

TZWorks® DNS Query Utility (*dqu*) Users Guide



Abstract

dqu is a standalone, command-line tool used to perform DNS queries on any specified DNS server. *dqu* has the ability to allow one to pipe in many queries in one session. Output can be in either CSV or XML formats. *dqu* runs on Windows, Linux and macOS.

Copyright © TZWorks LLC

www.tzworks.com

Contact Info: info@tzworks.com

Document applies to v0.49 of *dqu*

Updated: Apr 25, 2025

Table of Contents

1	Introduction	2
2	How to Use <i>dqu</i>	2
2.1	Examples	3
2.1.1	If you have a domain name that you wish to resolve into an IP address:	3
2.1.2	Performing a query but showing the detail of the packet that was sent and received.	4
2.1.3	You have an IP address that you wish to resolve to a domain name	4
2.1.4	You and an Email domain and want to resolve the mail exchange IP address associated with the email domain.	5
3	Available Options	5
4	Authentication and the License File.....	7
4.1	<i>Limited</i> versus <i>Demo</i> versus <i>Full</i> in the tool's Output Banner.....	8
5	References	8

TZWorks® DNS Query Utility (*dqu*) Users Guide

Copyright © TZWorks LLC

Webpage: http://www.tzworks.com/prototype_page.php?proto_id=16

Contact Information: info@tzworks.com

1 Introduction

dqu is a command line tool used to perform (a) DNS queries to, and (b) display replies from, a specified DNS server. All the DNS functionality that **dqu** provides can be obtained from other built in tools. *nslookup* is a good example. **dqu** adds the capability for one to pipe queries into the tool via standard input. It also formats the output in either CSV (default) or XML. This allows easy viewing in excel or some other offline analysis tool.

The specific DNS query records that **dqu** can currently handle are class 1 (for Internet) and include the following types:

- 'A' - for resolving a host address to an IP address
- 'PTR' - for resolving IP address to domain name
- 'MX' - for resolving email exchange to IP address
- 'NS' - for resolving an authoritative name server
- 'CNAME' - for resolving a canonical name
- 'TXT' - for resolving to an associated text string

Each query is tunable to handle: (a) number of retry attempts, (b) timeout in seconds to wait for a reply, (c) delay in seconds for each successive query, if processing requests from a STDIN (standard input) pipe, and (d) a specified DNS server to send the query to.

2 How to Use *dqu*

There are two main ways to use *dqu* for getting results from a DNS server: (a) single item queries and (b) multiple queries via piping in entries via STDIN (standard input). These options are shown with examples in the **dqu** command line menu, shown below:

```

Administrator: Windows PowerShell

usage:
dqu -url <domain to lookup> -server <dns srv ip>
dqu -ip <ip addr to lookup> -server <dns srv ip>
dqu -mx <mail host to lookup> -server <dns srv ip>
dqu -ns <domain to lookup> -server <dns srv ip>
dqu -cname <domain to lookup> -server <dns srv ip>
dqu -txt <url to query> -server <dns srv ip>

These versions use STDIN for piping in input:
dqu -url_pipe -server <dns srv ip>
dqu -ip_pipe -server <dns srv ip>
dqu -mx_pipe -server <dns srv ip>
dqu -ns_pipe -server <dns srv ip>
dqu -cname_pipe -server <dns srv ip>
dqu -txt_pipe -server <dns srv ip>

Available options:
-timeout <in secs>
-retry <num>           = if we timeout, num of retries
-delay <secs>         = delay in secs between each query
-xml                  = output the data in xml format
-csv                  = output in comma separated value format
-verbose              = shows std::hex dump of packets (not in xml mode)
-no_whitespace        = remove whitespace between csv delimiter
-csv_separator "|"    = use a pipe char for csv separator
-dateformat mm/dd/yyyy = "yyyy-mm-dd" is the default
-timeformat hh:mm:ss  = "hh:mm:ss.xxx" is the default
-filter <*partial*>   = filters data from stdin using pipe

example of using std input and redirecting the output to a file
type hosts.txt | dqu -url_pipe -server 8.8.8.8 > results.txt

```

Single item queries

Multiple queries via piping

2.1 Examples

Below are examples of using **dqu**, while querying the public Google DNS resolver. For a list of public DNS resolvers, see reference 2 at the end of this readme.

2.1.1 If you have a domain name that you wish to resolve into an IP address:

For this example, we will query information about the domain URL “google.com” from the DNS server 8.8.8.8 and have the output put into CSV format. The following output was received.

```
run time: 10/07/2013 23:12:41 [UTC]
cmdline: dqu64 -url google.com -server 8.8.8.8 -csv
```

query	comment	status	id	DNSserv	date	time-UTC	type	tname	class	ttl	size	data
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.73
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.78
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.66
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.72
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.65
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.69
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.68
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.64
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.71
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.70
google.com.	answer	success	3d	8.8.8.8	10/7/2013	23:12:41.894	1	A	1	264	4	74.125.228.67

2.1.2 Performing a query but showing the detail of the packet that was sent and received.

```
run time: 10/07/2013 23:20:01 [UTC]
cmdline: dqu64 -url google.com -server 8.8.8.8 -verbose
```

query	comment	status	id	DNSserv	date	time-UTC	type	tname	class	ttl	size	data
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.96
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.99
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.103
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.98
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.97
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.100
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.105
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.104
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.110
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.102
google.com.	answer	success	0027	8.8.8.8	10/07/2013	23:20:01.031	1	A	1	44	4	74.125.228.101

```
-----
packet that was sent
0000 0000: 27 00 01 00 00 01 00 00 00 00 00 00 06 67 6f 6f '.....goo
0000 0010: 67 6c 65 03 63 6f 6d 00 00 01 00 01          gle.com.....

packet that was recvd
0000 0000: 27 00 81 80 00 01 00 0b 00 00 00 00 06 67 6f 6f '.....goo
0000 0010: 67 6c 65 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 gle.com.....
0000 0020: 00 01 00 00 00 2c 00 04 4a 7d e4 60 c0 0c 00 01 .....}..'
0000 0030: 00 01 00 00 00 2c 00 04 4a 7d e4 63 c0 0c 00 01 .....}].c....
0000 0040: 00 01 00 00 00 2c 00 04 4a 7d e4 67 c0 0c 00 01 .....}].g....
0000 0050: 00 01 00 00 00 2c 00 04 4a 7d e4 62 c0 0c 00 01 .....}].b....
0000 0060: 00 01 00 00 00 2c 00 04 4a 7d e4 61 c0 0c 00 01 .....}].a....
0000 0070: 00 01 00 00 00 2c 00 04 4a 7d e4 64 c0 0c 00 01 .....}].d....
0000 0080: 00 01 00 00 00 2c 00 04 4a 7d e4 69 c0 0c 00 01 .....}].i....
0000 0090: 00 01 00 00 00 2c 00 04 4a 7d e4 68 c0 0c 00 01 .....}].h....
0000 00a0: 00 01 00 00 00 2c 00 04 4a 7d e4 6e c0 0c 00 01 .....}].n....
0000 00b0: 00 01 00 00 00 2c 00 04 4a 7d e4 66 c0 0c 00 01 .....}].f....
0000 00c0: 00 01 00 00 00 2c 00 04 4a 7d e4 65          .....}].e
-----
```

2.1.3 You have an IP address that you wish to resolve to a domain name

```
run time: 10/07/2013 23:24:21 [UTC]
cmdline: dqu64 -ip 24.248.75.200 -server 8.8.8.8
```

query	comment	status	id	DNSserv	date	time-UTC	type	tname	class	ttl	size	data
200.75.248.24.in-addr.arpa.	answer	success	0036	8.8.8.8	10/07/2013	23:24:21.031	1	A	1	24	4	www1.cox.com.

2.1.4 You and an Email domain and want to resolve the mail exchange IP address associated with the email domain.

query	comment	status	id	DNSserv	date	time-UTC	type	tname	class	tth	size	data
gmail.com.	answer	success	2c	8.8.8.8	10/7/2013	23:29:21.297	15	MX	1	3600	32	10;alt1.gmail-smtp-in.l.google.com.
gmail.com.	answer	success	2c	8.8.8.8	10/7/2013	23:29:21.297	15	MX	1	3600	4	5;gmail-smtp-in.l.google.com.
gmail.com.	answer	success	2c	8.8.8.8	10/7/2013	23:29:21.297	15	MX	1	3600	9	40;alt4.gmail-smtp-in.l.google.com.
gmail.com.	answer	success	2c	8.8.8.8	10/7/2013	23:29:21.297	15	MX	1	3600	9	30;alt3.gmail-smtp-in.l.google.com.
gmail.com.	answer	success	2c	8.8.8.8	10/7/2013	23:29:21.297	15	MX	1	3600	9	20;alt2.gmail-smtp-in.l.google.com.

3 Available Options

Option	Description
-server	Specifies the DNS server's IP address to use as your DNS resolver. The syntax is: -server <DNS server IP address> .
-url	Lookup a URL. The syntax is: -url <domain to lookup> -server <DNS server IP address> .
-ip	Lookup an IP address. The syntax is: -ip <IP address to lookup> -server <DNS server IP address> .
-mx	Lookup a mail exchange record. The syntax is: -mx <mail host to lookup> -server <DNS server IP address> .
-ns	Lookup a name space. The syntax is: -ns <domain to lookup> -server <DNS server IP address> .
-cname	Lookup a cname record. The syntax is: -cname <domain to lookup> -server <DNS server IP address> .
-txt	DNS query with any text. The syntax is: -txt <URL to query> -server <DNS server IP address> .
-url_pipe	Same function as the -url option, but allows piping in URL requests via STDIN (standard input). Each request passed in is processed in sequence. The syntax is: -url_pipe -server <DNS server IP address> .

<i>-ip_pipe</i>	Same function as the <i>-ip</i> option, but allows piping in IP requests via STDIN (standard input). Each request passed in is processed in sequence. The syntax is: <i>-ip_pipe -server <DNS server IP address></i> .
<i>-mx_pipe</i>	Same function as the <i>-mx</i> option, but allows piping in MX requests via STDIN (standard input). Each request passed in is processed in sequence. The syntax is: <i>-mx_pipe -server <DNS server IP address></i> .
<i>-ns_pipe</i>	Same function as the <i>-ns</i> option, but allows piping in NS requests via STDIN (standard input). Each request passed in is processed in sequence. The syntax is: <i>-ns_pipe -server <DNS server IP address></i> .
<i>-cname_pipe</i>	Same function as the <i>-cname</i> option, but allows piping in CNAME requests via STDIN (standard input). Each request passed in is processed in sequence. The syntax is: <i>-cname_pipe -server <DNS server IP address></i> .
<i>-txt_pipe</i>	Same function as the <i>-txt</i> option, but allows piping in TXT requests via STDIN (standard input). Each request passed in is processed in sequence. The syntax is: <i>-txt_pipe -server <DNS server IP address></i> .
<i>-filter</i>	Filters data passed in via STDIN via the one of the pipe options. The syntax is <i>-filter <*partialname1* *partialname2* ...></i> . The wildcard character '*' is restricted to either before the name or after the name.
<i>-timeout</i>	Number of seconds to wait for a response before failing. The syntax is: <i>-timeout <number of seconds></i> .
<i>-retry</i>	Number of retries before failing. The syntax is: <i>-retry <number of times></i> .
<i>-delay</i>	Number of seconds to wait before each query (for commands that use piping). The syntax is: <i>-delay <number of seconds></i> .
<i>-verbose</i>	Display raw packet data as a hexadecimal dump. Not available if used with <i>-xml</i> option.
<i>-xml</i>	Output data in XML format.
<i>-csv</i>	Outputs the data fields delimited by commas. Since filenames can have commas, to ensure the fields are uniquely separated, any commas in the filenames get converted to spaces.
<i>-no_whitespace</i>	Used in conjunction with <i>-csv</i> option to remove any whitespace between the

	field value and the CSV separator.
<i>-csv_separator</i>	Used in conjunction with the <i>-csv</i> option to change the CSV separator from the default comma to something else. Syntax is <i>-csv_separator " "</i> to change the CSV separator to the pipe character.
<i>-dateformat</i>	Output the date using the specified format. Default behavior is <i>-dateformat "yyyy-mm-dd"</i> . Using this option allows one to adjust the format to mm/dd/yy, dd/mm/yy, etc. The restriction with this option is the forward slash (/) or dash (-) symbol needs to separate month, day and year and the month is in digit (1-12) form versus abbreviated name form.
<i>-timeformat</i>	Output the time using the specified format. Default behavior is <i>-timeformat "hh:mm:ss.xxx"</i> One can adjust the format to microseconds, via <i>"hh:mm:ss.xxxxxx"</i> or nanoseconds, via <i>"hh:mm:ss.xxxxxxxxxx"</i> , or no fractional seconds, via <i>"hh:mm:ss"</i> . The restrictions with this option is a colon (:.) symbol needs to separate hours, minutes and seconds, a period (.) symbol needs to separate the seconds and fractional seconds, and the repeating symbol 'x' is used to represent number of fractional seconds. (Note: the fractional seconds applies only to those time formats that have the appropriate precision available. The Windows internal filetime has, for example, 100 nsec unit precision available. The DOS time format and the UNIX 'time_t' format, however, have no fractional seconds). Some of the times represented by this tool may use a time format without fractional seconds, and therefore, will not show a greater precision beyond seconds when using this option.
<i>-utf8_bom</i>	All output is in Unicode UTF-8 format. If desired, one can prefix an UTF-8 <i>byte order mark</i> to the CSV output using this option.

4 Authentication and the License File

This tool has authentication built into the binary. The primary authentication mechanism is the digital X509 code signing certificate embedded into the binary (Windows and macOS).

The other mechanism is the runtime authentication, which applies to all the versions of the tools (Windows, Linux and macOS). The runtime authentication ensures that the tool has a valid license. The license needs to be in the same directory of the tool for it to authenticate. Furthermore, any modification to the license, either to its name or contents, will invalidate the license.

4.1 *Limited* versus *Demo* versus *Full* in the tool's Output Banner

The tools from *TZWorks* will output header information about the tool's version and whether it is running in *limited*, *demo* or *full* mode. This is directly related to what version of a license the tool authenticates with. The *limited* and *demo* keywords indicates some functionality of the tool is not available, and the *full* keyword indicates all the functionality is available. The lacking functionality in the *limited* or *demo* versions may mean one or all of the following: (a) certain options may not be available, (b) certain data may not be outputted in the parsed results, and (c) the license has a finite lifetime before expiring.

5 References

1. RFC 1035 – Domain Names – Implementation and specification.
2. Some available public DNS resolvers:
 - a. <http://code.google.com/speed/public-dns/> 8.8.8.8, 8.8.4.4
 - b. <http://www.dnsadvantage.com/> 156.154.70.1, 156.154.71.1
 - c. <http://www.opendns.com/> 208.67.222.222, 208.67.220.220
 - d. <http://www.nortondns.com/> 198.153.192.1, 198.153.194.1
 - e. <http://www.scrubit.com/> 67.138.54.100, 207.225.209.66 GTEI net 4.2.2.1 to 4.2.2.6
BellAltantic 151.197.0.38, 151.197.0.39
3. Windows command line utility: nslookup
4. Windows command line utility: ipconfig /displaydns = to display the DNS resolver cache
5. Windows command line utility: ipconfig /flushdns = to flush the DNS resolver cache