DNS Looking Glass: usage of the online service

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

First publication of this article on 20 January 2013. Last update on of 2 February 2013

http://www.bortzmeyer.org/dns-lg-usage.html

This text describes how to use the DNS Looking Glass <http://www.bortzmeyer.org/dns-lg.html> service hosted at http://dns.bortzmeyer.org/, http://dnslg.generic-nic.net/ or http://dns-lg.nlnetlabs.nl/ (and other future services using the same software <https://github.com/bortzmeyer/dns-lg>, although they may have different installation settings).

You use this program through REST requests (if you do not know REST, do not worry ; basically, it means we use ordinary HTTP requests, with structured output formats). The URL for the requests will be http://dns.bortzmeyer.org/$DOMAIN[/$TYPE][/$CLASS] where DOMAIN is the domain name and TYPE a DNS record type (such as AAAA or MX).

More formally, following the language of URI Templates (RFC 6570 [1]), the URLs of this service are http://dns.bortzmeyer.org/{+domain}/{querytype}/{queryclass}{?format,server,buffersize,dodn
(Or replace http://dns.bortzmeyer.org by another prefix such as http://dnslg.generic-nic.net or http://dns-lg.nlnetlabs.nl/.)

There is a non-standard pseudo-querytype ADDR to request both A and AAAA records, specially for the links in the HTML output. And you can use ANY to request any type the server may know about.

Some typical examples :

— MX records of ietf.org : <http://dns.bortzmeyer.org/ietf.org/MX>
— NAPTR records of de : <http://dns.bortzmeyer.org/de/NAPTR>
— TXT records of fr : <http://dns.bortzmeyer.org/fr/TXT>
— DNSKEY records of fr : <http://dns.bortzmeyer.org/fr/DNSKEY>

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6570.txt

Records containing Unicode are not a problem, as shown with <http://dns.bortzmeyer.org/mailclub.tel/TXT>.

You can use IDN, of course, and a funny example is <http://dns.bortzmeyer.org/%E2%98%81%E2%86%92%E2%9D%84%E2%86%92%E2%98%83%E2%86%92%E2%98%80%E2%86%92%E2%98%BA%E2%86%92%E2%98%82%E2%86%92%E2%98%B9%E2%86%92%E2%9C%9D.ws/SOA> (the percent-encoded URL is not very readable but the resulting Web page is better).

You can use this program from an ordinary Web browser, which will request the proper format. The program uses HTTP content negotiation to get the better output format. But, if you prefer, you can add manually the option `format=FORMAT` where FORMAT is XML, HTML, TEXT, ZONE or JSON. So, for instance, to get the IPv6 address of `www.example.com` in XML, it will be <http://dns.bortzmeyer.org/www.example.com/AAAA?format=XML>. If you use a command-line client like curl, you can add a HTTP header to indicate the format you want (here, JSON) :

```
% curl -s -H 'Accept: application/json' http://dns.bortzmeyer.org/xxx/SOA
```

The HTML option's format is not documented yet. It is mostly for human consumption, not for parsing by a program.

The XML output format follows partially Internet-Drafts draft-mohan-dns-query-xml <http://tools.ietf.org/id/draft-mohan-dns-query-xml> for the outer elements (plus some extensions) and of draft-daley-dns-schema <http://tools.ietf.org/id/draft-daley-dns-schema> for the resource data. (Note that the query format does **not** follow the first draft's syntax.)

The JSON option's format is documented in the file JSON.txt in the program's distribution.

The Text option's format is not documented. It is intended for human reading. If you need a structured format (to parse it from a program), use XML or JSON. If you prefer text-based formats, for instance for processing with common Unix command-line tools (awk, grep, etc), the best solution is probably the Zone format.

The Zone option's format follows section 5 of RFC 1035, with tabs as separators.

You can add an option to select the name server to query (the default one is chosen by the server, it is the default resolver(s) of the machine) : `server=IP-ADDRESS` (IP address only, names are **not** supported).

This option is necessary when you want to query the name and version of a name server. Funny examples are <http://dns.bortzmeyer.org/version.bind/TXT/CH?server=176.31.113.162> or <http://dns.bortzmeyer.org/version.bind/TXT/CH?server=78.193.86.178>.

To activate DNSSEC in the responses (to send the DO bit), use option `dodnssec=1` in the URL. This option will allow you to see the AD (Authentic Data) flag.

To use TCP (instead of UDP) for the request, use option `tcp=1` in the URL.

By default, the server queries the name servers with EDNS0 and a buffer size of 4096 bytes. To change that, use the option `buffersize` with the value you want. Setting it to 0 will disable EDNS.

For finding a domain name from an IP address, you can do requests with the `.arpa` domain name, for instance <http://dns.bortzmeyer.org/241.5.5.192.in-addr.arpa/PTR> but you can also use the option `reverse` to ask for the address to be turned into an arpa domain name, for instance <http://dns.bortzmeyer.org/192.5.5.241?reverse=1>.

To avoid abuse, there is a rate-limiter so, if you receive HTTP status code 429 (see RFC 6585), it means you have been too aggressive. Slow down your requests.

You can also use the DNS looking glass from a program : see the subdirectory usages/ of the distribution <https://github.com/bortzmeyer/dns-lg/tree/master/usages> for samples.

Note that the service does not use HTTPS today so you cannot really use it to detect censorship if the environment is really hostile (for instance if Aladeen mangles the DNS answers and also the unprotected HTTP traffic).