Workgroup:	rpp
Internet-Draft:	draft-simmen-rpp-dns-data-00
Published:	17 June 2025
Intended Status:	Informational
Expires:	19 December 2025
Author:	C. Simmen *DENIC eG*

DNS data representation for use in RESTful Provisioning Protocol (RPP)

Abstract

This document proposes a representation for various DNS data for use in the RESTful Provisioning Protocol (RPP). Specified in JSON, the document describes common DNS record types used for domain provisioning as well as giving advice on how to adopt future record types.

EPP focused on distinct host objects containing data used for delegation purposes [RFC5732] and a separate extension focused on transferring DNSSEC relevant data [RFC5910]. Current registry system implementations improve these by grouping name servers into a nsset, or allowing domain provisioning without delegation. In addition new delegation mechanisms are developed [I-D.draft-ietf-deleg] to achieve a faster name resolution by providing properties of the child name server at delegation time.

Regardless of the specific use case all of the above data is meant to become visible in DNS. For this a structure close to the targeted system (DNS) makes it easy to adopt to current and future developments.

The RFC Editor will remove this note

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://github.com/christian-simmen/draft-simmen-rpp-dns-data. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-simmen-rpp-dns-data/.

Discussion of this document takes place on the WG Working Group mailing list (mailto:rpp@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/rpp/. Subscribe at https://www.ietf.org/mailman/listinfo/rpp/.

Source for this draft and an issue tracker can be found at https://github.com/christian-simmen/draft-simmen-rpp-dns-data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 December 2025.

Copyright Notice

Table of Contents

1. Introduction

In EPP host objects [RFC5732] are introduced. In the context of domain name service provisioning those objects are used as delegation information (NS) with optional GLUE (A) records. By the time of writing new transport protocols are used for DNS like DNS over HTTPS [RFC8484] or DNS over QUIC [RFC9250]. Along with this development the need for more fine grained delegation information is emerging. The DELEG record type [I-D.draft-ietf-deleg] can be seen as an example.

Apart from plain delegation information other DNS related data like DNSSEC information is common to be provisioned through EPP [RFC5910].

Some current registry system implementations are further improving the management of dns data. For example FRED (CZ.NIC) is grouping name servers into name server sets. RRI (DENIC) provides an option to provision a delegation-less domain by storing other DNS record types at the registry.

For all of the mentioned data is meant to be visible in DNS shifting from managing host objects to managing DNS data of a domain object will give an adavantage for adopting future resource record types as well covering current use cases.

2. Domain Names in DNS

DNS domain names are hierarchically ordered label separated by a dot ".". Each label represent the delegation of a subordinate namespace or a host name. DNS resource records [RFC1035] are expressed as a dataset containing:

"NAME" "CLASS" "TYPE" "TTL" "RDLENGTH" "RDATA"

A set of resource records describes the behavior of a namespace. Each resource record shares the same top level format.

NAME The owner name of the DNS entry which **MAY** be the domain itself or a subordinate hostname.

CLASS The RR CLASS

TYPE The RR TYPE of data present in the RDATA field.

TTL Time interval a RR may be cached by name servers

RDLENGTH The length of the RDATA field. RDLENGTH will be safely ignored in RPP

RDATA The actual payload data. Structures defer for each type.

3. JSON representation

3.1. Rules

3.1.1. DNS data extending an domain object

Delegation data, as well as DNSSEC data, is intended to find it's way into the parent side DNS servers. Because of the strong connection to the provisioned domain object and DNS servers both aspects should be visible in the RPP data model. Therefore the domain object is extended by an array of DNS entries. The properties of an object in this array **MUST** be a representation of the top level format as described in section 3.2.1 of [RFC1035]. All keys **MUST** be lowercase. Whitespaces **MUST** be translated to underscores ("_").

```
{
  "domain": "example.com",
  "dns": [
    {
      "name": "",
      "class": "",
      "type": "",
      "ttl": "",
      "rdata": {}
    }
  ]
}
```

3.1.2. DNS record structure representation

name

The owner name of the DNS entry which **MAY** be the domain itself or a subordinate hostname. A server **MUST NOT** accept a NAME which is not a subordinate label to the provisioned domain name.

A server **MUST** accept values as "@", "relative names" and fully qualified domain names (FQDN).

"@" **MUST** be interpreted as the provisioned domain name.

"relative names" **MUST** be appended by the server with the provisioned domain name.

"FQDN" identified by a trailing dot (".") **MUST NOT** be interpreted by the server. A server **MUST** check if the provided name is a subordinate to the provisioned domain, or the domain itself.

Example: ~~~~ { "domain": "example.com", "dns": [{ "name": "@", "type": "A", "rdata": { "address": "1.1.1.1" } }, { "name": "www", "type": "A", "rdata": { "address": "2.2.2.2" } }, { "name": "web.example.com.", "type": "A", "rdata": { "address": "3.3.3.3" } }] } ~~~~ would imply three resulting records: An A RR for "example.com" ("@") set to 1.1.1.1. An A RR for "www.example.com" ("www" relative) set to 2.2.2.2. An A RR for "web.example.com" (FQDN) set to 3.3.3.3.

3.1.2.1. class

A client **SHOULD** omit the class. The server **MUST** assume "IN" as class of a transferred dataset and **MAY** decline other values. If present the value **MUST** be chosen from section 3.2.4. CLASS values of [RFC1035].

3.1.2.2. type

The TYPE of data present in the RDATA. This also implies the expected fields in RDATA. If present the value **MUST** chosen from section 3.2.2. TYPE values of [RFC1035] or other RFC describing the RR TYPE.

3.1.2.3. ttl

A server **MUST** set a default value as TTL and **MAY** decline other values. A client **SHOULD** omit this value.

3.1.2.4. rdlength

RDLENGTH specifies the length of the RDATA field and will be ignored in RPP. A client **MUST NOT** include this field. A server **MUST** ignore this field if present.

3.1.2.5. rdata

The RDATA structure depends on the TYPE and **MUST** be expressed as a JSON object. Property names **MUST** follow the definition of the RDATA described by the corresponding RFC. Property names **MUST** be translated to lowercase. Whitespaces **MUST** be translated to underscores ("_").

Example: Section 3.3.11 NS RDATA format of [RFC1035] describes the RDATA of a NS RR as "NSDNAME". Section 3.3.9 MX RDATA format of [RFC1035] describes the RDATA of a MX RR as "PREFERENCE", "EXCHANGE". The resulting structure is therefore: ~~~~ { "domain": "example.com", "dns": [{ "name": "@", "type": "NS", "rdata": { "nsdname": "a.iana-servers.net." } }, { "name": "@", "type": "MX", "rdata": { "preference": "10", "exchange": "mx1.example.net" } }] } ~~~~

3.1.3. Additional controls

In addition to the regular data a server **MAY** allow a client to control specific operational behavior. A client **MAY** add an JSON object with a number of "controls" to the DNS dataset.

```
{
  "domain": "example.com",
  "dns": [
    {
      "name": "<name>",
      "type": "<type>",
      "rdata": {
        "rdata_key": "<rdata_value>",
      }
      "controls": {
        "<named_control>": "<named_control_value>"
      }
    }
  ]
}
```

3.1.4. Future DNS record types

Future record types **SHOULD** be added by breaking down the RDATA field specified by the RFC of the corresponding DNS record type.

3.2. Use cases

3.2.1. Domain delegation

To enable domain delegation a server **MUST** support the "NS", "A" and "AAAA" record types ([RFC1035],[RFC3596]).

A minimal delegation can be expressed by adding an array of name servers to the DNS data of a domain:

```
{
  "domain": "example.com",
  "dns": [
    {
      "name": "@",
      "type": "NS",
      "rdata": {
        "nsdname": "a.iana-servers.net."
      }
    },
    {
      "name": "@",
      "type": "NS",
      "rdata": {
        "nsdname": "b.iana-servers.net."
      }
    }
  ]
}
```

If GLUE records are needed the client may add records of type "A" or "AAAA" :

```
{
  "domain": "example.com",
  "dns": [
    {
      "name": "@",
      "type": "NS",
      "rdata": {
        "nsdname": "a.iana-servers.net."
      }
    },
    {
      "name": "@",
      "type": "NS",
      "rdata": {
        "nsdname": "ns.example.com"
      }
    },
    {
      "name": "ns.example.com.",
      "type": "A",
      "rdata": {
        "address": "1.2.3.4"
      }
    },
    {
      "name": "ns.example.com.",
      "type": "AAAA",
      "rdata": {
        "address": "dead::beef"
      }
    }
  ]
}
```

3.2.2. DNSSEC

To enable DNSSEC provisioning a server **SHOULD** support either "DS" or "DNSKEY" or both record types. The records **MUST** be added to the "dns" array of the domain. If provided with only "DNSKEY" a server **MUST** calculate the DS record. If both record types are provided a server **MAY** use the DNSKEY to validate the DS record.

```
    {
      "domain": "example.com",
      "dns": [
        {
          "name": "@",
          "type": "NS",
          "rdata": {
            "nsdname": "a.iana-servers.net."
          }
        },
        {
          "name": "@",
          "type": "NS",
          "rdata": {
            "nsdname": "b.iana-servers.net."
          }
        },
        {
          "name": "@",
          "type": "DS",
          "rdata": {
            "key_tag": "370",
            "algorithm": 13,
            "digest_type": 2,
            "digest":
"BE74359954660069D5C63D200C39F5603827D7DD02B56F120EE9F3A86764247C"
          }
        }
      ]
    }
```

```
    {
      "domain": "example.com.",
      "dns": [
        {
          "name": "@",
          "type": "NS",
          "rdata": {
            "nsdname": "a.iana-servers.net."
          }
        },
        {
          "name": "@",
          "type": "NS",
          "rdata": {
            "nsdname": "b.iana-servers.net."
          }
        },
        {
          "name": "@",
          "type": "DNSKEY",
          "rdata": {
            "flags": 257,
            "protocol": 3,
            "algorithm": 13,
            "public_key":
"kXKkvWU3vGYfTJGl3qBd4qhiWp5aRs7YtkCJxD2d+t7KXqwahww5IgJtxJT2yFItlggazyfXqJEV
OmMJ3qT0tQ=="
          }
        }
      ]
    }
```

3.2.3. Maximum signature lifetime

Maximum signature lifetime (maximum_signature_lifetime) describes the maximum number of seconds after signature generation a parents signature on signed DNS information should expire. The maximum_signature_lifetime value applies to the RRSIG resource record (RR) over the signed DNS RR. See Section 3 of [RFC4034] for information on the RRSIG resource record (RR).

A client **MAY** add maximum_signature_lifetime to the controls of an entry which is intended to be signed on the parent side. A server **MAY** ignore this value, e.g. for policy reasons.

```
    {
      "domain": "example.com",
      "dns": [
        {
          "name": "@",
          "type": "NS",
          "rdata": {
            "nsdname": "a.iana-servers.net."
          }
        },
        {
          "name": "@",
          "type": "NS",
          "rdata": {
            "nsdname": "b.iana-servers.net."
          }
        },
        {
          "name": "@",
          "type": "DS",
          "rdata": {
            "key_tag": "370",
            "algorithm": 13,
            "digest_type": 2,
            "digest":
"BE74359954660069D5C63D200C39F5603827D7DD02B56F120EE9F3A86764247C"
          },
          "controls": {
            "maximum_signature_lifetime": 86400
          }
        }
      ]
    }
```

3.2.4. Other DNS data

A server **MAY** support additional RR types, e.g. to support delegation-less provisioning.

```
{
  "domain": "example.com",
  "dns": [
    {
      "name": "@",
      "type": "A",
      "rdata": {
        "address": "1.2.3.4"
      }
    },
    {
      "name": "www.example.com.",
      "type": "A",
      "rdata": {
        "address": "1.2.3.4"
      }
    },
    {
      "name": "@",
      "type": "AAAA",
      "rdata": {
        "address": "dead::beef"
      }
    },
    {
      "name": "www.example.com.",
      "type": "A",
      "rdata": {
        "address": "dead::beef"
      }
    },
    {
      "name": "@",
      "type": "MX",
      "rdata": {
        "preference": "10",
        "exchange": "mx1.example.com"
      }
    },
    {
      "name": "mx1.example.com.",
      "type": "A",
      "rdata": {
        "address": "5.6.7.8"
      }
    },
    {
      "name": "@",
      "type": "MX",
      "rdata": {
        "preference": "20",
        "exchange": "mx2.example.net"
      }
    },
    {
      "name": "@",
      "type": "TXT",
```

```
      "rdata": {
        "txt_data": "v=spf1 -all"
      }
    }
  ]
}
```

4. Signaling supported record types

The server **MUST** provide a list of supported record types to the client.

5. Conventions and Definitions

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

6. Security Considerations

A server **SHOULD** choose the supported record types wisely and **MAY** restrict the number of accepted entries. Also see security considerations of [RFC4627].

7. IANA Considerations

This document has no IANA actions.

8. Appendix

9. References

9.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <https://www.rfc-editor.org/rfc/rfc4627>.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/rfc/rfc5730>.

[RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, DOI 10.17487/RFC5732, August 2009, <https://www.rfc-editor.org/rfc/rfc5732>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

9.2. Informative References

[I-D.draft-bortzmeyer-dns-json-01] Bortzmeyer, S., "JSON format to represent DNS data", Work in Progress, Internet-Draft, draft-bortzmeyer-dns-json-01, 25 February 2013, <https://datatracker.ietf.org/doc/html/draft-bortzmeyer-dns-json-01>.

[I-D.draft-ietf-deleg] April, T., Špaček, P., Weber, R., and Lawrence, "Extensible Delegation for DNS", Work in Progress, Internet-Draft, draft-ietf-deleg-00, 6 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-deleg-00>.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <https://www.rfc-editor.org/rfc/rfc3596>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/rfc/rfc4034>.

[RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, DOI 10.17487/RFC5910, May 2010, <https://www.rfc-editor.org/rfc/rfc5910>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/rfc/rfc8484>.

[RFC9250] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", RFC 9250, DOI 10.17487/RFC9250, May 2022, <https://www.rfc-editor.org/rfc/rfc9250>.

Acknowledgments

Author's Address

Christian Simmen
DENIC eG
Email: simmen@denic.de