Stream: Internet Engineering Task Force (IETF)
RFC: 9660
Category: Standards Track
Published: October 2024
ISSN: 2070-1721
Authors: H. Salgado *NIC Chile* M. Vergara *DigitalOcean* D. Wessels *Verisign*

# RFC 9660
# The DNS Zone Version (ZONEVERSION) Option

## Abstract

The DNS ZONEVERSION option is a way for DNS clients to request, and for authoritative DNS servers to provide, information regarding the version of the zone from which a response is generated. The SERIAL field from the Start of Authority (SOA) resource record (RR) is a good example of a zone's version, and it is the only one defined by this specification. Additional version types may be defined by future specifications.

Including zone version data in a response simplifies and improves the quality of debugging and diagnostics since the version and the DNS answer are provided atomically. This can be especially useful for zones and DNS providers that leverage IP anycast or multiple backend systems. It functions similarly to the DNS Name Server Identifier (NSID) option described in RFC 5001.

## Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9660.

## Copyright Notice

## Table of Contents

1. Introduction 3
1.1. Requirements Language 3
1.2. Terminology 4
2. The ZONEVERSION Option 4
2.1. Wire Format 4
2.2. Presentation Format 5
3. ZONEVERSION Processing 6
3.1. Initiators 6
3.2. Responders 6
3.2.1. Responding to Invalid ZONEVERSION Queries 6
3.2.2. ZONEVERSION Is Not Transitive 7
4. The SOA-SERIAL ZONEVERSION Type 7
4.1. Type SOA-SERIAL Presentation Format 7
5. Example Usage 7
6. IANA Considerations 8
6.1. DNS EDNS(0) Option Code Registration 8
6.2. ZONEVERSION TYPE Values Registry 8
6.2.1. Designated Expert Review Directives 9
7. Security Considerations 10
8. Normative References 10
9. Informative References 11
Appendix A. Implementation Considerations 11
Appendix B. Implementation References 11
Acknowledgements 12

Authors' Addresses 12

# 1. Introduction

The ZONEVERSION option allows DNS queriers to request, and authoritative DNS servers to provide, a token representing the version of the zone from which a DNS response was generated. It is similar to the NSID option [RFC5001], which can be used to convey the identification of a name server that generates a response.

The Domain Name System allows data to be loosely coherent [RFC3254], because synchronization can never be instantaneous, and some uses of DNS do not require strong coherency anyway. This means that a record obtained by one response could be out of sync with other authoritative sources of the same data at the same point in time. This can make it difficult to debug some problems when there is a need to couple the data with the version of the zone it came from. Furthermore, in today's Internet, it is common for high volume and important DNS zones to utilize IP anycast (Section 4.9 of [RFC4786]) and/or load-balanced backend servers. In general, there is no way to ensure that two separate queries are delivered to the same server. The ZONEVERSION option both simplifies and improves DNS monitoring and debugging by directly associating the data and the version together in a single response.

The SOA SERIAL field (Section 4.3.5 of [RFC1034]) is one example of zone versioning. Its purpose is to facilitate the distribution of zone data between primary and secondary name servers. It is also often useful in DNS monitoring and debugging. This document specifies the SOA SERIAL as one type of ZONEVERSION data.

Some DNS zones may use other distribution and synchronization mechanisms that are not based on the SOA SERIAL number, such as relational databases or other proprietary methods. In those cases, the SOA SERIAL field may not be relevant with respect to the versioning of its content. To accommodate these use cases, new ZONEVERSION types could be defined in future specifications. Alternatively, zone operators may use one of the Private Use ZONEVERSION code points allocated by this specification.

The ZONEVERSION option is **OPTIONAL** to implement by DNS clients and name servers. It is designed for use only when a name server provides authoritative response data. It is intended only for hop-to-hop communication and is not transitive.

## 1.1. Requirements Language

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.

### 1.2. Terminology

In this document, "original QNAME" is used to mean what the DNS terminology document [RFC9499] calls "QNAME (original)":

> The name actually sent in the Question section in the original query, which is always echoed in the (final) reply in the Question section when the QR bit is set to 1.

In this document, an "enclosing zone" of a domain name means a zone in which the domain name is present as an owner name or any parent of that zone. For example, if B.C.EXAMPLE and EXAMPLE are zones but C.EXAMPLE is not, the domain name A.B.C.EXAMPLE has B.C.EXAMPLE, EXAMPLE, and the root as enclosing zones.

## 2. The ZONEVERSION Option

This document specifies a new EDNS(0) [RFC6891] option, ZONEVERSION, which can be used by DNS clients and servers to provide information regarding the version of the zone from which a response is generated.

### 2.1. Wire Format

The ZONEVERSION option is encoded as follows:

OPTION-CODE for the ZONEVERSION option is 19.

OPTION-LENGTH for the ZONEVERSION option **MUST** have a value of 0 for queries and **MUST** have the value of the length (in octets) of the OPTION-DATA for responses.

OPTION-DATA for the ZONEVERSION option is omitted in queries. For responses, it is composed of three fields:

- an unsigned 1-octet Label Count (LABELCOUNT) indicating the number of labels for the name of the zone that VERSION value refers to
- an unsigned 1-octet type number (TYPE) distinguishing the format and meaning of VERSION
- an opaque octet string conveying the zone version data (VERSION)

```
                 +0 (MSB)                       +1 (LSB)
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 0: |           LABELCOUNT          |            TYPE               |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
 2: |                            VERSION                            |
    /                                                               /
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

*Figure 1: Diagram with the OPTION-DATA Format for the ZONEVERSION Option*

The LABELCOUNT field indicates the name of the zone that the ZONEVERSION option refers to, by means of taking the last LABELCOUNT labels of the original QNAME. For example, an answer with QNAME "a.b.c.example.com" and a ZONEVERSION option with a LABELCOUNT of value 2 indicates that the zone name in which this ZONEVERSION refers to is "example.com.".

In the case of a downward referral response, the LABELCOUNT number helps to differentiate between the parent and child zones, where the parent is authoritative for some portion of the QNAME above a zone cut. Also, if the ANSWER section has more than one RR set with different zones (like a CNAME and a target name in another zone), the number of labels in the QNAME disambiguates such a situation.

The value of the LABELCOUNT field **MUST NOT** count the null (root) label that terminates the original QNAME. The value of the LABELCOUNT field **MUST** be less than or equal to the number of labels in the original QNAME. The Root zone (".") has a LABELCOUNT field value of 0.

### 2.2. Presentation Format

The presentation format of the ZONEVERSION option is as follows:

The OPTION-CODE field **MUST** be represented as the mnemonic value ZONEVERSION.

The OPTION-LENGTH field **MAY** be omitted, but if present, it **MUST** be represented as an unsigned decimal integer.

The LABELCOUNT value of the OPTION-DATA field **MAY** be omitted, but if present, it **MUST** be represented as an unsigned decimal integer. The corresponding zone name **SHOULD** be displayed (i.e., LABELCOUNT labels of the original QNAME) for easier human consumption.

The TYPE and VERSION fields of the option **SHOULD** be represented according to each specific TYPE.

# 3. ZONEVERSION Processing

## 3.1. Initiators

A DNS client **MAY** signal its support and desire for zone version information by including an empty ZONEVERSION option in the EDNS(0) OPT pseudo-RR of a query to an authoritative name server. An empty ZONEVERSION option has OPTION-LENGTH set to zero.

A DNS client **SHOULD NOT** send the ZONEVERSION option to non-authoritative name servers.

A DNS client **MUST NOT** include more than one ZONEVERSION option in the OPT pseudo-RR of a DNS query.

## 3.2. Responders

A name server that (a) understands the ZONEVERSION option, (b) receives a query with the ZONEVERSION option, (c) is authoritative for one or more enclosing zones of the original QNAME, and (d) chooses to honor a particular ZONEVERSION request responds by including a TYPE and corresponding VERSION value in a ZONEVERSION option in an EDNS(0) OPT pseudo-RR in the response message.

Otherwise, a server **MUST NOT** include a ZONEVERSION option in the response.

A name server **MAY** include more than one ZONEVERSION option in the response if it supports multiple TYPEs. A name server **MAY** also include more than one ZONEVERSION option in the response if it is authoritative for more than one enclosing zone of the original QNAME. A name server **MUST NOT** include more than one ZONEVERSION option for a given TYPE and LABELCOUNT.

Note: the ZONEVERSION option should be included for any response satisfying the criteria above including, but not limited to, the following:

- Downward referral (see "Referrals" in Section 4 of [RFC9499]), even though the response's Authoritative Answer bit is not set. In this case, the ZONEVERSION data **MUST** correspond to the version of the referring zone.
- Name error (NXDOMAIN), even though the response does not include any Answer section RRs.
- NODATA (Section 3 of [RFC9499]), even though the response does not include any Answer section RRs.
- Server failure (SERVFAIL) when the server is authoritative for the original QNAME.

### 3.2.1. Responding to Invalid ZONEVERSION Queries

A name server that understands the ZONEVERSION option **MUST** return a FORMERR response when:

- The ZONEVERSION OPTION-LENGTH is not zero.

- More than one ZONEVERSION option is present.

#### 3.2.2. ZONEVERSION Is Not Transitive

The ZONEVERSION option is not transitive. A name server (recursive or otherwise) **MUST NOT** blindly copy the ZONEVERSION option from a query it receives into a subsequent query that it sends onward to another server. A name server **MUST NOT** send a ZONEVERSION option back to a client that did not request it.

## 4. The SOA-SERIAL ZONEVERSION Type

The first and only ZONEVERSION option TYPE defined in this document is a zone's serial number as found in the Start of Authority (SOA) RR.

As mentioned previously, some DNS zones may use alternative distribution and synchronization mechanisms that are not based on the SOA SERIAL number, and the SERIAL field may not be relevant with respect to the versioning of zone content. In those cases, a name server **SHOULD NOT** include a ZONEVERSION option with type SOA-SERIAL in a reply.

The value for this type is "0".

The mnemonic for this type is "SOA-SERIAL".

The EDNS(0) OPTION-LENGTH for this type **MUST** be set to "6" in responses.

The VERSION value for the SOA-SERIAL type **MUST** be a copy of the unsigned 32-bit SERIAL field of the SOA RR, as defined in Section 3.3.13 of [RFC1035].

### 4.1. Type SOA-SERIAL Presentation Format

The presentation format of this type content is as follows:

> The TYPE field **MUST** be represented as the mnemonic value "SOA-SERIAL".
>
> The VERSION field **MUST** be represented as an unsigned decimal integer.

## 5. Example Usage

A name server that (a) implements this specification, (b) receives a query with the ZONEVERSION option, (c) is authoritative for the zone of the original QNAME, and (d) utilizes the SOA SERIAL field for versioning of said zone should include a ZONEVERSION option in its response. In the response's ZONEVERSION option, the EDNS(0) OPTION-LENGTH would be set to 6 and the OPTION-DATA would consist of the 1-octet LABELCOUNT, the 1-octet TYPE with value 0, and the 4-octet SOA-SERIAL value.

The example below demonstrates expected output of a diagnostic tool that implements the ZONEVERSION option, displaying a response from a compliant authoritative DNS server:

```
$ dig @ns.example.com www.example.com aaaa +zoneversion \
+norecurse +nocmd

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7077
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; ZONEVERSION: 02 00 78 95 a4 e9 ("SOA-SERIAL: 2023073001 \
; (example.com.)")
;; QUESTION SECTION:
;www.example.com.    IN  AAAA

;; ANSWER SECTION:
www.example.com.  43200  IN  AAAA  2001:db8::80

;; AUTHORITY SECTION:
example.com.    43200  IN  NS  ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.    43200  IN  AAAA  2001:db8::53

;; Query time: 15 msec
;; SERVER: 2001:db8::53#53(2001:db8::53) (UDP)
;; WHEN: dom jul 30 19:51:04 -04 2023
;; MSG SIZE  rcvd: 129
```

*Figure 2: Example Usage and Dig Output*

# 6. IANA Considerations

## 6.1. DNS EDNS(0) Option Code Registration

This document defines a new EDNS(0) option, entitled "ZONEVERSION" (see Section 2), with the assigned value of 19 from the "DNS EDNS0 Option Codes (OPT)" registry:

| Value | Name | Status | Reference |
|---|---|---|---|
| 19 | ZONEVERSION | Standard | RFC 9660 |

*Table 1: DNS EDNS0 Option Codes (OPT) Registry*

## 6.2. ZONEVERSION TYPE Values Registry

IANA has created and will maintain a new registry called "ZONEVERSION TYPE Values" in the "Domain Name System (DNS) Parameters" registry group as follows:

| Range | Registration Procedures |
|---|---|
| 0-245 | Specification Required |
| 246-254 | Private Use |
| 255 | Reserved |

*Table 2: Registration Procedures for the ZONEVERSION TYPE Values Registry*

Initial values for the "ZONEVERSION TYPE Values" registry are given below; future assignments in the 1-245 values range are to be made through Specification Required [RFC8126]. Assignments consist of a TYPE value as an unsigned 8-bit integer recorded in decimal, a Mnemonic name as an uppercase ASCII string with a maximum length of 15 characters, and the required document reference.

| ZONEVERSION TYPE | Mnemonic | Reference |
|---|---|---|
| 0 | SOA-SERIAL | RFC 9660 |
| 1-245 | Unassigned | |
| 246-254 | Reserved for Private Use | RFC 9660 |
| 255 | Reserved | RFC 9660 |

*Table 3: ZONEVERSION TYPE Values Registry*

The change controller for this registry is IETF.

#### 6.2.1. Designated Expert Review Directives

The allocation procedure for new code points in the "ZONEVERSION TYPE Values" registry is Specification Required, thus review is required by a designated expert as stated in [RFC8126].

When evaluating requests, the expert should consider the following:

- Duplication of code point allocations should be avoided.
- A Presentation Format section should be provided with a clear code point mnemonic.
- The referenced document and stated use of the new code point should be appropriate for the intended use of a ZONEVERSION TYPE assignment. In particular, the reference should state clear instructions for implementers about the syntax and semantic of the data. Also, the length of the data must have proper limits.

The expert reviewing the request **MUST** respond within 10 business days.

## 7. Security Considerations

The EDNS extension data is not covered by RRSIG records, so there's no way to verify its authenticity nor integrity using DNSSEC, and it could theoretically be tampered with by a person in the middle if the transport is made by insecure means. Caution should be taken to use the EDNS ZONEVERSION data for any means besides troubleshooting and debugging.

If there's a need to certify the trustworthiness of ZONEVERSION, it will be necessary to use an encrypted and authenticated DNS transport, a transaction signature (TSIG) [RFC8945], or SIG(0) [RFC2931].

If there's a need to authenticate the data origin for the ZONEVERSION value, an answer with the SOA-SERIAL type as defined above could be compared to a separate regular SOA query with a DO flag, whose answer shall be DNSSEC signed. Since these are separate queries, the caveats about loose coherency already stated in the Introduction (Section 1) would apply.

With the SOA-SERIAL type defined above, there's no risk on disclosure of private information, as the SERIAL of the SOA record is already publicly available.

Please note that the ZONEVERSION option cannot be used for checking the correctness of an entire zone in a server. For such cases, the ZONEMD record [RFC8976] might be better suited for such a task. ZONEVERSION can help identify and correlate a specific answer with a version of a zone, but it has no special integrity or verification function besides a normal field value inside a zone, as stated above.

## 8. Normative References

**[RFC1034]** Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

**[RFC1035]** Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/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/info/rfc2119>.

**[RFC6891]** Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

**[RFC8126]** Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

**[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/info/rfc8174>.

## 9. Informative References

**[ImplRef]** "Zoneversion Implementations", commit f5f68a0, August 2023, <https://github.com/huguei/rrserial>.

**[RFC2931]** Eastlake 3rd, D., "DNS Request and Transaction Signatures ( SIG(0)s )", RFC 2931, DOI 10.17487/RFC2931, September 2000, <https://www.rfc-editor.org/info/rfc2931>.

**[RFC3254]** Alvestrand, H., "Definitions for talking about directories", RFC 3254, DOI 10.17487/RFC3254, April 2002, <https://www.rfc-editor.org/info/rfc3254>.

**[RFC4786]** Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786, December 2006, <https://www.rfc-editor.org/info/rfc4786>.

**[RFC5001]** Austein, R., "DNS Name Server Identifier (NSID) Option", RFC 5001, DOI 10.17487/RFC5001, August 2007, <https://www.rfc-editor.org/info/rfc5001>.

**[RFC8945]** Dupont, F., Morris, S., Vixie, P., Eastlake 3rd, D., Gudmundsson, O., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", STD 93, RFC 8945, DOI 10.17487/RFC8945, November 2020, <https://www.rfc-editor.org/info/rfc8945>.

**[RFC8976]** Wessels, D., Barber, P., Weinberg, M., Kumari, W., and W. Hardaker, "Message Digest for DNS Zones", RFC 8976, DOI 10.17487/RFC8976, February 2021, <https://www.rfc-editor.org/info/rfc8976>.

**[RFC9499]** Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/info/rfc9499>.

## Appendix A. Implementation Considerations

With very few exceptions, EDNS(0) option values in a response are independent of the queried name. This is not the case for ZONEVERSION, so its implementation may be more or less difficult, depending on how EDNS(0) options are handled in the name server.

## Appendix B. Implementation References

There is a patched NSD server (version 4.7.0) with support for ZONEVERSION as well as live test servers installed for compliance tests. Also, there is a client command "dig" with added zoneversion support, along with test libraries in Perl, Python, and Go. See [ImplRef] for more information.

## Acknowledgements

The authors are thankful for all the support and comments made in the DNSOP Working Group mailing list, chats, and discussions. A special thanks for suggestions to generalize the option using a registry of types from Petr Špaček and Florian Obser, suggestions for implementation from Stéphane Bortzmeyer, clarifications on security from George Michaelson, zone name disambiguation from Joe Abley and Brian Dickson, and reviews from Tim Wicinski and Peter Thomassen.

## Authors' Addresses

**Hugo Salgado**
NIC Chile
Miraflores 222, piso 14
CP 8320198 Santiago
Chile
Phone: +56 2 29407700
Email: hsalgado@nic.cl

**Mauricio Vergara Ereche**
DigitalOcean
101 6th Ave
New York, NY 10013
United States of America
Email: mvergara@digitalocean.com

**Duane Wessels**
Verisign
12061 Bluemont Way
Reston, VA 20190
United States of America
Phone: +1 703 948-3200
Email: dwessels@verisign.com
URI: https://verisign.com