

Registration Protocols Extensions (regext)
Internet-Draft
Intended status: Standards Track
Expires: 16 July 2026

A. Newton
ICANN
12 January 2026

RDAP Simple Redaction
draft-newton-regext-rdap-simple-redaction-01

Abstract

This document defines a simple redaction extension for the Registration Data Access Protocol (RDAP).

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 16 July 2026.

Copyright Notice

Copyright (c) 2026 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. Background	2
2. Redaction Keys	4
2.1. Redaction of Data in Strings	4
2.1.1. Unstructured Text	4
2.1.2. Telephone Numbers	4
2.1.3. Email Addresses	5
2.1.4. URIs with Host Names	5
2.1.5. Dates and Times	5
2.2. Redaction of Other JSON Object Members	5
3. Specifying Keys	6
3.1. Keying Strategies	9
3.1.1. Unclean Data	9
3.1.2. Handles	10
4. Examples	10
4.1. Unstructured Addresses	10
4.2. Structured Addresses	11
4.3. A Complete Example	12
5. Security Considerations	19
6. IANA Considerations	19
7. References	19
7.1. Normative References	19
7.2. Informative References	20
Author's Address	20

1. Background

This document defines a simple redaction extension for redacting information from Registration Data Access Protocol (RDAP) responses. This extension offers a much simpler approach to the redaction of information than RFC 9537. Additionally, it has several advantages over RFC 9537:

- * The reasons for a redaction may be specified in multiple languages.
- * All redactions may be processed by a client programmatically.
- * String data may be partially redacted.
- * It re-uses remarks and notices from [RFC9083] and is backward-compatible with clients that do not support this extension.
- * Re-use of remarks and notices enables linking of redacted data to redaction policies.
- * In most cases, notifications of redacted data is passed to the user with clients that do not support this extension.
- * Clients are not required to evaluate complicated and error-prone JSONPath expressions.

- * Usage in RDAP object containers, such as RDAP search results or bulk Whois-like documents, does not require rewriting of JSONPath expressions.

The following is an example of an RDAP response using simple redaction:

```
{
  "rdapConformance" : [ "Rdap_level_0", "simpleRedaction" ],
  "objectClassName" : "entity",
  "handle": "////REDACTED_HANDLE////",
  "vcardArray": [
    "vcard",
    [
      [ "version", {}, "text", "4.0" ],
      [ "fn", {}, "text", "////REDACTED_FULL_NAME////" ],
      [ "email",
        { "type": "work" },
        "text", "redacted_email@redacted.invalid"
      ]
    ]
  ],
  "remarks": [
    {
      "description": [
        "These values have been redacted according to policy."
      ],
      "simpleRedaction_keys" : {
        "keys" : [
          "////REDACTED_HANDLE////" ,
          "////REDACTED_FULL_NAME////" ,
          "redacted_email@redacted.invalid"
        ]
      }
    ]
  ]
}
```

The methods for noting redactions in this extension cannot redact numbers or booleans in JSON arrays. However, there are no known RDAP data structures either in [RFC9083] or any RDAP extension that use arrays containing numbers or booleans.

Additionally, there are no known redaction policies that require anything more than redaction of string data.

2. Redaction Keys

Simple redaction allows a server to define a set of keys, each used to signify when data has been redacted. Clients use these keys to notify a user of redacted information.

2.1. Redaction of Data in Strings

Data found in strings is the most common data type to be redacted. As of this writing, there are no known redaction policies that require the redaction of non-string data.

2.1.1. Unstructured Text

Keys signifying redaction for unstructured text, i.e. free form text, take the form of `////REDACTION_KEY////`. These keys begin with four forward-slash characters (`////`), followed by one or more of the upper and lower case characters A through Z, 0 through 9, hyphen (`-`), or under-bar (`_`), followed by four more forward-slash characters.

The following example demonstrates redaction of the full name value from a jCard ([RFC7095]) array:

```
["fn", {}, "text", "////REDACTED_FULL_NAME////"]
```

These keys may be placed in a string with other characters thus allowing for the partial redaction of a string:

```
"Alice ////LAST_NAME_REDACTION////"
```

2.1.2. Telephone Numbers

Keys for telephone numbers may either be represented as text or "tel" URIs. Both jCard and JSContact allow telephone numbers to be represented in either format.

2.1.2.1. Telephone Numbers as Text

Keys for telephone numbers represented as text use the same format as unstructured text (see Section 2.1.1).

The following is an example for jCard ([RFC7095]).

```
["tel", {}, "text", "////TELEPHONE_REDACTION////"]
```

2.1.2.2. TEL URIs

Keys for use in "tel" URIs ([RFC3966]) use local numbers with the phone context of "redacted.example". These keys begin with four dash characters ("----"), followed by one or more HEXDIGITs as defined by [RFC3966], followed by four more dash characters.

The following is an example of a "tel" URI used in a jCard array:

```
[ "tel", {}, "uri", "tel:----0000000----;extension=----999999----;phone-context=redacted.invalid" ]
```

When possible, it is RECOMMENDED to use a text representation as keys for "tel" URIs are more complex.

2.1.3. Email Addresses

Keys for email addresses MUST use a host part that is "redacted.invalid" but may use any local part allowable in an email address. For example: redacted_email@redacted.invalid.

The ".invalid" TLD is a special-use domain defined in [RFC6761] and is unusable on the Internet.

2.1.4. URIs with Host Names

Keys used in a URI with host names, such as an HTTP URI, MUST use a host name that is "redacted.invalid". For example: https://redacted.invalid/redacted_web_page.

The ".invalid" TLD is a special-use domain defined in [RFC6761] and is unusable on the Internet.

2.1.5. Dates and Times

Date and times in RDAP use [RFC3339]. Keys for these MUST use the year 0000. For example: 0000-00-00T23:20:50.52Z.

2.2. Redaction of Other JSON Object Members

Signaling of the redaction of other JSON data type is a JSON object use the insertion of the "simpleRedaction_data" member into the object where the data has been redacted by removal. For example, consider an RDAP autnum object:

```
{
  "objecClassName": "autnum",
  "handle": "AS-RANGE-FOO",
  "startNum":
  "startAutnum" : 65536,
  "endAutnum" : 65541
}
```

Should policy dictate the redaction of "endAutnum", this could be signaled in the following way:

```
{
  "objecClassName": "autnum",
  "handle": "AS-RANGE-FOO",
  "startNum":
  "startAutnum" : 65536,
  "simpleRedaction_data" : [
    { "key": "////NO_END_NUMBERS////", "members": ["endAutnum"]}
  ]
  "remarks": [
    {
      "description": [
        "No sane redaction policy would do this."
      ],
      "simpleRedaction_keys" : {
        "keys" : [
          "////NO_END_NUMBERS////"
        ]
      }
    }
  ]
}
```

The "simpleRedaction_data" member is an array of objects. Each object contains a "key" string signifying the redaction key and a "members" string array signifying the JSON members associated with the redaction key.

3. Specifying Keys

All redaction keys (Section 2) are explicitly specified by the server, inside a structure called "simpleRedaction_keys" which is inserted in either an RDAP remark or notice. Keys MUST be specified at least once but MAY be specified more than once.

Each "simpleRedaction_keys" object MUST have a member named "keys" which is an array of strings containing the redaction keys Section 2.

The following is an example:

```
"remarks": [
  {
    "description": [
      "These values have been redacted according to policy."
    ],
    "simpleRedaction_keys" : {
      "keys" : [
        "////REDACTED_HANDLE////" ,
        "////REDACTED_FULL_NAME////" ,
        "redacted_email@redacted.invalid"
      ]
    }
  }
]
```

A redaction key MAY appear in more than one remark or notice to allow a server to describe the cause of the redaction in more than one language.

The following is an example:

```
"remarks": [
  {
    "lang": "ja",
    "description": [
      "この拡張機能は、複数の言語での編集理由をサポートします。"
    ],
    "simpleRedaction_keys": {
      "keys": [
        "///REDACTED_FULL_NAME///"
      ]
    },
    "links": [
      {
        "value": "https://example.com/value",
        "rel": "about",
        "href": "https://example.com/ja/some-policy.html",
        "type": "text/html"
      }
    ]
  },
  {
    "lang": "en",
    "description": [
      "This extension supports redaction reasons in multiple languages."
    ],
    "simpleRedaction_keys": {
      "keys": [
        "///REDACTED_FULL_NAME///"
      ]
    },
    "links": [
      {
        "value": "https://example.com/value",
        "rel": "about",
        "href": "https://example.com/en/some-policy.html",
        "type": "text/html"
      }
    ]
  }
]
```

A client MUST NOT consider any key not found to be a "simpleRedaction_keys" structure as a valid redaction (i.e. do not signal to the user that the information has been redacted).

The "links" array found in remarks and notices, as described by [RFC9083] may be used to link to a web page describing the redaction policy. Clients may notify the user of the link, such as through hover-text or pop-ups. The "about" relationship MUST be used for this purpose. The following is an example.

```
"remarks": [
  {
    "description": [
      "This information has been redacted according to the linked policy.",
      "This extension supports associating redacted data links to redaction policies."
    ],
    "simpleRedaction_keys" : {
      "keys" : [
        "////REDACTED_HANDLE////" ,
        "////REDACTED_FULL_NAME////" ,
        "redacted_email@redacted.invalid"
      ]
    },
    "links": [
      {
        "value": "https://example.com/value",
        "rel": "about",
        "href": "https://example.com/some-policy.html",
        "type": "text/html"
      }
    ]
  }
]
```

This specification allows the use of "simpleRedaction_keys" in both RDAP remarks and notices. In RDAP, remarks are data pertaining to registered objects whereas notices are data pertaining to the response as a whole and the service being provided. Therefore, specifying redaction of data pertaining to registered objects should use RDAP remarks. And, unlikely as it would be, redaction of data pertaining service information should use RDAP notices.

3.1. Keying Strategies

3.1.1. Unclean Data

While it seems odd that some users would be allowed to give an email address with a host of "redacted.invalid" or a string that begins and ends with four forward-slashes to an Internet registration authority, some registries must deal with such data for various reasons.

One strategy servers may use would be to append a set of random characters to each key. For example, if a registered resource was given as "////I-fooled-you////" then the server could thwart this by appending the random characters "1234-NO-YOU-DID-NOT" to make the key "////I-fooled-you_1234-NO-YOU-DID-NOT////".

3.1.2. Handles

For servers operating under policies in which the "handle", as defined by [RFC9083] must be redacted, it would be beneficial to some clients to create unique redaction keys for each handle. While clients SHOULD use "self" links, as described in [RFC9083], to differentiate between objects returned in a response, in the absence of "self" links they often use the "handle". Therefore, servers SHOULD create a unique redaction key for each handle that is redacted.

4. Examples

4.1. Unstructured Addresses

[RFC7095] allows for the representation of unstructured postal addresses. The following is a simple example of an RDAP response with simple redactions where the postal address is given as unstructured.

```

{
  "rdapConformance" : [ "Rdap_level_0", "simpleRedaction" ],
  "objectClassName" : "entity",
  "handle":"foo",
  "vcardArray":[
    "vcard",
    [
      [ "version", {}, "text", "4.0"],
      [ "fn", {}, "text", "Bob"],
      [ "adr",
        {
          "type":"home",
          "label":"////REDACTED_STREET////\nVancouver\nBC\n////REDACTED_POSTAL_CODE////\n
"
        },
      ],
      [ "text",
        [
          "", "", "", "", "", "", "", ""
        ]
      ],
    ],
  ],
  "remarks": [
    {
      "description": [
        "These values have been redacted according to policy."
      ],
      "simpleRedaction_keys" : {
        "keys" : [
          "////REDACTED_STREET////" ,
          "////REDACTED_POSTAL_CODE////"
        ]
      }
    }
  ]
}

```

4.2. Structured Addresses

[RFC7095] allows for the representation of structured postal addresses. The following is a simple example of an RDAP response with simple redactions where the postal address is given as structured.

```

{
  "rdapConformance" : [ "Rdap_level_0", "simpleRedaction" ],
  "objectClassName" : "entity",
  "handle":"foo",
  "vcardArray":[
    "vcard",
    [
      [ "version", {}, "text", "4.0" ],
      [ "fn", {}, "text", "Bob" ],
      [ "adr",
        { "type":"work" },
        "text",
        [
          "",
          "////REDACTED_STREET////",
          "////REDACTED_STREET////",
          "Quebec",
          "QC",
          "////REDACTED_POSTAL_CODE////",
          "Canada"
        ]
      ]
    ]
  ],
  "remarks": [
    {
      "description": [
        "These values have been redacted according to policy."
      ],
      "simpleRedaction_keys" : {
        "keys" : [
          "////REDACTED_STREET////",
          "////REDACTED_POSTAL_CODE////"
        ]
      }
    ]
  ]
}

```

4.3. A Complete Example

The following is an example an RDAP response to a domain lookup in which the redactions specified in ICANN Registration Data Policy (<https://www.icann.org/resources/pages/registration-data-policy-2024-02-21-en>), have been applied.

```
{
  "rdapConformance": [
    "rdap_level_0", "simpleRedaction"
  ],
  "objectClassName": "domain",
  "handle": "////REGISTRY_DOMAIN_ID_REDACTION////",
  "ldhName": "example.com",
  "secureDNS": {
    "delegationSigned": false
  },
  "notices": [
    {
      "title": "Terms of Use",
      "description": [
        "Service subject to Terms of Use."
      ],
      "links": [
        {
          "rel": "self",
          "href": "https://www.example.com/terms-of-use",
          "type": "text/html",
          "value": "https://www.example.com/terms-of-use"
        }
      ]
    }
  ],
  "nameservers": [
    {
      "objectClassName": "nameserver",
      "ldhName": "ns1.example.com"
    },
    {
      "objectClassName": "nameserver",
      "ldhName": "ns2.example.com"
    }
  ],
  "entities": [
    {
      "objectClassName": "entity",
      "handle": "123",
      "roles": [
        "registrar"
      ],
      "publicIds": [
        {
          "type": "IANA Registrar ID",
          "identifier": "1"
        }
      ]
    }
  ]
}
```

```
],
"vcardArray": [
  "vcard",
  [
    [
      "version",
      {},
      "text",
      "4.0"
    ],
    [
      "fn",
      {},
      "text",
      "Example Registrar Inc."
    ],
    [
      "adr",
      {},
      "text",
      [
        "",
        "Suite 100",
        "123 Example Dr.",
        "Dulles",
        "VA",
        "20166-6503",
        "US"
      ]
    ],
    [
      "email",
      {},
      "text",
      "contact@organization.example"
    ],
    [
      "tel",
      {
        "type": "voice"
      },
      "uri",
      "tel:+1.7035555555;ext=1234"
    ],
    [
      "tel",
      {
        "type": "fax"
      }
    ]
  ]
]
```

```
    },
    "uri",
    "tel:+1.7035555556"
  ]
]
],
"entities": [
  {
    "objectClassName": "entity",
    "roles": [
      "abuse"
    ],
    "vcardArray": [
      "vcard",
      [
        [
          "version",
          {},
          "text",
          "4.0"
        ],
        [
          "fn",
          {},
          "text",
          "Abuse Contact"
        ],
        [
          "email",
          {},
          "text",
          "abuse@organization.example"
        ],
        [
          "tel",
          {
            "type": "voice"
          },
          "uri",
          "tel:+1.7035555555;ext=1234"
        ]
      ]
    ]
  }
]
},
{
  "objectClassName": "entity",
```

```
"handle": "////REGISTRY_REGISTRANT_ID_REDACTION////",
"roles": [
  "registrant"
],
"vcardArray": [
  "vcard",
  [
    [
      "version",
      {},
      "text",
      "4.0"
    ],
    [
      "fn",
      {},
      "text",
      "////REGISTRANT_NAME_REDACTION////"
    ],
    [
      "org",
      {},
      "text",
      "////REGISTRANT_ORG_REDACTION////"
    ],
    [
      "adr",
      {},
      "text",
      [
        "",
        "////REGISTRANT_STREET_REDACTION////",
        "////REGISTRANT_STREET_REDACTION////",
        "Quebec",
        "////REGISTRANT_CITY_REDACTION////",
        "////REGISTRANT_POSTAL_CODE_REDACTION////",
        "Canada"
      ]
    ],
    [
      "email",
      {},
      "text",
      "registrant-email-redaction@redacted.invalid"
    ],
    [
      "tel",
      {
```



```
        "type": "voice"
      },
      "text",
      "////0000000000////;ext=////1111111111////"
    ],
    [
      "tel",
      {
        "type": "fax"
      },
      "text",
      "////2222222222////"
    ]
  ]
},
{
  "objectClassName": "entity",
  "handle": "////REGISTRY_TECH_ID_REDACTION////",
  "roles": [
    "technical"
  ],
  "vcardArray": [
    "vcard",
    [
      [
        "version",
        {},
        "text",
        "4.0"
      ],
      [
        "fn",
        {},
        "text",
        "////TECH_NAME_REDACTION////"
      ],
      [
        "org",
        {},
        "text",
        "Example Inc."
      ],
      [
        "adr",
        {},
        "text",
        [
```

```
        "Suite 1234",
        "4321 Rue Somewhere",
        "Quebec",
        "QC",
        "G1V 2M2",
        "Canada"
    ],
    [
        "email",
        {},
        "text",
        "technical-email-redaction@redacted.invalid"
    ],
    [
        "tel",
        {
            "type": "voice"
        },
        "text",
        "////3333333333////;ext=321"
    ],
    [
        "tel",
        {
            "type": "fax"
        },
        "uri",
        "tel:+1-555-555-4321"
    ]
]
}
],
"events": [
    {
        "eventAction": "registration",
        "eventDate": "1997-06-03T00:00:00Z"
    },
    {
        "eventAction": "last changed",
        "eventDate": "2020-05-28T01:35:00Z"
    },
    {
        "eventAction": "expiration",
        "eventDate": "2021-06-03T04:00:00Z"
    }
]
```

```
],
"status": [
  "server delete prohibited",
  "server update prohibited",
  "server transfer prohibited",
  "client transfer prohibited"
],
"remarks": [
  {
    "description": [
      "These values have been redacted according to policy."
    ],
    "simpleRedaction_keys" : {
      "keys" : [
        "////REGISTRY_DOMAIN_ID_REDACTION////" ,
        "////REGISTRY_REGISTRANT_ID_REDACTION////" ,
        "////REGISTRANT_NAME_REDACTION////" ,
        "////REGISTRANT_STREET_REDACTION////" ,
        "////REGISTRANT_POSTAL_CODE_REDACTION////" ,
        "////0000000000////" ,
        "////1111111111////" ,
        "////REGISTRY_TECH_ID_REDACTION////" ,
        "////TECH_NAME_REDACTION////" ,
        "////1111111111////" ,
        "////2222222222////" ,
        "registrant-email-redaction@redacted.invalid" ,
        "tech-email-redaction@redacted.invalid" ,
        "////REGISTRANT_ORG_REDACTION////" ,
        "////REGISTRANT_CITY_REDACTION////"
      ]
    }
  }
]
```

5. Security Considerations

TBD.

6. IANA Considerations

TBD: registration of "simpleRedaction".

7. References

7.1. Normative References

- [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>>.
- [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, DOI 10.17487/RFC3966, December 2004, <<https://www.rfc-editor.org/info/rfc3966>>.
- [RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <<https://www.rfc-editor.org/info/rfc6761>>.
- [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>>.

7.2. Informative References

- [RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", RFC 7095, DOI 10.17487/RFC7095, January 2014, <<https://www.rfc-editor.org/info/rfc7095>>.

Author's Address

Andy Newton
ICANN
Email: andy@hxr.us