

Deprecating jCard in RDAP: why and how

Mario Loffredo - IIT-CNR/Registro.it

Gavin Brown – CentralNIC Registry

Why to deprecate jCard in RDAP ?

- jCard is the most frequent concern by RDAP developers ("RDAP Pilot Report", April 2019):
 - considered unintuitive
 - very complicated to implements for both clients and servers
 - incompatible with best practices for RESTful APIs
- JSContact specification provides a simpler and more efficient representation for contacts

JSContact overview

- A contact card representation alternative to vCard
 - JSON-based
 - vCard content supported, but no direct mapping
 - highly structured
 - topmost objects are **JSCard** and JSCardGroup
- Described by:
 - Stepanek R., Loffredo M.: “*JSContact: A JSON representation of contact data*”, <https://datatracker.ietf.org/doc/draft-ietf-jmap-jscontact/>
 - Loffredo M., Stepanek R.: “*JSContact: Converting from and to vCard*”, <https://datatracker.ietf.org/doc/draft-loffredo-jmap-jscontact-vcard/>

JSCard features

- JSCard can represent the same information as jCard
- JSCard is more efficient than jCard:
 - requires no extra work in serialization/deserialization
 - provides a means to represent internationalised and unstructured contact information
 - is able to represent redacted contacts
 - Is simple to process:
 - is object-oriented rather than array-oriented
 - includes no jagged arrays
 - prefers maps rather than arrays in implementing collections
- jCard-2-JSCard mapping examples are available at <https://github.com/mario-loffredo/jcard-deprecation/>

Transition procedure

1. Only jCard provided
 - jCard provided as default contact card
 - no other format available
2. jCard sunset
 - jCard provided as default contact card
 - JSCard requested by query parameter jscard=1
 - "jscard" returned instead of "vcardArray"
3. jCard deprecation
 - JSCard provided as default contact card
 - jCard requested by query parameter jcard=1
 - "vcardArray" returned instead of "jscard"
4. jCard deprecated
 - JSCard provided as default contact card

Note: "jscard" value must be added to rdapConformance whenever "jscard" is returned

Transition – jCard sunset notices

JSCard not requested

```
"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"
      }
    ]
  }
]
```

JSCard 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"
      }
    ]
  }
]
```

Transition – jCard deprecation notices

jCard not requested

```
"notices": [
  {
    "title": "jCard deprecation end",
    "description": ["2021-01-01T00:00:00Z"],
    "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?jcard=1"
      }
    ]
  }
]
```

jCard requested

```
"notices": [
  {
    "title": "jCard deprecation end",
    "description": ["2021-01-01T00:00:00Z"],
    "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"
      }
    ]
  }
]
```

Transition – Stages length

- The length of stages 2 and 3 is not fixed
- REST API best practices suggest that a convenient time could be anywhere between 3 - 8 months
- RDAP providers are recommended to monitor the server log to figure out whether times need to be changed

Transition - Goals

- Only one contact representation would be included in the response
- The response would always be compliant to RFC7483
- Clients would be informed about the transition timeline
- The backward compatibility would be guaranteed throughout the transition
- Servers and clients could execute their transitions independently

Documentation

- Loffredo M., Brown G.: "Using JSContact in Registration Data Access Protocol (RDAP) JSON Responses",
<https://datatracker.ietf.org/doc/draft-loffredo-regext-rdap-jcard-deprecation/>
- GitHub project: <https://github.com/mario-loffredo/jcard-deprecation/>
- Feedback appreciated!!

Questions

