calext
Internet-Draft
Intended status: Standards Track
Expires: 15 May 2026

R. Stepanek
Fastmail
M. Loffredo
IIT-CNR

11 November 2025

Protocol-Specific Profiles for JSContact
draft-ietf-calext-jscontact-profiles-09

Abstract

This document defines JSContact profiles, an IANA registry for named subsets of JSContact elements. It aims to facilitate using JSContact in context of contact data exchange protocols or other use cases, in which supporting all of JSContact semantics might be inappropriate.

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 15 May 2026.

Table of Contents

1. Notational Conventions

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.

The ABNF definitions in this document use the notations of [RFC5234]. ABNF rules not defined in this document are defined in [RFC5234] (such as the ABNF for DIGIT).

2. Introduction

The JSContact [RFC9553] contact card data model and format is designed for use in address book applications and directory services. Intended as an alternative to the prevalent vCard [RFC6350] data format, it covers vCard core semantics and extensions, and provides a rich model for personal names, postal addresses and localization. All JSContact elements are relevant for some contact card use case and, similar to vCard, implementations are expected to support these elements when exchanging contact card information using protocols such as CardDAV [RFC6352] and JMAP for Contacts [RFC9610].

In contrast, other protocols and internet standards might require exchanging _some_ contact card information, but not all of what JSContact provides. Section 1.7.4 of [RFC9553] outlines how JSContact implementations may ignore unknown JSContact elements, but

this only applies to future extensions of [RFC9553]; they are still expected to implement all elements of the core specification. Also, the extensibility of JSContact and the requirement to preserve arbitrary contact elements might not be adequate for some protocols.

To make use of JSContact under these circumstances, this document defines a new IANA registry for JSContact that allows for registering named subsets of JSContact elements. These subsets are referred to as "JSContact profiles" and are meant to bring the following benefits:

* Protocol designers might be encouraged to use JSContact, rather than coming up with their own contacts format. This facilitates cross-protocol data exchange and migration.

* Different protocols use the same IANA registry to express which JSContact elements they support. This facilitates understanding their commonalities and reusing existing profiles.

* A central registry provides implementors of JSContact libraries with a consistent format documenting which profile supports what elements, rather than having to look up that information from possibly distinctly organized internet drafts.

This document is organized as follows: Section 3 defines JSContact profiles, Section 4 illustrates JSContact profiles by example, Section 5 summarizes the relevant information for IANA to establish the JSContact Profiles registry.

3. JSContact Profiles

A JSContact profile is a named, versioned subset of JSContact properties, value types and values. These JSContact elements MAY be further restricted by the profile, but a profile can not loosen restrictions. For example, a profile can define an originally optional property to become mandatory, but it can not make a mandatory property become optional.

The JSContact elements MUST be registered in the IANA JSContact registry. A JSContact extension MAY define both a new profile and new properties or other elements, as long as they are registered at the same time. A JSContact profile MUST be registered at IANA (see Section 5). Section 3.1 and Section 3.2 define how to name and version a JSContact profile, Section 3.3 defines how to define the properties supported by that profile.

A JSContact object conforms to a profile if all its properties are in the subset of properties defined by that profile and the property values comply with the profile restrictions for that property. A conforming JSContact object not necessarily is valid in context of the protocol or use case that defined this profile, specifications MAY define further restrictions.

3.1. Profile Name

This names the profile. A JSContact profile MUST have a unique name. The name MUST only contain ASCII lowercase alphabetic and numeric characters, optionally separated by hyphens. It MUST start with an alphabetic character and it MUST be of at least 1 character and at most 255 characters in size. Formally, it MUST be a valid "profile-name" defined in Figure 1.

```
profile-name = LALPHA *( ["-"] LALPHA / DIGIT ) ; at most 255 octets in size

LALPHA       = %x61-7A ; a-z
```

Figure 1: ABNF Rule for JSContact Profile Name

3.2. Profile Version

This defines the current version of the profile. Each profile is versioned independently and version numbers are not related to values in the IANA JSContact Versions registry. The version identifier MUST be a valid "jsversion" value as defined in Section 1.9.1 of [RFC9553]. Any addition to the list of JSContact properties supported by that profile (Section 3.3) MUST update the current version.

As a note, if a semantic versioning scheme is not adequate for protocol designers making use of JSContact profiles, then an alternative approach is to register a new JSContact profile for each new version.

3.3. Supported Properties

A profile defines a list of property entries that together determine the set of properties supported by that profile. Each entry consists of the following elements:

Property Name: This is the name of the JSContact property that this entry refers to. This MUST be the property name registered in the "JSContact Properties" registry.

Property Context: This is a comma-separated list of JSContact object

types that support this property in this profile. Each of these types MUST also be listed in the property contexts for this property in the "JSContact Properties" registry, but a profile MAY only support the property in a subset of these contexts.

Restricted to be Mandatory: This restricts the property to be mandatory in the listed property contexts, despite the property originally being defined to be optional in the same contexts. The verbatim value "mandatory" indicates that it is mandatory in this profile, the absence of any value indicates that the property is mandatory or optional according to its original definition. This means that a profile can make an optional property become mandatory, but it can not make a mandatory property become optional.

Restricted Property Type: This restricts the property value type in the listed property contexts, despite the property originally being defined to have a less restrictive type definition. If set, the value MUST be a type signature as defined in Section 1.3.2 of [RFC9553] and it MUST be a subtype of the original type signature for these property contexts. The absence of any value indicates that the original type signature applies. This means that a profile can restrict the value types of a property, but it can not add value types for that property.

Restricted Enum Values: This restricts the enumerated values defined for this property to a subset of those values. The values MUST be listed in the "JSContact Enum Values" registry for this property and context. The allowed values are separated by comma, the absence of any value indicates that all enumerated values are allowed. A profile MAY exclude the default enumerated value of a property, in which case all object instances MUST have this property set to one of the allowed values.

Restricted PatchObject Keys: This restricts the PatchObject value of this property such that each JSON Pointer key in the PatchObject MUST consist of exactly one JSON Pointer reference token [RFC6901] (Section 3). Doing so restricts a PatchObject to only patch properties that are set in the same object and only replace their values entirely. The verbatim value "Yes" indicates that this profile does restrict PatchObject keys, the absence of any value indicates that it does not restrict them. This MUST NOT be set to "Yes", if the property value type is not a PatchObject.

Since Profile Version: This indicates the highest profile version in which this property definition was added or updated.

All profiles MUST support "@type" and "version", and therefore profiles MUST NOT include entries for these properties.

The set of supported JSContact properties is then determined by the property entries as follows:

1. Initialize the set with all properties of the Card object for which a property entry names this property and contains "Card" in the Property Context. If no property entry contains "Card" in the Property Context, initialize the set with all IANA-registered Card object properties.

2. For every property in the set having an object type as value type, add all properties for which a property entry names that property and contains the object type in the Property Context. If no property entry contains the object type in the Property Context, add all IANA-registered properties for that object type to the set.

3. Repeat the previous step until no further additions are possible.

A Card object is supported by the profile, if all its properties are part of the set of supported properties and all property values are valid according to the restrictions defined in the applicable property entries.

4. Example Profile

This section provides an example for a JSContact profile, as it would be registered in the IANA JSContact Profiles registry. See Section 5.1.1 and Section 5.1.2 for the exact meaning of each registry item. This profile is just for illustration, it is not registered at IANA.

Name:
   jscontact-simple

Profile Version:
   1.0

Properties:

Property Name	Property Context	Restricted to be Mandatory	Restricted Property Type	Restricted Enum Values	Restricted PatchObject Keys	Since Profile Version
addresses	Card					1.0
emails	Card					1.0
kind	Card			individual,org		1.0
localizations	Card				Yes	1.0
name	Card					1.0
full	Address	mandatory				1.0
components	Name					1.0
full	Name					1.0
kind	NameComponent					1.0
value	NameComponent					1.0

Table 1: Properties of the "jscontact-simple" profile

This profile describes contact cards that can only contain:

* Contact cards for individuals and organizations, but no other kinds such as groups or devices.

* Full postal address lines, but no address components or any other property of the Address object type.

* Full names and name components, but no other properties of the

Name object type.

* Email addresses, where all properties of the EmailAddress object are supported.

* Localizations, with the restriction that the PatchObject must only patch properties of the Card object.

The following Card object conforms to that profile:

```
{
  "@type": "Card",
  "version": "1.0",
  "name": {
     "components": [
      { "kind": "given", "value": "Hayao" },
      { "kind": "surname", "value": "Miyazaki" }
    ]
  },
  "addresses": {
    "a1": {
      "full": "71 Cherry Court, Somewhere, 123SO, UK"
    }
  },
  "emails": {
    "e1": {
      "address": "hayao@example.com"
    }
  },
  "localizations": {
    "jp": {
      "name": {
        "components": [
         { "kind": "surname", "value": "宮崎" },
         { "kind": "given", "value": "駿" }
        ]
      }
    }
  }
}
```

Note that:

* The Address, Card, Name, and NameComponent object values only contain properties for which a property entry exists in the profile.

* The EmailAddress object contains the "address" property. This is allowed because the profile contains a property entry for the "emails" property of the Card object, but it does not define a property entry for the EmailAddress object type. Consequently, all properties of the EmailAddress object can be set.

* The "full" property of the Address object is set. It is the only property allowed to be set in that profile for Address, and the "full" property is mandatory for this profile.

* The "kind" property of the the NameComponent objects is set to "surname" and "given", respectively. Since the profile does not restrict the enumerated values of this property, all valid NameComponent "kind" property values are supported.

* The "kind" property of the Card object is not set. The profile restricts the enumerated values of this property to "individual" and "org". Since "individual" is the default value for this property, there is no need to set it.

5. IANA Considerations

5.1. Creation of the JSContact Profile Registry

IANA will add the "JSContact Profile" registry to the "JSContact" registry group originally created in Section 3.2 of [RFC9553]. The purpose of this new registry is to describe protocol-specific profiles for JSContact data. The registry policy and change procedure of the "JSContact" registry group apply.

5.1.1. JSContact Profile Registry Template

This template defines how to register a new JSContact profile. It consists of the following items:

* Profile Name

* Profile Version

* Properties

Section 3.1 and Section 3.2 define the Profile Name and Profile Version item, respectively. For the Properties item, IANA will create a subregistry for each profile name (e.g. "Properties for xyz"), listing the properties supported by that profile. Each entry in that list is registered using the template defined in Section 5.1.2.

5.1.2. JSContact Profile Property Template

This template consists of the following items:

* Property Name

* Property Context

* Restricted to be Mandatory

* Restricted Property Type

* Restricted Enum Values

* Restricted PatchObject Keys

* Since Profile Version

See Section 3.3 for the definitions of these items.

5.1.3. Initial Contents of the JSContact Profile Registry

This document does not define any initial contents for the newly created registries.

6. Security Considerations

This document does not provide new security considerations. The security considerations of Section 4 of [RFC9553] apply.

7. References

7.1. 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>.

[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>.

[RFC6901] Bryan, P., Ed., Zyp, K., and M. Nottingham, Ed., "JavaScript Object Notation (JSON) Pointer", RFC 6901, DOI 10.17487/RFC6901, April 2013, <https://www.rfc-editor.org/info/rfc6901>.

[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>.

[RFC9553] Stepanek, R. and M. Loffredo, "JSContact: A JSON Representation of Contact Data", RFC 9553, DOI 10.17487/RFC9553, May 2024, <https://www.rfc-editor.org/info/rfc9553>.

7.2. Informative References

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

[RFC6352] Daboo, C., "CardDAV: vCard Extensions to Web Distributed Authoring and Versioning (WebDAV)", RFC 6352, DOI 10.17487/RFC6352, August 2011, <https://www.rfc-editor.org/info/rfc6352>.

[RFC9610] Jenkins, N., Ed., "JSON Meta Application Protocol (JMAP) for Contacts", RFC 9610, DOI 10.17487/RFC9610, December 2024, <https://www.rfc-editor.org/info/rfc9610>.

Authors' Addresses

Robert Stepanek
Fastmail
PO Box 234
Collins St. West
Melbourne VIC 8007
Australia
Email: rsto@fastmailteam.com

Mario Loffredo
IIT-CNR
Via Moruzzi, 1
56124 Pisa
Italy
Email: mario.loffredo@iit.cnr.it