

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 25 December 2025

J.G. Gould
D.K. Keathley
VeriSign, Inc.
M. Loffredo
IIT-CNR/Registro.it
23 June 2025

Versioning in the Registration Data Access Protocol (RDAP)
draft-ietf-regext-rdap-versioning-03

Abstract

This document describes an RDAP extension for an extensible set of versioning types with the features of identifying the RDAP extension versions supported by the server, the RDAP extension versions included in an RDAP response, and enabling a client to specify the desired RDAP extension versions to include in the RDAP query and RDAP response.

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 25 December 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	4
3. Versioning RDAP Features	4
3.1. Extension Version Identifier	4
3.2. Extension Versioning Request	5
3.2.1. Versioning Query Parameter	6
3.2.2. Versioning Extensions Media Type Parameter	6
3.3. Versioning RDAP Response	7
3.3.1. RDAP Conformance	7
3.3.2. "versioning_help" Member	8
3.3.3. "versioning_data" Member	15
4. Versioning Types	20
4.1. Opaque Versioning	20
4.1.1. Opaque Extension Version Identifier	21
4.1.2. Opaque Versioning "versioning_help" Member	21
4.1.3. Opaque Versioning "versioning_data" Member	23
4.2. Semantic Versioning	23
4.2.1. Semantic Extension Version Identifier	24
4.2.2. Semantic Versioning "versioning_help" Member	25
4.2.3. Semantic Versioning "versioning_data" Member	27
5. Versioning Considerations	27
5.1. Versioning Extension Considerations	27
5.2. Versioning Type Considerations	28
6. Extension Version Identifier	28
7. IANA Considerations	29
7.1. RDAP Extensions Registry	29
7.2. RDAP JSON Values Registry	29
8. Security Considerations	30
9. Implementation Status	30
9.1. IIT-CNR/Registro.it RDAP Server	30
10. Acknowledgements	30
11. References	30
11.1. Normative References	30
11.2. Informative References	32
Appendix A. Change History	32
A.1. Change from 00 to 01	32
A.2. Change from 01 to 02	32

A.3. Change from 02 to REGEXT 00	33
A.4. Change from REGEXT 00 to REGEXT 01	33
A.5. Change from REGEXT 01 to REGEXT 02	33
A.6. Change from REGEXT 02 to REGEXT 03	34
Authors' Addresses	34

1. Introduction

RDAP supports identifying the RDAP extensions included in an RDAP response using the RDAP Conformance data structure, defined in Section 4.1 of [RFC9083]. The RDAP Conformance values are identifiers with no standard mechanism to support structured, machine-parseable version signaling by the server. This document describes an RDAP extension for an extensible set of versioning types, with the pre-defined versioning types of Opaque Versioning (Section 4.1) and Semantic Versioning (Section 4.2), that have the following features:

- * Enabling a client to specify the desired RDAP extension versions to include in the RDAP query and RDAP response, using the Extension Versioning Request (Section 3.2). The client can specify the desired RDAP extension versions to include in the RDAP query and RDAP response with the "versioning" query parameter or the RDAP-X media type "extensions" parameter [I-D.ietf-regext-rdap-x-media-type].
- * Identifying the RDAP extension versions supported by the server. The RDAP extension versions supported by the server are returned in an extension to the RDAP help response, defined in Section 7 of [RFC9083], using the "versioning_help" member (Section 3.3.2). The "versioning_help" member (Section 3.3.2) includes the list of supported extensions and additional extension information, such as the available set of extension versions and the extension version document.
- * Identifying the RDAP extension versions included in an RDAP response. The RDAP extension versions included in an RDAP response are identified using the "versioning_data" member (Section 3.3.3). The "versioning_data" member (Section 3.3.3) includes a list of extension versions included in the RDAP response that extends the extension identifiers in the RDAP Conformance to include versioning information.

There are two predefined versioning types that include:

- * Opaque Versioning, as defined in Section 8 of [I-D.ietf-regext-rdap-extensions], to support RDAP extensions that are identified using the Extension Identifier included in the RDAP Conformance.
- * Semantic Versioning to support RDAP extensions that include a major and minor version with the Extension Identifier.

The versioning types are registered in the JSON Values Registry, where new versioning types can be defined and registered in the future. The specification of a versioning type defines how the base set of features of the versioning extension are used for the versioning type, such as the "type" value, the format of the Extension Version Identifier, the supported "versioning_help" members, and the supported "versioning" members.

2. Conventions Used in This Document

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.

3. Versioning RDAP Features

The extension includes a base set of features that can be used for any defined versioning type. The features enable the client to discover what extension versions are supported by the server, enable the client to specify what extension versions are desired in an RDAP query and RDAP response, and provide the extension versions included in the RDAP response by the server. Examples are provided using the two predefined versioning types in Section 4. The format of the Extension Version Identifier (Section 3.1) is general enough to support a broad range of future versioning types.

3.1. Extension Version Identifier

The Extension Version Identifier adds support for versioning to the Extension Identifier in [RFC7480] that is returned in the RDAP Conformance data structure of [RFC9083]. Each Versioning Type will define the scheme and the meaning of the "versioning" Augmented Backus-Naur Form (ABNF) grammar [RFC5234] rule. The Extension Version Identifier is used to uniquely identify a version of an RDAP extension in the Extension Versioning Request (Section 3.2), the members of the "versioning_help" Member (Section 3.3.2), and the members of the "versioning_data" Member (Section 3.3.3).

```
extension-version-identifier = identifier versioning
identifier = ALPHA *( ALPHA / DIGIT / "_" ) ; Extension Identifier
versioning = [ "-" 1*VCHAR]
```

Figure 1: Extension Version Identifier ABNF

Example Extension Version Identifiers:

```
"versioning"
    Opaque versioning of the "versioning" extension.

"versioning-0.4"
    Semantic versioning of the "versioning" extension.

"semantic_ext1-1.0"
    Semantic versioning of the "semantic_ext1" extension.

"opaque_ext1"
    Opaque versioning of the "opaque_ext1" extension.

"dated_ext1-20241129"
    Use of potential dated versioning with the format YYYYMMDD.
```

3.2. Extension Versioning Request

The client MAY provide an Extension Versioning Request to indicate the desired extension versions to include in the RDAP query and RDAP response. There are two Extension Versioning Request methods with the Versioning Query Parameter (Section 3.2.1) and the Versioning Extensions Media Type Parameter (Section 3.2.2). The server MUST support both methods of Extension Versioning Request methods and the client MUST use at most a single Extension Versioning Request method in the RDAP query.

The Extension Versioning Request provides a hint from the client what extension versions to include in the RDAP query by the client and RDAP response by the server. It is up to server policy what extension versions to support in the RDAP query and the extension versions to include in the RDAP response, based on the supported set of extensions, the default extension versions, and the hint provided by the client using the Extension Versioning Request. The server MUST ignore an unsupported extension version in constructing the response.

3.2.1. Versioning Query Parameter

The "versioning" query parameter MAY be used with RDAP queries to specify the RDAP extension versions to include in the RDAP query and the RDAP response. The "versioning" query parameter is a list of one or more Extension Version Identifiers, as defined in Section 3.1, separated by commas. The ABNF format for the "versioning" query parameter, using the "extension-version-identifier" rule from Figure 1, is:

```
versioning = "versioning=" extension-version-identifier
            *( "," extension-version-identifier)
```

Figure 2: Versioning Query Parameter ABNF

Example RDAP queries using the "versioning" query parameter:

`https://example.com/rdap/help?versioning=versioning-0.4`
Help query using extension "versioning-0.4". The response can be found in Figure 6.

`https://example.com/rdap/domain/`
`versioning.example?versioning=semantic_ext1-0.1`
Domain query using extension "semantic_ext1-0.1". The response can be found in Figure 9.

`https://example.com/rdap/domain/`
`versioning.example?versioning=semantic_ext1-0.1,opaque_ext2`
Domain query using extensions "semantic_ext1-0.1" and "opaque_ext2". The response can be found in Figure 10.

3.2.2. Versioning Extensions Media Type Parameter

The Extensions Media Type Parameter [I-D.ietf-regext-rdap-x-media-type] MAY be used with RDAP queries to specify the RDAP extension versions to include in the RDAP query and the RDAP response. The RDAP extensions included in the "extensions" parameter are Extension Version Identifiers, as defined in Section 3.1. The "extensions" Versioning Extensions Type Parameter is a list of one or more Extension Version Identifiers, as defined in Section 3.1, separated by whitespace (%x20). The use of the hyphen ("-") character in the Extension Version Identifier ABNF (Figure 1) ensures that an Extension Version Identifier with a non-empty "versioning" ABNF rule will not conflict with an Extension Identifier, since hyphens are prohibited in the Extension Version Identifier ABNF (Figure 1). The ABNF format for the "extensions" parameter, using the "extension-version-identifier" rule from Figure 1, is:

```
extensions = "extensions="
  DQUOTE
  extension-version-identifier
  *(SP extension-version-identifier)
  DQUOTE
```

Figure 3: Versioning Extensions Media Type Parameter ABNF

Example Versioning Extensions Media Type Parameter values:

```
extensions="versioning-0.4"
```

Versioning Extensions Media Type Parameter for the "versioning" extension with the semantic versioning of "versioning-0.4". The response for the /help query "https://example.com/rdap/help" with the Versioning Extensions Media Type Parameter value can be found in Figure 6.

```
extensions="semantic_ext1-0.1"
```

Versioning Extensions Media Type Parameter for the "semantic_ext1" extension with the semantic versioning of "semantic_ext1-0.1". The domain query "https://example.com/rdap/domain/versioning.example" response with the Versioning Extensions Media Type Parameter value can be found in Figure 9.

```
extensions="semantic_ext1-0.1 opaque_ext2"
```

Versioning Extensions Media Type Parameter for the "semantic_ext1" and "opaque_ext2" extensions with the semantic versioning of "semantic_ext1-0.1" and opaque versioning of "opaque_ext2", respectively. The domain query "https://example.com/rdap/domain/versioning.example" response with the Versioning Extensions Media Type Parameter value can be found in Figure 9.

3.3. Versioning RDAP Response

3.3.1. RDAP Conformance

RDAP responses that contain values described in this document MUST indicate conformance with this specification by including an "rdapConformance" ([RFC9083]) value of "versioning". The "versioning" extension identifier is described in Section 7.1.

Example "rdapConformance" member with the versioning extension:

```
"rdapConformance": [  
  "rdap_level_0",  
  "versioning"  
]
```

Figure 4: "rdapConformance" with Versioning Extension

3.3.2. "versioning_help" Member

The "versioning_help" member MUST be added to the response of a /help query, defined in [RFC9082], to indicate the extension versions that are supported by the server. This extends the information provided by the guidance in Section 7 of [I-D.ietf-regext-rdap-extensions], with inclusion of the all supported extension identifiers in the RDAP Conformance to the response of a /help query. The "versioning_help" member contains an array of extension objects with the following child members:

"extension": REQUIRED Extension Identifier, defined in [RFC7480], and returned in the "rdapConformance" data structure of [RFC9083]. An example is the "versioning" extension identifier.

"type": REQUIRED Versioning Type used by the extension. The Versioning Types are registered using the "versioning" type in the JSON Values Registry. There are two predefined Versioning Types, including "opaque" (see Opaque Versioning (Section 4.1)) and "semantic" (see Semantic Versioning (Section 4.2))

"versions": An array of extension version objects with the following child members:

"version:" The Extension Version Identifier, defined in Section 3.1. An example is the "versioning-0.4" extension version identifier.

"default:" OPTIONAL boolean value indicating which of the extension version objects is the default returned by the server when the client doesn't specify the version using the "versioning" query parameter, defined in Section 3.2. When there is a single extension version object, then the "default" member value is true by default. When there is more than one extension version object, the server MUST include only one extension version object with the "default" member set to true, and the remaining extension version objects "default" member value is false by default.

"start:" OPTIONAL date and time that the extension version will

be supported. When the member is not included, the extension version object is supported. Once the date and time has passed, the "start" member MUST be removed. The syntax for the date and time is defined in [RFC3339].

"end:" OPTIONAL date and time that the extension version will stop being supported. When the member is not included, the extension version object has no planned support expiry. Once the date and time has passed, the extension version object MUST be removed and the extension object MUST be removed if the last extension version object is removed. The syntax for the date and time is defined in [RFC3339].

"links:" OPTIONAL "links" array to extension version documentation. The relationship of these links is defined by the IANA registry described in [RFC8288]. The JSON name/values of "rel", "href", "hreflang", "title", "media", and "type" correspond to values found in Section 3 of [RFC8288]. The "value", "rel", and "href" JSON values MUST be specified. All other JSON values are OPTIONAL.

Example "links" member for the "ext2-0.1" extension version identifier:

```
"links": [
  {
    "value": "https://ext2.example/ext2-01.txt",
    "rel": "describedby",
    "href": "https://ext2.example/ext2-01.txt",
    "type": "text/plain"
  }
]
```

Figure 5: Example of "links" Member

Example response to /help query with "versioning_help" and "versioning_data" members and no client specified versioning in the /help query "https://example.com/rdap/help":

```
{
  "rdapConformance": [
    "rdap_level_0",
    "versioning",
    "opaque_ext1",
    "opaque_ext2",
    "semantic_ext1",
    "semantic_ext2",
    "semantic_ext3"
  ]
}
```

```
],
"versioning_help": [
  {
    "extension": "rdap_level_0",
    "type": "opaque",
    "versions": [
      {
        "version": "rdap_level_0"
      }
    ]
  },
  {
    "extension": "versioning",
    "type": "semantic",
    "versions": [
      {
        "version": "versioning-0.3"
      },
      {
        "version": "versioning-0.4",
        "default": true
      }
    ]
  }
],
{
  "extension": "opaque_ext1",
  "type": "opaque",
  "versions": [
    {
      "version": "opaque_ext1"
    }
  ]
},
"end": "2024-12-31T23:59:59Z"
},
{
  "extension": "opaque_ext2",
  "type": "opaque",
  "versions": [
    {
      "version": "opaque_ext2"
    }
  ]
},
"start": "2024-12-31T23:59:59Z",
"links": [
  {
    "value": "https://ext2.example/doc/html/opaque_ext2.txt",
    "rel": "describedby",
    "href": "https://ext2.example/doc/html/opaque_ext2.txt",
```

```
        "type": "text/plain"
      }
    ]
  },
  {
    "extension": "semantic_ext1",
    "type": "semantic",
    "versions": [
      {
        "version": "semantic_ext1-0.1",
        "end": "2024-12-31T23:59:59Z"
      },
      {
        "version": "semantic_ext1-1.0",
        "default": true
      },
      {
        "version": "semantic_ext1-1.1",
        "start": "2024-12-31T23:59:59Z"
      }
    ]
  }
],
{
  "extension": "semantic_ext2",
  "type": "semantic",
  "versions": [
    {
      "version": "semantic_ext2-0.1",
      "end": "2024-12-31T23:59:59Z",
      "links": [
        {
          "value": "https://ext2.example/doc/html/semantic_ext2-01.txt",
          "rel": "describedby",
          "href": "https://ext2.example/doc/html/semantic_ext2-01.txt",
          "type": "text/plain"
        }
      ]
    }
  ]
}
],
{
  "extension": "semantic_ext3",
  "type": "semantic",
  "versions": [
    {
      "version": "semantic_ext3-1.0",
      "start": "2024-12-31T23:59:59Z"
    }
  ]
}
```

```

    ]
  }
],
"versioning_data": [
  {
    "extension": "rdap_level_0",
    "type": "opaque",
    "version": "rdap_level_0"
  },
  {
    "extension": "versioning",
    "type": "semantic",
    "version": "versioning-0.4"
  }
]
}

```

Figure 6: Response to /help Query with No Client Specified Versioning

Example response to /help query with the "versioning_help" and "versioning" members and client specified "versioning-0.3" extension version in the /help query "https://example.com/rdap/help?versioning=versioning-0.3". The "versioning-0.3" extension version uses the "versioning" member and not the "versioning_data" member.

```

{
  "rdapConformance": [
    "rdap_level_0",
    "versioning",
    "opaque_ext1",
    "opaque_ext2",
    "semantic_ext1",
    "semantic_ext2",
    "semantic_ext3"
  ],
  "versioning_help": [
    {
      "extension": "rdap_level_0",
      "type": "opaque",
      "versions": [
        {
          "version": "rdap_level_0"
        }
      ]
    },
    {
      "extension": "versioning",

```

```
"type": "semantic",
"versions": [
  {
    "version": "versioning-0.3"
  },
  {
    "version": "versioning-0.4",
    "default": true
  }
]
},
{
  "extension": "opaque_ext1",
  "type": "opaque",
  "versions": [
    {
      "version": "opaque_ext1"
    }
  ],
  "end": "2024-12-31T23:59:59Z"
},
{
  "extension": "opaque_ext2",
  "type": "opaque",
  "versions": [
    {
      "version": "opaque_ext2"
    }
  ],
  "start": "2024-12-31T23:59:59Z",
  "links": [
    {
      "value": "https://ext2.example/doc/html/opaque_ext2.txt",
      "rel": "describedby",
      "href": "https://ext2.example/doc/html/opaque_ext2.txt",
      "type": "text/plain"
    }
  ]
},
{
  "extension": "semantic_ext1",
  "type": "semantic",
  "versions": [
    {
      "version": "semantic_ext1-0.1",
      "end": "2024-12-31T23:59:59Z"
    }
  ],
  {
```

```
        "version": "semantic_ext1-1.0",
        "default": true
      },
      {
        "version": "semantic_ext1-1.1",
        "start": "2024-12-31T23:59:59Z"
      }
    ]
  },
  {
    "extension": "semantic_ext2",
    "type": "semantic",
    "versions": [
      {
        "version": "semantic_ext2-0.1",
        "end": "2024-12-31T23:59:59Z",
        "links": [
          {
            "value": "https://ext2.example/doc/html/semantic_ext2-01.txt",
            "rel": "describedby",
            "href": "https://ext2.example/doc/html/semantic_ext2-01.txt",
            "type": "text/plain"
          }
        ]
      }
    ]
  }
],
{
  "extension": "semantic_ext3",
  "type": "semantic",
  "versions": [
    {
      "version": "semantic_ext3-1.0",
      "start": "2024-12-31T23:59:59Z"
    }
  ]
}
],
"versioning": [
  {
    "extension": "rdap_level_0",
    "type": "opaque",
    "version": "rdap_level_0"
  },
  {
    "extension": "versioning",
    "type": "semantic",
    "version": "versioning-0.3"
  }
]
```

```

    }
  ]
}
```

Figure 7: Response to /help Query with "versioning-0.3" Extension Version

3.3.3. "versioning_data" Member

The "versioning_data" member MUST be added to the RDAP response when there is one or more RDAP extension fields. The "versioning_data" member is included as a member of the object class in a lookup response, such as the object classes defined in [RFC9083], and as a member of the object instances in a search response, such as the object instances defined in [RFC9083], and as a member of the response to a /help query, such as shown in Response to /help Query with No Client Specified Versioning (Figure 6). The "versioning" member contains an array of versioning objects with the following child members:

"extension": REQUIRED Extension Identifier, defined in [RFC7480], and returned in the "rdapConformance" data structure of [RFC9083]. An example is the "versioning" extension identifier.

"type": REQUIRED Versioning Type used by the extension. The Versioning Types are registered using the "versioning" type in the JSON Values Registry. There are two predefined Versioning Types, including "opaque" (see Opaque Versioning (Section 4.1)) and "semantic" (see Semantic Versioning (Section 4.2))

"version": REQUIRED Extension Version Identifier, defined in Section 3.1, that is included in the response. An example is the "versioning-0.4" extension version identifier.

Example domain lookup response with "versioning_data" member and no client specified versioning in the lookup query

"https://example.com/rdap/domain/versioning.example":

```

{
  "rdapConformance": [
    "rdap_level_0",
    "versioning",
    "semantic_ext1",
    "opaque_ext2"
  ],
  "objectClassName": "domain",
  "handle": "XXXX",
  "ldhName": "versioning.example",
```

```
"links": [
  {
    "value": "https://example.net/domain/versioning.example",
    "rel": "self",
    "href": "https://example.net/domain/versioning.example",
    "type": "application/rdap+json"
  }
],
"status": [
  "ok"
],
"events": [
  {
    "eventAction": "registration",
    "eventDate": "1990-12-31T23:59:59Z"
  },
  {
    "eventAction": "expiration",
    "eventDate": "2025-12-31T23:59:59Z"
  }
],
"semantic_ext1": {
  "value": "example 1",
  "newoptionalstring": "new value"
},
"opaque_ext2": {
  "name": "example 2"
},
"versioning_data": [
  {
    "extension": "rdap_level_0",
    "type": "opaque",
    "version": "rdap_level_0"
  },
  {
    "extension": "versioning",
    "type": "semantic",
    "version": "versioning-0.4"
  },
  {
    "extension": "semantic_ext1",
    "type": "semantic",
    "version": "semantic_ext1-1.0"
  },
  {
    "extension": "opaque_ext2",
    "type": "opaque",
    "version": "opaque_ext2"
  }
]
```



```
}  
]  
}
```

Figure 8: RDAP Domain Lookup Response with No Client Specified
Versioning

Example domain lookup response with "versioning_data" member and client specified "semantic_ext1-0.1" extension version in the lookup query "https://example.com/rdap/domain/versioning.example?versioning=semantic_ext1-0.1":

```
{  
  "rdapConformance": [  
    "rdap_level_0",  
    "versioning",  
    "semantic_ext1",  
    "opaque_ext2"  
  ],  
  "objectClassName": "domain",  
  "handle": "XXXX",  
  "ldhName": "versioning.example",  
  "links": [  
    {  
      "value": "https://example.net/domain/versioning.example",  
      "rel": "self",  
      "href": "https://example.net/domain/versioning.example",  
      "type": "application/rdap+json"  
    }  
  ],  
  "status": [  
    "ok"  
  ],  
  "events": [  
    {  
      "eventAction": "registration",  
      "eventDate": "1990-12-31T23:59:59Z"  
    },  
    {  
      "eventAction": "expiration",  
      "eventDate": "2025-12-31T23:59:59Z"  
    }  
  ],  
  "semantic_ext1": {  
    "value": "example 1"  
  },  
  "opaque_ext2": {  
    "name": "example 2"  
  }  
}
```

```

},
"versioning_data": [
  {
    "extension": "rdap_level_0",
    "type": "opaque",
    "version": "rdap_level_0"
  },
  {
    "extension": "versioning",
    "type": "semantic",
    "version": "versioning-0.4"
  },
  {
    "extension": "semantic_ext1",
    "type": "semantic",
    "version": "semantic_ext1-0.1"
  },
  {
    "extension": "opaque_ext2",
    "type": "opaque",
    "version": "opaque_ext2"
  }
]
}

```

Figure 9: RDAP Domain Lookup Response with "semantic_ext1-0.1" Extension Version

Example domain lookup response with "versioning_data" member and client specified "semantic_ext1-0.1" and "opaque_ext2" extension versions in the lookup query "https://example.com/rdap/domain/versioning.example?versioning=semantic_ext1-0.1,opaque_ext2":

```

{
  "rdapConformance": [
    "rdap_level_0",
    "versioning",
    "semantic_ext1",
    "opaque_ext2"
  ],
  "objectClassName": "domain",
  "handle": "XXXX",
  "ldhName": "versioning.example",
  "links": [
    {
      "value": "https://example.net/domain/versioning.example",
      "rel": "self",
      "href": "https://example.net/domain/versioning.example",
    }
  ]
}

```

```
    "type": "application/rdap+json"
  },
  "status": [
    "ok"
  ],
  "events": [
    {
      "eventAction": "registration",
      "eventDate": "1990-12-31T23:59:59Z"
    },
    {
      "eventAction": "expiration",
      "eventDate": "2025-12-31T23:59:59Z"
    }
  ],
  "semantic_ext1": {
    "value": "example 1"
  },
  "opaque_ext2": {
    "name": "example 2"
  },
  "versioning_data": [
    {
      "extension": "rdap_level_0",
      "type": "opaque",
      "version": "rdap_level_0"
    },
    {
      "extension": "versioning",
      "type": "semantic",
      "version": "versioning-0.4"
    },
    {
      "extension": "semantic_ext1",
      "type": "semantic",
      "version": "semantic_ext1-0.1"
    },
    {
      "extension": "opaque_ext2",
      "type": "opaque",
      "version": "opaque_ext2"
    }
  ]
}
```

Figure 10: RDAP Domain Lookup Response with "semantic_ext1-0.1" and "opaque_ext2" Extension Versions

4. Versioning Types

Versioning Types are extensible by formally defining the Extension Version Identifier, how the Versioning Type is represented in the "versioning_help" member (Section 3.3.2), and how the Versioning Type is represented in the "versioning" member (Section 3.3.2). The definition of the extension version identifier determines how the Versioning Type is represented in the Extension Versioning Request (Section 3.2). Each Versioning Type needs to have a unique "versioning" RDAP JSON Values Registry Type field value. This document pre-defines two Versioning Types with Opaque Versioning (Section 4.1) and Semantic Versioning (Section 4.2). Other Versioning Types can be defined and registered in the RDAP JSON Values Registry.

4.1. Opaque Versioning

Opaque Versioning is defined in Section 8 of [I-D.ietf-regext-rdap-extensions], to support RDAP extensions that are identified using the Extension Identifier included in the RDAP Conformance and is the base Versioning Type that can be applied to any RDAP extension. All RDAP extensions are registered in the RDAP Extensions Registry using the Extension Identifier in [RFC7480], which is the basis for other Versioning Types. An RDAP Extension that supports another Versioning Type, such as the Semantic Versioning (Section 4.2), MAY be referred to using Opaque Versioning in the Extension Versioning Request (Section 3.2) and the server returns the extension version with the "default" (Section 3.3.2) member set to true.

Example reference to Opaque Versioning of the "versioning" extension:

```
https://example.com/rdap/domain/  
versioning.example?versioning=versioning
```

Reference to Opaque Versioning of the "versioning" extension.

The string literal "rdap_level_0" included in the RDAP Conformance data structure, defined in Section 4.1 of [RFC9083], is a special form of Opaque Versioning that signifies the version of the RDAP protocol itself. The "rdap_level_0" literal is treated as an Opaque Extension Version Identifier (Section 4.1.1) and included in both the "versioning_help" member (Section 3.3.2) and the "versioning_data" member (Section 3.3.3). Future versions of the RDAP protocol, such as "rdap_level_1", can be defined in the future and is treated as a string literal in a similar fashion as "rdap_level_0", but MAY use a different Versioning Type.

4.1.1. Opaque Extension Version Identifier

The Opaque Extension Version Identifier directly matches the Extension Identifier registered in the RDAP Extensions Registry. The ABNF for the Extension Identifier is defined in "Figure 1: ABNF for JSON Names" of [RFC7480].

Example Opaque Extension Versioning Identifiers:

"versioning"

Opaque versioning of the "versioning" extension.

"opaque_ext1"

Opaque versioning of the "opaque_ext1" extension.

4.1.2. Opaque Versioning "versioning_help" Member

For an RDAP extension that only supports Opaque Extension Versioning, there is only a single extension version. All of the members of the "versioning_help" member (Section 3.3.2) are supported with Opaque Versioning, but there will be only a single "versions" member object that has the "version" member matching the "extension" member value. Clients can leverage the Extension Version Identifier value in the Extension Versioning Request (Section 3.2) for Opaque Versioning.

Example response to /help query with "versioning_help" and "versioning_data" members with support for only Opaque Versioning in the /help query "https://example.com/rdap/help":

```
{
  "rdapConformance": [
    "rdap_level_0",
    "versioning",
    "opaque_ext1",
    "opaque_ext2",
  ],
  "versioning_help": [
    {
      "extension": "rdap_level_0",
      "type": "opaque",
      "versions": [
        {
          "version": "rdap_level_0"
        }
      ]
    },
    {
      "extension": "versioning",
```

```
    "type": "opaque",
    "versions": [
      {
        "version": "versioning"
      },
    ]
  },
  {
    "extension": "opaque_ext1",
    "type": "opaque",
    "versions": [
      {
        "version": "opaque_ext1"
      }
    ],
    "end": "2024-12-31T23:59:59Z"
  },
  {
    "extension": "opaque_ext2",
    "type": "opaque",
    "versions": [
      {
        "version": "opaque_ext2"
      }
    ],
    "start": "2024-12-31T23:59:59Z",
    "links": [
      {
        "value": "https://ext2.example/opaque_ext2.txt",
        "rel": "describedby",
        "href": "https://ext2.example/paque_ext2.txt",
        "type": "text/plain"
      }
    ]
  }
],
"versioning_data": [
  {
    "extension": "rdap_level_0",
    "type": "opaque",
    "version": "rdap_level_0"
  },
  {
    "extension": "versioning",
    "type": "opaque",
    "version": "versioning"
  }
]
```

```
}
```

Figure 11: Response to /help Query with only Opaque Versioning

4.1.3. Opaque Versioning "versioning_data" Member

All of the members of the "versioning_data" member (Section 3.3.3) are supported with Opaque Versioning, where the "extension" and "version" members are both set with the Extension Identifier and the "type" is set to "opaque". The Response to /help Query with only Opaque Versioning (Figure 11) includes an example of the "versioning" extension represented using Opaque Versioning.

4.2. Semantic Versioning

The Semantic Versioning uses a subset of the Semantic Versioning [SemVer] rules, where the patch version, pre-release version, and build metadata are not used. RDAP versioning is only associated with changes to the protocol interface that is the public API of [SemVer], so there is no concept of patching, pre-release, and build metadata information. Only the [SemVer] major version and minor version are used for Extension Versioning, and is based on the stability of the extension as opposed to backward compatibility. The RDAP extension given a version number MAJOR.MINOR, increment the:

1. MAJOR version when the extension interface is considered stable and complete
2. MINOR version includes an update to the extension interface that is not considered stable and complete

The following are the Extension Versioning rules based on modifications to the Semantic Versioning [SemVer] rules:

1. A normal version MUST take the form X.Y where X.Y are non-negative integers, and MUST NOT contain leading zeroes. X is the major version and Y is the minor version. Each element MUST increase numerically. For instance: 1.1 -> 1.2 -> 1.3.
2. Major version zero (0.y) is for initial definition. Anything MAY change at any time. The extension interface SHOULD NOT be considered stable. An extension that is an initial Internet Draft prior to Working Group Last Call (WGLC) SHOULD use a zero major version.

3. Version 1.0 defines the extension interface. The way in which the version number is incremented after this is dependent on this extension interface and how it changes. An extension that is an initial Internet Draft that has passed Working Group Last Call (WGLC) SHOULD use version 1.0.
4. Minor version Y ($x.Y \mid x > 0$) MUST be incremented if new interface changes are introduced in the extension interface from the last major version.
5. Major version X ($X.y \mid X > 0$) MUST be incremented once the new interface is considered stable and complete. The minor version MUST be reset to 0 when major version is incremented.
6. Precedence refers to how versions are compared to each other when ordered.
 1. Precedence MUST be calculated by separating the version into major and minor identifiers in that order.
 2. Precedence is determined by the first difference when comparing each of these identifiers from left to right as follows: Major and minor are always compared numerically. Example: 1.0 < 2.0 < 2.1.

4.2.1. Semantic Extension Version Identifier

The Semantic Versioning Identifier defines the versioning ABNF rule in Extension Version Identifier ABNF (Figure 1) to include the major and minor version of the extension that follows the modified Semantic Versioning [SemVer] rules.

```
semantic-version-identifier = identifier versioning
identifier = ALPHA *( ALPHA / DIGIT / "_" ) ; Extension Identifier
versioning = ; "-" major "." minor
major = numeric-identifier
minor = numeric-identifier
numeric-identifier = "0" / positive-digit *DIGIT
positive-digit = %x31-39 ; 1-9
```

Figure 12: Semantic Versioning Identifier ABNF

Example Semantic Extension Versioning Identifiers:

"versioning-0.4"

The "versioning" extension version identifier that is not considered stable.

"semantic_ext1-1.0"

First stable version of the "semantic_ext1" extension.

"semantic_ext1-1.1"

Update to "semantic_ext1-1.0" of the "semantic_ext1" extension that is not considered stable.

"semantic_ext1-2.0"

New stable version of the "semantic_ext1" extension.

4.2.2. Semantic Versioning "versioning_help" Member

For an RDAP extension that supports Semantic Extension Versioning, there can be many Extension Versioning Identifiers associated with the Extension Identifier, so the "versions" member of the "versioning_help" member can include a list of more than one extension version object. All of the members of the "versioning_help" member (Section 3.3.2) are supported with Semantic Versioning.

Example response to /help query with "versioning_help" and "versioning_data" members with support for Semantic Versioning in the /help query "https://example.com/rdap/help":

```
{
  "rdapConformance": [
    "rdap_level_0",
    "versioning",
    "opaque_ext1",
    "opaque_ext2",
  ],
  "versioning_help": [
    {
      "extension": "rdap_level_0",
      "type": "opaque",
      "versions": [
        {
          "version": "rdap_level_0"
        }
      ]
    },
    {
      "extension": "versioning",
      "type": "semantic",
      "versions": [
        {
          "version": "versioning-0.3"
        }
      ],
    },
  ],
}
```

```
    {
      "version": "versioning-0.4",
      "default": true
    }
  ],
},
{
  "extension": "semantic_ext1",
  "type": "semantic",
  "versions": [
    {
      "version": "semantic_ext1-0.1",
      "end": "2024-12-31T23:59:59Z"
    },
    {
      "version": "semantic_ext1-1.0",
      "default": true
    },
    {
      "version": "semantic_ext1-1.1",
      "start": "2024-12-31T23:59:59Z"
    }
  ]
},
{
  "extension": "semantic_ext2",
  "type": "semantic",
  "versions": [
    {
      "version": "semantic_ext2-0.1",
      "end": "2024-12-31T23:59:59Z",
      "links": [
        {
          "value": "https://ext2.example/semantic_ext2-01.txt",
          "rel": "describedby",
          "href": "https://ext2.example/semantic_ext2-01.txt",
          "type": "text/plain"
        }
      ]
    }
  ]
},
{
  "extension": "semantic_ext3",
  "type": "semantic",
  "versions": [
    {
      "version": "semantic_ext3-1.0",
```

```
        "start": "2024-12-31T23:59:59Z"
      }
    ]
  },
  "versioning_data": [
    {
      "extension": "rdap_level_0",
      "type": "opaque",
      "version": "rdap_level_0"
    },
    {
      "extension": "versioning",
      "type": "opaque",
      "version": "versioning-0.4"
    }
  ]
}
```

Figure 13: Response to /help Query with only Semantic Versioning

4.2.3. Semantic Versioning "versioning_data" Member

All of the members of the "versioning_data" member (Section 3.3.3) are supported with Semantic Versioning, where the "extension" member is set with the Extension Identifier, the "version" member of the extension included in the response is set with Semantic Extension Version Identifier (Section 4.2.1), and the "type" is set to "semantic". The Response to /help Query with only Semantic Versioning (Figure 13) includes an example of the "versioning" extension represented using Semantic Versioning with "versioning-0.4".

5. Versioning Considerations

This section covers considerations for extensions that support the versioning extension and for defining new Versioning Types.

5.1. Versioning Extension Considerations

All extensions support Opaque Versioning (Section 4.1) by default and MAY support other forms for Versioning Types, such as the Semantic Versioning (Section 4.2). The following are considerations for the servers that support the versioning extension:

1. An extension document SHOULD define an "Extension Version Identifier" section that includes the Versioning Types supported by the extension with the associated extension version identifier.
2. A server MUST ignore Extension Version Identifiers that are unknown in the Extension Versioning Request.
3. A server SHOULD first process the Extension Versioning Request (Section 3.2) prior to processing the query parameters of other extensions to implement the client's extension versioning hint.
4. A server MAY define the policy of the extension versions that are supported and the timing of the extension versions, such as how many overlapping extension versions to support in parallel.

5.2. Versioning Type Considerations

The following are considerations for the definition of new Versioning Types in a Versioning Type Specification. The Opaque Versioning (Section 4.1) and Semantic Versioning (Section 4.2) with the associated RDAP JSON Values Registry (Section 7.2) registrations are concrete examples of a Versioning Type Specification. The Versioning Type Specification MUST include the following:

1. Definition of a unique RDAP JSON Values Registry "versioning" value.
2. Definition of the format of the Extension Versioning identifier.
3. Definition of the members of the "versioning_help" member that apply.
4. Definition of the members of the "versioning_data" member that apply.

6. Extension Version Identifier

This extension supports the following Versioning Types:

Opaque Versioning: The Opaque Extension Versioning Identifier is "versioning".

Semantic Versioning: The Semantic Extension Versioning Identifier is "versioning-0.4".

7. IANA Considerations

7.1. RDAP Extensions Registry

IANA is requested to register the following value in the RDAP Extensions Registry:

Extension identifier: versioning

Registry operator: Any

Published specification: This document.

Contact: IETF <iesg@ietf.org>

Intended usage: This extension identifies RDAP extension versioning.

7.2. RDAP JSON Values Registry

Section 10.2 of [RFC9083] defines the RDAP JSON Values Registry with pre-defined Type field values and the use of the "Expert Review" policy defined in [RFC8126]. This specification defines a new "versioning" RDAP JSON Values Registry Type field value that can be used to register pre-defined Versioning Types values. The registered "versioning_data" value is referenced using the "type" field of the "versioning" field (Section 3.3.3) or "versioning_help" field object (Section 3.3.2). IANA is instructed to update the RDAP JSON Values Registry to accept the additional "versioning" type field values.

The following values should be registered by the IANA in the RDAP JSON Values Registry described in [RFC9083]:

Value: opaque

Type: versioning

Description: Opaque Extension Versioning Type, where the rdapConformance value directly indicates the extension version.

Registrant Name: IETF

Registrant Contact Information: iesg@ietf.org

Value: semantic

Type: versioning

Description: Semantic Extension Versioning Type, where the rdapConformance value is appended with a semantic version number.

Registrant Name: IETF

Registrant Contact Information: `iesg@ietf.org`

8. Security Considerations

The RDAP extensions available to clients can be subject to server disclosure and authorization policies. Inclusion of available RDAP extensions and their available versions in the "versioning_help" member, defined in Section 3.3.2, of the RDAP help response and inclusion of the RDAP extensions in the RDAP response with the "versioning_data" member, defined in Section 3.3.3, can be dependent on authentication and authorization of the client by the server.

9. Implementation Status

NOTE: Please remove this section and the reference to RFC 7942 prior to publication as an RFC.

9.1. IIT-CNR/Registro.it RDAP Server

Responsible Organization: Institute of Informatics and Telematics of National Research Council (IIT-CNR)/Registro.it

Location: `https://rdap.pubtest.nic.it/`

Description: This implementation includes support for RDAP queries using data from the public test environment of .it ccTLD.

Level of Maturity: This is an "alpha" test implementation.

Coverage: This implementation includes all of the features described in this specification.

Contact Information: Mario Loffredo, `mario.loffredo@iit.cnr.it`

10. Acknowledgements

The authors wish to thank the following persons for their feedback and suggestions: Scott Hollenbeck, Andy Newton, and Jasdip Singh.

11. References

11.1. Normative References

- [I-D.ietf-regext-rdap-extensions]
Newton, A., Singh, J., and T. Harrison, "RDAP Extensions",
Work in Progress, Internet-Draft, draft-ietf-regext-rdap-
extensions-06, 29 April 2025,
<<https://datatracker.ietf.org/doc/html/draft-ietf-regext-rdap-extensions-06>>.
- [I-D.ietf-regext-rdap-x-media-type]
Newton, A. and J. Singh, "Extensions Parameter for the
RDAP Media Type", Work in Progress, Internet-Draft, draft-
ietf-regext-rdap-x-media-type-03, 9 April 2025,
<<https://datatracker.ietf.org/doc/html/draft-ietf-regext-rdap-x-media-type-03>>.
- [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>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,
<<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax
Specifications: ABNF", STD 68, RFC 5234,
DOI 10.17487/RFC5234, January 2008,
<<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC7480] Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the
Registration Data Access Protocol (RDAP)", STD 95,
RFC 7480, DOI 10.17487/RFC7480, March 2015,
<<https://www.rfc-editor.org/info/rfc7480>>.
- [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>>.
- [RFC8288] Nottingham, M., "Web Linking", RFC 8288,
DOI 10.17487/RFC8288, October 2017,
<<https://www.rfc-editor.org/info/rfc8288>>.

- [RFC9082] Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <<https://www.rfc-editor.org/info/rfc9082>>.
- [RFC9083] Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <<https://www.rfc-editor.org/info/rfc9083>>.

11.2. Informative References

- [SemVer] "Semantic Versioning 2.0.0 (text from June 19, 2020)", <<https://github.com/semver/semver/blob/8b2e8eec394948632957639dfa99fc7ec6286911/semver.md>>.

Appendix A. Change History

A.1. Change from 00 to 01

1. Updated the Abstract to reference the use of machine-parseable identifiers.
2. Updated the Introduction to reference the informal support for versioning in RFC9083.
3. Removed rule "An extension using Extension Versioning MUST declare the extension in a specification." from Section 3.1 "Extension Versioning".

A.2. Change from 01 to 02

1. Restructure the draft to support the features of providing versioning meta-data in the help response, providing extension versioning information in the query response, and provide the ability for the client to specify the desired set of extension versions in the query with an extensible set of versioning types. The following changes apply:
 1. Added the "Versioning RDAP Features" section to define the base set of features of the extension that applies to an extensible set of versioning types.
 2. Updated the Extension Version Identifier to be more generic to support an extensible set of extension version identifiers by the set of versioning types.
 3. Created the "Extension Versioning Request" section to support client extension versioning hints using either the "versioning" query parameter or the versioning extensions media type parameter in [I-D.ietf-regext-rdap-x-media-type].

4. Updated the "versioning_help" Member" section to include the versioning type for each extension and to reference the guidance in Section 7 of [I-D.ietf-regext-rdap-extensions] and extending it by the versioning extension.
5. Updated the "versioning" Member" section to include the versioning type for returned RDAP response extensions.
6. Created the "Versioning Types" section to define the two pre-defined versioning types of Opaque Versioning and Semantic Versioning following a common template.
7. Created the "Versioning Extension Considerations" section to provide guidance on the use of the versioning extension by RDAP extensions and servers, and to provide guidance on the definition of new versioning types.
8. In the "IANA Considerations" section, add a new RDAP JSON Values Type of "versioning" to support defining multiple versioning types that can be used by RDAP extensions.
9. In the "IANA Considerations" section, Pre-register the RDAP JSON Values "versioning" types defined in the draft, including "opaque" and "semantic".

A.3. Change from 02 to REGEXT 00

1. Changed to regext working group draft by changing draft-gould-regext-rdap-versioning to draft-ietf-regext-rdap-versioning.

A.4. Change from REGEXT 00 to REGEXT 01

1. Replaced I-D.newton-regext-rdap-x-media-type with I-D.ietf-regext-rdap-x-media-type and I-D.newton-regext-rdap-extensions with I-D.ietf-regext-rdap-extensions.
2. Changed 'defined息' to 'defined' in the Introduction.

A.5. Change from REGEXT 01 to REGEXT 02

1. Changed "versioning-help" to "versioning_help" to comply with RFC 7480 and subsequently changed the version of the versioning extension to 'versioning-0.3'.
2. Changed "Opaque Versioning Identifier" to "Opaque Extension Version Identifier" to be consistent with the term "Semantic Extension Version Identifier".
3. Inclusion of "rdap_level_0" as an extension in the "versioning_help" and "versioning" members.
4. In "Versioning RDAP Features" section, fixed "what extension versions is supported" to "what extension versions are supported".
5. Add clarification to the last sentence of the "Introduction" section with the specification of a versioning type.
6. Capitalized "Versioning Type" throughout the draft.

7. Changed "extensions" Versioning Extensions Type Parameter to the shorter form "extensions" parameter in the "Versioning Extensions Media Type Parameter" section.
8. Made the "Versioning RDAP Response" section a sibling instead of a child of the "Extension Versioning Request" section. Also made the "'versioning-help' Member" section and the "'versioning' Member" section children of the "Versioning RDAP Response" section.

A.6. Change from REGEXT 02 to REGEXT 03

1. Changed "versioning" to "versioning_data" and subsequently changed the version of the versioning extension to 'versioning-0.4'.
2. Added normative language to the "Extension Versioning Request" section about server processing of a client request for an unsupported extension version.
3. Added an example to "Opaque Versioning" section about the reference to Opaque versioning of an extension supporting Semantic versioning.
4. Added the "Implementation Status" section.
5. Made examples consistent with the use of "rdap_level_0" extension.
6. Fixed typos.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: jgould@verisign.com
URI: <http://www.verisigninc.com>

Daniel Keathley
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: dkeathley@verisign.com
URI: <http://www.verisigninc.com>

Mario Loffredo
IIT-CNR/Registro.it
Via Moruzzi,1
56124 Pisa
Italy
Email: mario.loffredo@iit.cnr.it
URI: <http://www.iit.cnr.it>