=X              - set query type (ex. A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,SOA,SRV)
    querytype=X         - same as type
    class=X             - set query class (ex. IN (Internet), ANY)
    [no]msxfr           - use MS fast zone transfer
    ixfrver=X           - current version to use in IXFR transfer request
server NAME     - set default server to NAME, using current default server
lserver NAME    - set default server to NAME, using initial server
root            - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
    -a          -  list canonical names and aliases
    -d          -  list all records
    -t TYPE     -  list records of the given RFC record type (ex. A,CNAME,MX,NS,PTR etc.)
view FILE         - sort an 'ls' output file and view it with pg
exit            - exit the program

>
```