

Registration Protocols Extensions
Internet-Draft
Intended status: Standards Track
Expires: July 28, 2021

M. Loffredo
IIT-CNR/Registro.it
G. Brown
CentralNic Group plc
January 24, 2021

**Using JSContact in Registration Data Access Protocol (RDAP) JSON
Responses
draft-loffredo-regext-rdap-jcard-deprecation-04**

Abstract

This document describes an RDAP extension which represents entity contact information in JSON responses using JSContact.

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 July 28, 2021.

Copyright Notice

Copyright (c) 2021 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Rationale	3
1.2.	Conventions Used in This Document	3
2.	JSContact	3
3.	Using JSCard objects in RDAP Responses	4
3.1.	RDAP Query Parameters	6
4.	Transition Considerations	7
4.1.	RDAP Features Supporting a Transition Process	7
4.1.1.	Notices and Link Relationships	7
4.1.2.	rdapConformance Property	7
4.1.3.	Query Parameters	7
4.2.	Transition Procedure	7
4.2.1.	Transition Stages	8
4.2.1.1.	Stage 1: only jCard provided	8
4.2.1.2.	Stage 2: jCard sunset	8
4.2.1.3.	Stage 3: jCard deprecation	9
4.2.1.4.	Stage 4: jCard deprecated	11
4.2.1.5.	Length	11
4.2.1.6.	Goals	11
5.	Implementation Status	12
5.1.	IIT-CNR/Registro.it	12
6.	IANA Considerations	13
7.	Security Considerations	13
8.	References	13
8.1.	Normative References	13
8.2.	Informative References	14
Appendix A.	Change Log	15
A.1.	Change from 00 to 01	15
A.2.	Change from 01 to 02	15
A.3.	Change from 02 to 03	16
A.4.	Change from 03 to 04	16
	Authors' Addresses	16

[1.](#) Introduction

This document specifies an extension to the Registration Data Access Protocol (RDAP) that allows RDAP servers to use JSContact ([\[draft-ietf-jmap-jscontact\]](#)) to represent the contact information associated with entities in RDAP responses, instead of jCard ([\[RFC7095\]](#)). It also describes the process by which an RDAP server can transition from jCard to JSContact. RDAP query and response extensions are defined to facilitate the transition process.

1.1. Rationale

According to the feedback from RDAP Pilot Working Group ([[RDAP-PILOT-WG](#)], a group of RDAP server implementers representing registries and registrars of generic TLDs), the most commonly raised implementation concern, for both servers and client implementers, related to the use of jCard ([[RFC7095](#)]) to represent the contact information associated with entities. Working Group members reported jCard to be unintuitive, complicated to implement for both clients and servers, and incompatible with best practices for RESTful APIs.

JSContact ([[draft-ietf-jmap-jscontact](#)]) provides a simpler and more efficient representation for contact information. In addition, similarly to jCard, it provides a means to represent internationalised and unstructured contact information. Support for internationalised contact information has been recognised being necessary to facilitate the future internationalisation of registration data directory services.

1.2. 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.

2. JSContact

The JSContact specification defines a data model and JSON representation of contact information that can be used for data storage and exchange in address book or directory applications. It aims to be an alternative to the vCard data format ([[RFC6350](#)]) and to be unambiguous, extendable and simple to process. In contrast with jCard, it is not a direct mapping from the vCard data model and expands semantics where appropriate.

The JSContact specification declares two main object types: "JSCard", which represents a single contact "card", and "JSCardGroup" which represents a collection of JSCard objects. For the purpose of this document, only JSCard objects are considered.

JSCard differs from jCard in that it:

- o follows an object-oriented rather than array-oriented approach;
- o is simple to process;

- o requires no extra work in serialization/deserialization from/to a data model;
- o includes no "jagged" arrays;
- o prefers maps rather than arrays to implement collections;
- o is able to represent redacted contacts (both "name" and "fullName" properties are optional).

[[draft-ietf-jmap-jscontact-vcard](#)] provides informational guidance on the conversion of jCard objects into JSCard objects, and vice versa.

3. Using JSCard objects in RDAP Responses

Entity objects in RDAP responses MAY include a "jscard" property whose value is a JSCard object instead of the "vCardArray" property defined in [[RFC7483](#)].

Servers returning the "jscard" property in their response MUST include "jscard" in the "rdapConformance" array.

An example of an RDAP response containing a "jscard" property is shown in Figure 1. The "jscard" object in this example has been converted from the example included in [section 5.1 of \[RFC7483\]](#).

```
{
  "rdapConformance": [
    "rdap_level_0",
    "jscard"
  ],
  "objectClassName" : "entity",
  "handle": "XXXX",
  "jscard": {
    "uid": "XXXX",
    "fullName": { "value": "Joe User" },
    "kind": "individual",
    "preferredContactLanguages": {
      "fr": { "preference": 1 },
      "en": { "preference": 2 }
    },
    "organization": [ { "value": "Example" } ],
    "jobTitle": [ { "value": "Research Scientist" } ],
    "role": [ { "value": "Project Lead" } ],
    "addresses": [
      {
        "context": "work",
        "extension": "Suite 1234",
```



```
        "street": "4321 Rue Somewhere",
        "locality": "Quebec",
        "region": "QC",
        "postcode": "G1V 2M2",
        "country": "Canada",
        "coordinates": "geo:46.772673,-71.282945",
        "timeZone": "Canada/Eastern"
    },
    {
        "context": "private",
        "fullAddress": {
            "value": "123 Maple Ave\nSuite
90001\nVancouver\nBC\n1239\n"
        }
    }
],
"phones": [
    {
        "context": "work",
        "type": "voice",
        "labels": {
            "cell": true,
            "video": true,
            "text": true
        },
        "isPreferred": true,
        "value": "tel:+1-555-555-1234;ext=102"
    }
],
"emails": [
    {
        "context": "work",
        "value": "joe.user@example.com"
    }
],
"online": [
    {
        "context": "work",
        "type": "uri",
        "labels": { "key": true },
        "value": "http://www.example.com/joe.user/joe.asc"
    },
    {
        "context": "private",
        "type": "uri",
        "labels": { "url": true },
        "value": "http://example.org"
    }
]
```


]

```
}
"roles":[ "registrar" ],
"publicIds":[
  {
    "type":"IANA Registrar ID",
    "identifier":"1"
  }
],
"remarks":[
  {
    "description":[
      "She sells sea shells down by the sea shore.",
      "Originally written by Terry Sullivan."
    ]
  }
],
"links":[
  {
    "value":"http://example.com/entity/XXXX",
    "rel":"self",
    "href":"http://example.com/entity/XXXX",
    "type" : "application/rdap+json"
  }
],
"events":[
  {
    "eventAction":"registration",
    "eventDate":"1990-12-31T23:59:59Z"
  }
],
"asEventActor":[
  {
    "eventAction":"last changed",
    "eventDate":"1991-12-31T23:59:59Z"
  }
]
}
```

Figure 1: Example of "jscard" in RDAP response

3.1. RDAP Query Parameters

Two new query parameters are defined for the purpose of this document.

The query parameters are OPTIONAL extensions of path segments defined in [[RFC7482](#)]. They are as follows:

- o "jscard": a boolean value that allows a client to request the "jscard" property in the RDAP response;
- o "jcard": a boolean value that allows a client to request the "vcardArray" property in the RDAP response.

These parameters are furtherly explained in [Section 4](#).

[4.](#) Transition Considerations

[4.1.](#) RDAP Features Supporting a Transition Process

[4.1.1.](#) Notices and Link Relationships

RDAP allows servers to communicate service information to clients through notices. An RDAP response may contain one or more notice objects ([\[RFC7483\]](#), [Section 4.3](#)), each of which may include a set of link objects, which can be used to provide clients with references and documentation. These link objects may have a "rel" property which defines the relationship type, as described in [\[RFC8288\]](#), [Section 4](#). The transition process outlined in this document uses two types of link relation:

- o "deprecation", as described in [\[draft-ietf-httpapi-deprecation-header\]](#);
- o "alternate", as described in [\[RFC8288\]](#).

[4.1.2.](#) rdapConformance Property

The information about the specifications used in the construction of the response is also described by the strings which appear in the "rdapConformance" property of the RDAP response.

[4.1.3.](#) Query Parameters

Clients are able to ask servers to use specific RDAP features by using appropriate query parameters as described in [\[RFC7482\]](#).

[4.2.](#) Transition Procedure

The procedure for jCard to JSCard transition consists of four contiguous stages. During the procedure, the presence of "jscard" tag in the rdapConformance array indicates that JSCard is returned instead of jCard. The time format used to notify clients about this procedure is defined in [\[RFC3339\]](#).

Some elements of the following procedure are based on the best practices in [[API-DEPRECATION](#)].

[4.2.1.](#) Transition Stages

[4.2.1.1.](#) Stage 1: only jCard provided

This stage corresponds to providing jCard as default contact card ([[RFC7483](#)]). The RDAP server is not able to provide an alternate contact card. The rdapConformance array MUST NOT contain the "jscard" tag.

[4.2.1.2.](#) Stage 2: jCard sunset

During this stage, the server uses jCard by default, but the RDAP server will return JSCard if the client sets the query parameter "jscard" to a true value. The rdapConformance array MUST contain the "jscard" tag if JSCard is requested.

The RDAP server SHOULD include a notice titled "jCard sunset end". Such a notice should include a description reporting the jCard sunset end time and two links:

- o "deprecation": a link to a URI-identified resource documenting the jCard deprecation;
- o "alternate": if JSCard is not requested, a link to the JSCard version of same resource as identified by the current query string plus the parameter "jscard" set to a true value (Figure 2); otherwise, only the "deprecation" link is provided (Figure 3).


```

"notices": [
  {
    "title": "jCard sunset end",
    "description": ["2020-07-01T00:00:00Z"],
    "links": [{
      "value": "http://example.net/entity/XXXX",
      "rel": "deprecation",
      "type": "text/html",
      "href": "http://www.example.com/jcard_deprecation.html"
    },
    {
      "value": "http://example.net/entity/XXXX",
      "rel": "alternate",
      "type": "application/rdap+json",
      "href": " http://example.net/entity/XXXX?jscard=1"
    }
  ]
}
]

```

Figure 2: jCard sunset - JSCard not requested

```

"notices": [
  {
    "title": "jCard sunset end",
    "description": ["2020-07-01T00:00:00Z"],
    "links": [
      {
        "value": "http://example.net/entity/XXXX?jscard=1",
        "rel": "deprecation",
        "type": "text/html",
        "href": "http://www.example.com/jcard_deprecation.html"
      }
    ]
  }
]

```

Figure 3: jCard sunset - JSCard requested

4.2.1.3. Stage 3: jCard deprecation

This stage corresponds to the provisioning of JSCard by default, but the RDAP will return jCard if the client sets the query parameter "jscard" to a true value. The `rdapConformance` array contains the "jscard" tag unless jCard is requested. The "jscard" query parameter is ignored.

The RDAP server SHOULD to return a notice titled "jCard deprecation end". Such a notice should include a description reporting the jCard deprecation end time and two links:

- o "deprecation": a link to a URI-identified resource documenting the jCard deprecation;
- o "alternate": if jCard is not requested, a link to the jCard version of the same resource as identified by the current query string plus the parameter "jcard" set to 1/true/yes (Figure 4); otherwise, a link to the JSCard version of the same resource as identified by the current query string without the parameter "jcard" (Figure 5).

```
"notices": [  
  {  
    "title": "jCard deprecation end",  
    "description": ["2020-12-31T23:59:59Z"],  
    "links": [  
      {  
        "value": "http://example.net/entity/XXXX",  
        "rel": "deprecation",  
        "type": "text/html",  
        "href": "http://www.example.com/jcard_deprecation.html"  
      },  
      {  
        "value": "http://example.net/entity/XXXX",  
        "rel": "alternate",  
        "type": "application/rdap+json",  
        "href": " http://example.net/entity/XXXX?jcard=1"  
      }  
    ]  
  }  
]
```

Figure 4: jCard deprecation - jCard not requested


```
"notices": [  
  {  
    "title": "jCard deprecation end",  
    "description": ["2020-12-31T23:59:59Z"],  
    "links": [  
      {  
        "value": "http://example.net/entity/XXXX?jcard=1",  
        "rel": "deprecation",  
        "type": "text/html",  
        "href": "http://www.example.com/jcard_deprecation.html"  
      },  
      {  
        "value": "http://example.net/entity/XXXX?jcard=1",  
        "rel": "alternate",  
        "type": "application/rdap+json",  
        "href": " http://example.net/entity/XXXX"  
      }  
    ]  
  }  
]
```

Figure 5: jCard deprecation - jCard requested

4.2.1.4. Stage 4: jCard deprecated

This stage corresponds to providing JSCard as default contact card. The RDAP server is not able to provide an alternate contact card. The `rdapConformance` array always contains "jscard" tag. The RDAP server doesn't include any notice about the jCard deprecation process. Both "jscard" and "jcard" query parameters are ignored.

4.2.1.5. Length

The length of both jCard sunset and jCard deprecation periods are not fixed by this specification. Best practices in REST API deprecation suggest that, depending on the deprecated API's reach, user base and service offering, a convenient time could be anywhere between 3 - 8 months. Anyway, RDAP providers are recommended to monitor the server log to figure out whether declared times need to be changed to meet client requirements.

4.2.1.6. Goals

The procedure described in this document achieves the following goals:

- o only one contact representation would be included in the response;

- o the response would always be compliant to [\[RFC7483\]](#);
- o clients would be informed about the transition timeline;
- o the backward compatibility would be guaranteed throughout the transition;
- o servers and clients could execute their transitions independently.

5. Implementation Status

NOTE: Please remove this section and the reference to [RFC 7942](#) prior to publication as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC 7942](#) [[RFC7942](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC 7942](#), "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

[5.1.](#) IIT-CNR/Registro.it

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 a "proof of concept" research implementation.

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

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

6. IANA Considerations

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

Extension identifier: jscard

Registry operator: Any

Published specification: This document.

Contact: IETF <iesg@ietf.org>

Intended usage: This extension represents a contact card provided in an RDAP response according to the JSContact specification ([\[draft-ietf-jmap-jscontact\]](#)).

7. Security Considerations

Unlike jCard, the formatted name as well as any other personally identifiable information is not required in JSCard. The only mandatory property, namely "uid", is usually an opaque string. Therefore, redacted properties can be merely excluded without using placeholder values.

8. References

8.1. Normative References

- [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>>.
- [RFC6350] Perreault, S., "vCard Format Specification", [RFC 6350](#), DOI 10.17487/RFC6350, August 2011, <<https://www.rfc-editor.org/info/rfc6350>>.

- [RFC7095] Kewisch, P., "jCard: The JSON Format for vCard", [RFC 7095](#), DOI 10.17487/RFC7095, January 2014, <<https://www.rfc-editor.org/info/rfc7095>>.
- [RFC7482] Newton, A. and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Format", [RFC 7482](#), DOI 10.17487/RFC7482, March 2015, <<https://www.rfc-editor.org/info/rfc7482>>.
- [RFC7483] Newton, A. and S. Hollenbeck, "JSON Responses for the Registration Data Access Protocol (RDAP)", [RFC 7483](#), DOI 10.17487/RFC7483, March 2015, <<https://www.rfc-editor.org/info/rfc7483>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [BCP 205](#), [RFC 7942](#), DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [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>>.

8.2. Informative References

- [API-DEPRECATION] Sandoval, K., "How to Smartly Sunset and Deprecate APIs", August 2019, <<https://web.archive.org/web/20200417084255/https://nordicapis.com/how-to-smartly-sunset-and-deprecate-apis/>>.
- [[draft-ietf-httpapi-deprecation-header](#)] Dalal, S. and E. Wilde, "The Deprecation HTTP Header Field", <<https://datatracker.ietf.org/doc/draft-ietf-httpapi-deprecation-header/>>.
- [[draft-ietf-jmap-jscontact](#)] Stepanek, R. and M. Loffredo, "JSContact: A JSON representation of contact data", <<https://datatracker.ietf.org/doc/draft-ietf-jmap-jscontact/>>.

[[draft-ietf-jmap-jscontact-vcard](#)]

Loffredo, M. and R. Stepanek, "JSContact: Converting from and to vCard", <<https://datatracker.ietf.org/doc/draft-ietf-jmap-jscontact-vcard/>>.

[RDAP-PILOT-WG]

ICANN RDAP Pilot WG, "RDAP Pilot Report", April 2019, <<https://www.icann.org/en/system/files/files/rdap-pilot-report-25apr19-en.pdf>>.

[Appendix A](#). Change Log

[A.1](#). Change from 00 to 01

1. Changed category from "Best Current Practice" to "Standards Track"
2. Replaced the example of Figure 1
3. Changed the title of the "Migration from JCard to JSCard" section to "Transition Considerations"
4. Added [Section 3.1](#)
5. Updated [Section 6](#)
6. Updated [Section 7](#)
7. Rearranged the description of stage 1 in [Section 4.2.1](#)
8. Changed the names of the transition stages 1 and 2
9. Corrected Figure 2, Figure 4, Figure 5
10. Changed the rdapConformance tag "jscard_level_0" to "jscard"
11. Removed the "Best Practices for deprecating a REST API features" section, but added a useful reference.

[A.2](#). Change from 01 to 02

1. Removed the sentence "which cannot be represented using jCard" in [Section 1.1](#).

A.3. Change from 02 to 03

1. Updated section "Conventions Used in This Document".
2. Updated the contact in "IANA Considerations" section.
3. Changed the reference [draft-loffredo-jmap-jscontact-vcard](#) to [draft-ietf-jmap-jscontact-vcard](#).
4. Added reference to [RFC8174](#).
5. Other minor edits.

A.4. Change from 03 to 04

1. Updated the reference [draft-dalal-deprecation-header](#) to [draft-ietf-httpapi-deprecation-header](#).

Authors' Addresses

Mario Loffredo
IIT-CNR/Registro.it
Via Moruzzi,1
Pisa 56124
IT

Email: mario.loffredo@iit.cnr.it
URI: <http://www.iit.cnr.it>

Gavin Brown
CentralNic Group plc
Saddlers House, 44 Gutter Lane
London, England EC2V 6BR
GB

Phone: +44 20 33 88 0600
Email: gavin.brown@centralnic.com
URI: <https://www.centralnic.com>

