NAME

`metazone` — BIND configuration in a DNS zone

DESCRIPTION

A “metazone” is a DNS zone that describes the configuration of other DNS zones.

Metazones allow you to use standard DNS mechanisms - AXFR, IXFR, NOTIFY, UPDATE - to control the configuration of multiple name servers, instead of using a separate out-of-band distribution system.

The **`metazone`** program converts between metazones and `named.conf` fragments in either direction.

STRUCTURE

Apex

A metazone has apex SOA and NS resource records as required by the DNS protocol. The SOA record is used for zone refresh timing and to locate the primary master, in the usual way. The NS resource records will never be used and are just placeholders required by the protocol.

The SOA MINIMUM field is used for the version number of the metazone format. The format specified in this manual is 44.

Views

Immediately under the apex are a number of single-label subdomains. When the metazone is converted to `named.conf` syntax, the zone configurations are written to multiple files, one for each subdomain:

> `named.zones.`⟨view⟩

This allows you to have a single metazone containing the configuration for multiple views. (When you have **in-view** zones, the configurations for multiple views must be updated in sync.)

A number of view names are reserved for compatibility with other metazone formats:

allow-query **allow-transfer** **also-notify**
server-addresses **server-names** **servers**
in-view **masters** **zones**
view **zones**

Zones

Each zone configuration is represented in the metazone using one TXT resource record. The owner of the TXT record is

> ⟨*`zone`*⟩.⟨*`view`*⟩.⟨*`metazone`*⟩.

The TXT RDATA contains the zone’s configuration clause.

Note that if you need a configuration clause longer than 255 bytes, you can use one TXT record containing multiple strings. The strings are concatenated to form the configuration clause. You must not use multiple TXT records because you cannot control their order.

EXAMPLE

```
; This name for the metazone cannot collide with a real domain.
$ORIGIN _metazone.

; The required apex records.
$TTL    1h
@       SOA     localhost. hostmaster.localhost. (
                1436531696 1h 1h 1w 44 )
@       NS      localhost.
```

```
; Configuration for the "int" view
$ORIGIN int._metazone.

example.org    TXT    ( "{"
                        " type slave;"
                        " masters { ns-pri-int; };"
                        " allow-query { internal; };"
                        " allow-transfer { internal; };"
                        "};" )

; Configuration for the "ext" view
$ORIGIN ext._metazone.

example.org    TXT    ( "{"
                        " type slave;"
                        " masters { ns-pri-ext; };"
                        " allow-transfer { xfer-2nd; };"
                        " also-notify { notify-2nd; };"
                        "};" )
```

COMPATIBILITY

The metazone structure described in this manual is significantly different from the original metazone design by Paul Vixie.

We use Vixie's format numbering scheme. Vixie metazones have format number 42. Our metazones have format number 44. The reserved view names in our scheme are a superset of the keywords used by Vixie metazones.

Vixie metazones encode the configuration for a zone in multiple RRsets. This encoding requires a relatively complicated translated into a name server configuration. However, Vixie metazones are designed to be interoperable across multiple name server implementations.

Our metazones are specific to BIND, since they include literal named.conf snippets, and do not make any attempt to support other name server implementations.

SEE ALSO

metazone(1), nsnotifyd(1), named.conf(5).

Paul Vixie, *Federated domain name service using DNS metazones*, http://ss.vix.su/˜vixie/mz.pdf, June 2005.

AUTHOR

Tony Finch ⟨dot@dotat.at⟩ ⟨fanf2@cam.ac.uk⟩
at Cambridge University Information Services