Authors: R. Stepanek   M. Loffredo
                Fastmail   *IIT-CNR*

RFC 9554
vCard Format Extensions for JSContact

Abstract

This document defines a set of new properties for vCard and extends the use of existing ones. Their primary purpose is to align the same set of features between the JSContact and vCard formats, but the new definitions also aim to be useful within just the vCard format. This document updates RFC 6350 ("vCard Format Specification").

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9554.

Copyright Notice

Table of Contents

1. Introduction

The JSContact [RFC9553] format aims to be an alternative to the vCard [RFC6350] format for representation of contact and address book data. As such, it introduces new semantics that are not covered in the current definition of vCard and its various extensions. Converting contact data between the two formats is defined in [RFC9555] with the goal of not losing any semantics during conversion. To achieve this, this document defines a new set of properties for vCard and extends existing definitions.

1.1. 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.

1.2. ABNF Notations

The ABNF definitions in this document use the notations of [RFC5234]. ABNF rules not defined in this document are defined in either [RFC5234] (such as the ABNF for CRLF, WSP, DQUOTE, VCHAR, ALPHA, and DIGIT) or [RFC6350].

2. Updated Properties

2.1. ADR

This specification modifies the definition of the ADR property. It extends its structured value with additional address components to better support the variety of international addresses. It separates the address parts, which currently are typically combined in street address component values, into distinct components.

Implementations **SHOULD** write a combined value of these components in the street address component for backwards compatibility, but they **SHOULD** ignore the street component during reads if the ADR property value contains any of the new components.

The following change is made to the first paragraph under "Special note", as originally specified in Section 6.3.1 of [RFC6350]. The remaining paragraphs of that section in the original specification still apply.

Special note: The structured type value consists of a sequence of address components. The component values **MUST** be specified in their corresponding position. The structured type value corresponds, in sequence, to the

post office box;

extended address (e.g., apartment or suite number);

street address;

locality (e.g., city);

region (e.g., state or province);

postal code;

country name (full name in the language specified in Section 5.1 of [RFC6350]);

room, suite number, or identifier;

apartment number, extension designation, or box number;

building floor or level;

street number;

street name;

building, tower, or condominium;

block name or number;

subdistrict;

district;

landmark or another publicly known prominent feature that can substitute the street name and number (e.g., "White House" and "Taj Mahal"); and

the cardinal direction or quadrant (e.g., "north").

The following change is made to the definition of "ADR-value" under "ABNF", as originally specified in Section 6.3.1 of [RFC6350].

ABNF

```
ADR-value = ; defined in RFC 6350, Section 6.3.1.:
            ADR-component-pobox ";"
            ADR-component-ext ";"
            ADR-component-street ";"
            ADR-component-locality ";"
            ADR-component-region ";"
            ADR-component-code ";"
            ADR-component-country ";"
            ; defined in this document:
            ADR-component-room ";"
            ADR-component-apartment ";"
            ADR-component-floor ";"
            ADR-component-streetnumber ";"
            ADR-component-streetname ";"
            ADR-component-building ";"
            ADR-component-block ";"
            ADR-component-subdistrict ";"
            ADR-component-district ";"
            ADR-component-landmark ";"
            ADR-component-direction

ADR-component-pobox    = list-component
ADR-component-ext      = list-component
ADR-component-street   = list-component
ADR-component-locality = list-component
ADR-component-region   = list-component
ADR-component-code     = list-component
ADR-component-country  = list-component
ADR-component-room     = list-component
ADR-component-apartment = list-component
ADR-component-floor    = list-component
ADR-component-streetnumber = list-component
ADR-component-streetname = list-component
ADR-component-building = list-component
ADR-component-block    = list-component
ADR-component-subdistrict = list-component
ADR-component-district = list-component
ADR-component-landmark = list-component
ADR-component-direction = list-component
```

The following change is made under "Example", as originally specified in Section 6.3.1 of [RFC6350].

Example: In this example, the post office box and the extended address components are absent. The street number and name are both added as separate components and are combined in the street component for backwards compatibility.

```
ADR;GEO="geo:12.3457,78.910":
   ;;123 Main Street;Any Town;CA;91921-1234;U.S.A
   ;;;;123;Main Street;;;;;;
```

2.2. N

This specification modifies the definition of the N property. It extends its structured value with additional name components to better support international names and generation markers. In doing so, this also facilitates formatting N property values using the Unicode Common Locale Data Repository (CLDR) Person Name [CLDRPersonName] formatting standard.

One new component is for secondary surnames, because in some cultures, such secondary surname kinds are used to indicate the paternal and maternal family names or generational names indicating father or grandfather. Another new component indicates a generation ("II", "XVI") or parental relation ("Jr.", "Sr.").

Currently, implementations typically place secondary surnames in the family name component and generational markers in the honorific suffixes component. For backwards compatibility, implementations **SHOULD** add such values to both the newly defined components and their backwards-compatible counterpart. Reading N property values, implementations **SHOULD** ignore any value in the backwards-compatible component if an equal value is set in the new component accordingly. For example, a "Jr." that occurs in both honorific suffixes and generation should only be handled as a generational marker.

The following change is made to the first paragraph under "Special note", as originally specified in Section 6.2.2 of [RFC6350]. The remaining paragraphs of that section in the original specification still apply.

Special note: The structured property value corresponds, in sequence, to the

> family names (also known as surnames);
>
> given names;
>
> additional names;
>
> honorific prefixes;
>
> honorific suffixes;
>
> secondary surname; and
>
> generation.

The following change is made under "ABNF", as originally specified in Section 6.2.2 of [RFC6350].

ABNF

```
N-param = "VALUE=text" / sort-as-param / language-param
            / altid-param / any-param
N-value = list-component 6(";" list-component)
```

The following change is made under "Examples", as originally specified in Section 6.2.2 of [RFC6350].

Examples

```
N:Public;John;Quinlan;Mr.;Esq.

N:Stevenson;John;Philip,Paul;Dr.;Jr.,M.D.,A.C.P.;;Jr.
```

No change is required for the definition of the SORT-AS parameter, but the new components also apply for use with this parameter.

3. New Properties

3.1. CREATED

Property name: CREATED

Purpose: Defines the date and time when the vCard was created.

Value type: A single timestamp value.

Cardinality: *1

Property parameters: VALUE

Description: This is the timestamp when the vCard was created. Copying the vCard across systems does not count as a new creation nor a new revision. Instead, the timestamp value typically stays unchanged for the existence of the vCard.

Format definition:

```
created        = "CREATED" createdparam ":" timestamp

createdparam   = *(
                   ;
                   ;
                   ; The following are OPTIONAL
                   ; but MUST NOT occur more than once.
                   ;
                   (";" "VALUE" "=" "timestamp") /
                   ;
                   ; The following are OPTIONAL
                   ; and MAY occur more than once.
                   ;
                   (";" any-param)
                   ;
                   )
```

Example(s):

```
CREATED:20220705T093412Z
CREATED;VALUE=TIMESTAMP:20211022T140000-05
```

3.2. GRAMGENDER

Property name: GRAMGENDER

Purpose: Defines which grammatical gender to use in salutations and other grammatical constructs.

Value type: A single text value that is restricted to an enumerated list of allowed values.

Cardinality: *

Property parameters: LANGUAGE

Description: This property defines the grammatical gender that the contact prefers to be addressed by or referred to as in written or spoken form. For example, the German language distinguishes by grammatical gender in salutations such as "Sehr geehrte" (feminine) and "Sehr geehrter" (masculine). Multiple occurrences of this property MUST be distinguished by the LANGUAGE parameter.

Format definition:

```
gramgender        = "GRAMGENDER" gramgender-param
                        ":" gramgender-value

gramgender-param =
                  *(
                     ;
                     ; The following are OPTIONAL
                     ; but MUST NOT occur more than once.
                     ;
                     (";" language-param) /
                     ;
                     ; The following are OPTIONAL
                     ; and MAY occur more than once.
                     ;
                     (";" any-param)
                     ;
                     )
gramgender-value = "animate" /
                   "common" /
                   "feminine" /
                   "inanimate" /
                   "masculine" /
                   "neuter" /
                   iana-token /
                   x-name
```

Example(s):

```
GRAMGENDER;LANGUAGE=de:feminine
```

3.3. LANGUAGE

Property name: LANGUAGE

Purpose: Defines the default language that human-readable text values in this vCard are assumed to be written in.

Value type: A single Language-Tag value as defined in Section 4 of [RFC6350].

Cardinality: *1

Property parameters: The LANGUAGE parameter **MUST NOT** be assigned to this property.

Description: This property defines the language that property values of type TEXT are assumed to be written in for this vCard. If a vCard property includes the LANGUAGE parameter, then the parameter value has higher precedence than the LANGUAGE property value.

Format definition:

```
language-prop      = "LANGUAGE" any-param ":" Language-Tag
                   ; Language-Tag is defined in RFC 6350, Section 4.
```

Example(s):

```
LANGUAGE:de-AT
```

3.4. PRONOUNS

Property name: PRONOUNS

Purpose: Defines the pronouns that shall be used to refer to the entity represented by this vCard.

Value type: A single text value.

Cardinality: *

Property parameters: LANGUAGE, PREF, TYPE, ALTID

Description: This property contains the pronouns that the contact chooses to use for themselves. The value is free-form text. These pronouns shall be used when addressing or referring to the contact. Multiple occurrences of this property **MAY** define pronouns for multiple languages, preferences, and contexts. Multiple pronouns in the same language **SHOULD** use the PREF parameter; otherwise, the order of preference is implementation-specific.

Format definition:

```
pronouns       = "PRONOUNS" pronouns-param ":" text

pronouns-param =
                *(
                  ;
                  ; The following are OPTIONAL
                  ; but MUST NOT occur more than once.
                  ;
                  (";" language-param) /
                  (";" pref-param) /
                  (";" type-param) /
                  (";" altid-param) /
                  ;
                  ; The following are OPTIONAL
                  ; and MAY occur more than once.
                  ;
                  (";" any-param)
                  ;
                  )
```

Example(s):

```
PRONOUNS;LANGUAGE=en;PREF=1:xe/xir
PRONOUNS;LANGUAGE=en;PREF=2:they/them
```

3.5. SOCIALPROFILE

Property name: SOCIALPROFILE

Purpose: Specifies the URI or username for social media profiles associated with the object the vCard represents.

Value type: A single URI or TEXT value. The default value type is URI.

Cardinality: *

Property parameters: The SERVICE-TYPE parameter **MUST** be assigned to this property if the value type is TEXT, and it **MAY** be assigned if the value type is URI. In either case, it **MUST NOT** be assigned more than once.

Description: Several vCard address book implementations currently use an experimental X-SOCIALPROFILE property to store social media profiles for contacts. This specification provides an IANA-registered property for the same purpose. In addition to the typical use of this property with URI values, it also allows setting usernames for social media services as free-text TEXT values, in which case the service name **MUST** be provided as a parameter. Names **MUST** be considered equal if they match case-insensitively.

Format definition:

```
socialpr       = "SOCIALPROFILE" socialpr-param ":"
                                  socialpr-value

socialpr-param = "VALUE=uri" / "VALUE=text" /
                 service-type-param / any-param

socialpr-value = URI / text
```

Example(s):

```
SOCIALPROFILE;SERVICE-TYPE=Mastodon:https://example.com/@foo
SOCIALPROFILE:https://example.com/ietf
SOCIALPROFILE;SERVICE-TYPE=SomeSite;VALUE=text:peter94
```

4. New Parameters

4.1. AUTHOR

Parameter name: AUTHOR

Purpose: Identifies the author of the associated property value.

Description: This parameter **MAY** be set on any property where conveying authorship is desired. It identifies the author as a URI [RFC3986]. Since every valid URI includes the COLON (U+003A) character, the parameter value **MUST** be quoted. As an alternative or in addition to this parameter, the AUTHOR-NAME parameter allows naming an author as a free-text value (see Section 4.2).

Format definition:

```
author-param     = "AUTHOR" "=" DQUOTE URI DQUOTE
```

Example(s):

```
NOTE;AUTHOR="mailto:john@example.com":This is some note.
```

4.2. AUTHOR-NAME

Parameter name: AUTHOR-NAME

Purpose: Names the author of the associated property value.

Description: This parameter **MAY** be set on any property where conveying authorship is desired. It names the author as a free-text value. The parameter value **MUST NOT** be empty. Implementations **MUST** take care to quote the name part; if otherwise, the part will not be a valid "param-value" (see Section 3.3 of [RFC6350]). As an alternative or in addition to this parameter, the AUTHOR parameter allows identifying an author by URI (see Section 4.1).

Format definition:

```
author-name-param     = "AUTHOR-NAME" "=" param-value ; not empty
```

Example(s):

```
NOTE;AUTHOR-NAME=John Doe:This is some note.
NOTE;AUTHOR-NAME="_:l33tHckr:_":A note by an unusual author name.
```

4.3. CREATED

Parameter name: CREATED

Purpose: Defines the date and time when a property was created in a vCard.

Description: This parameter **MAY** be set on any property to define the point in time when the property was created. The value **MUST** be a valid TIMESTAMP value as defined in Section 4.3.5 of [RFC6350]. Generally, updating a property value **SHOULD NOT** change the creation timestamp.

Format definition:

```
created-param = "CREATED" "=" param-value
              ; a valid TIMESTAMP of Section 4.3.5 of RFC 6350
```

Example(s):

```
NOTE;CREATED=20221122T151823Z:This is some note.
```

4.4. DERIVED

Parameter name: DERIVED

Purpose: Specifies that the value of the associated property is derived from some other property values in the same vCard.

Description: This property parameter **SHOULD** be specified on a property if the property value is derived from some other properties in the same vCard. When present with a value of "true", clients **MUST NOT** update the property.

As an example, an implementation may derive the value of the FN property from the name components of the N property. It indicates this fact by setting the DERIVED parameter on the FN property to "true".

Format definition:

```
derived-param    = "DERIVED" "=" ("true" / "false")
; Default is false
```

Example(s):

```
N:;John;Quinlan;Mr.;
FN;DERIVED=TRUE:Mr. John Quinlan
```

4.5. LABEL

Parameter name: LABEL

Purpose: Used with the ADR property. Its value contains a formatted text representation of that address, e.g., for delivery.

Description: Section 6.3.1 of [RFC6350] defines the ADR property, noting that the property can also include a LABEL parameter to present a delivery address label for the address. But this parameter was not included in the IANA "vCard Parameters" registry (Section 10.3.2 of [RFC6350]) and, accordingly, is not a registered standard vCard element. This specification defines and registers the LABEL parameter for use with the ADR property as originally intended.

Format definition:

```
label-param     = "LABEL" "=" param-value
```

Example(s):   The LABEL parameter as illustrated in the ADR property example in Section 6.3.1 of [RFC6350].

```
ADR;LABEL="Mr. John Q. Public, Esq.\nMail Drop: TNE QB\n123
  Main Street\nAny Town, CA  91921-1234\nU.S.A.":
  ;;123 Main Street;Any Town;CA;91921-1234;U.S.A.
```

4.6. PHONETIC

Parameter name: PHONETIC

Purpose: Defines how to pronounce the value of another property in the same vCard.

Description: This property parameter indicates that the value of its property contains the phonetic representation of another same-named property in the same vCard. Exemplary uses are defining how to pronounce Japanese names and romanizing Mandarin or Cantonese names and address components.

The parameter value indicates the phonetic system and **MUST** be one of the values enumerated in the IANA "vCard Parameter Values" registry (Section 7.4). This specification defines the following values:

"ipa":   denotes the International Phonetic Alphabet [IPA].

"jyut":   denotes the Cantonese romanization system "Jyutping".

"piny":   denotes the Standard Mandarin romanization system "Hanyu Pinyin".

"script":   denotes the unknown phonetic system. The SCRIPT (Section 4.8) parameter **MUST** be set in addition to the PHONETIC parameter.

The value type of the property on which the PHONETIC parameter is set **MUST** be of the same type as its related property. If a component value is set in the property on which the PHONETIC parameter is set, then a component value also **MUST** be set at that same position in the related property. On the other hand, not every component value in the related property needs to have a phonetic representation.

The ALTID (Section 5.4 of [RFC6350]) parameter **MUST** be set with equal values on both the related property and the property having the PHONETIC parameter set. If more than one same-named property has both the PHONETIC parameter set and an equal ALTID parameter value, then at most, one of these properties **MAY** not have the LANGUAGE parameter set, and all others **MUST** have the LANGUAGE parameter set. The LANGUAGE parameters **MUST NOT**

have equal values. The LANGUAGE parameter value **SHOULD NOT** contain a script subtag in its Language-Tag value, and any such subtag **MUST** be ignored in favor of the SCRIPT (Section 4.8) parameter value.

This specification defines the PHONETIC parameter for use with the ADR and N properties.

Format definition:

```
phonetic-param = "PHONETIC=" phonetic-value

phonetic-value = "ipa" / "piny" / "jyut" / "script" /
                 iana-token / x-name
```

Example(s):

```
N;ALTID=1;LANGUAGE=zh-Hant:孫;中山;文,逸仙;;;;
N;ALTID=1;PHONETIC=jyut;
   SCRIPT=Latn;LANGUAGE=yue:syun1;zung1saan1;man4,jat6sin1;;;;
```

4.7. PROP-ID

Parameter name: PROP-ID

Purpose: Identifies a property among all its siblings of the same property name.

Description: This parameter uniquely identifies a property among all of its siblings with the same name within a vCard. A valid PROP-ID value must be of 1 and a maximum of 255 octets in size, and it **MUST** only contain the ASCII alphanumeric characters ("A-Za-z0-9"), hyphen (-), and underscore ("_"). The identifier's only purpose is to uniquely identify siblings; its value has no other meaning. If an application makes use of PROP-ID, it **SHOULD** assign a unique identifier to each sibling property of the same name within their embedding component. The same identifier **MAY** be used for properties of a different name, and it **MAY** also be assigned to a same-named property that is not a sibling.

Resolving duplicate identifier conflicts is specific to the application. Similarly, handling properties where some but not all siblings have a PROP-ID assigned is application-specific.

Format definition:

```
prop-id-param  = "PROP-ID" "=" 1*255(ALPHA / DIGIT / "-"/ "_")
```

Example(s):

```
PHOTO;PROP-ID=p827:data:image/jpeg;base64,MIICajCCAdOgAwIBAg
        <...remainder of base64-encoded data...>
```

4.8. SCRIPT

Parameter name: SCRIPT

Purpose: Defines the script that a property value is written in.

Description: This parameter allows defining a script for a property value without also defining a language as the LANGUAGE parameter would. The value **MUST** be a script subtag as defined in Section 2.2.3 of [RFC5646]. This specification makes use of the SCRIPT parameter in combination with the PHONETIC (Section 4.6) parameter.

Format definition:

```
script-param = 4ALPHA
```

Example(s):

```
SCRIPT=Latn
```

4.9. SERVICE-TYPE

Parameter name: SERVICE-TYPE

Purpose: Defines the online service name associated with a messaging or social media profile.

Description: This parameter **MAY** be specified on an IMPP or a SOCIALPROFILE property to name the online service associated with that property value. Its value is case-sensitive; its letter cases **MUST** be preserved.

Several vCard address book implementations currently use an experimental X-SERVICE-TYPE parameter. This specification provides an IANA-registered parameter for the same purpose.

Format definition:

```
service-type-param    = "SERVICE-TYPE" "=" param-value
```

Example(s):

```
SOCIALPROFILE;SERVICE-TYPE=Mastodon:https://example.com/@foo
```

4.10. USERNAME

Parameter name: USERNAME

Purpose: Defines a username such as the user of a messaging or social media service.

Description: This parameter **MAY** be specified on an IMPP or a SOCIALPROFILE property to name the user with that property value. Its value is case-sensitive; its letter cases **MUST** be preserved. The IMPP or SOCIALPROFILE value type **MUST** be URI.

Format definition:

```
username-param    = "USERNAME" "=" param-value
```

Example(s):

```
SOCIALPROFILE;USERNAME="The Foo":https://example.com/@foo
```

5. New Values

5.1. Billing Address Type Value

Value: billing

Purpose: Indicates using this address for billing, e.g., to send invoices to.

Conformance: This value can be used with the TYPE parameter applied on the ADR property.

Example(s):

```
ADR;TYPE=billing:;;123 Main Street;Any Town;CA;91921-1234;U.S.A.
```

5.2. Delivery Address Type Value

Value: delivery

Purpose: Indicates using this address for delivery, e.g., to send packages to.

Conformance: This value can be used with the TYPE parameter applied on the ADR property.

Example(s):

```
ADR;TYPE=delivery:;;123 Main Street;Any Town;CA;91921-1234;U.S.A.
```

6. Security Considerations

This specification extends "vCard Format Specification" [RFC6350]. The same security considerations as outlined in Section 9 of [RFC6350] apply.

7. IANA Considerations

7.1. Changes to the vCard Properties Registry

7.1.1. New vCard Property Definitions

IANA has added the following entries to the "vCard Properties" registry, as defined in Section 10.3.1 of [RFC6350].

Namespace	Property	Reference
	CREATED	RFC 9554, Section 3.1
	GRAMGENDER	RFC 9554, Section 3.2
	LANGUAGE	RFC 9554, Section 3.3
	PRONOUNS	RFC 9554, Section 3.4
	SOCIALPROFILE	RFC 9554, Section 3.5

Table 1: New vCard Properties

7.1.2. Updated vCard Properties

IANA has added Section 2.1 of this document as a reference for the ADR property and Section 2.2 of this document as a reference for the N property in the "vCard Properties" registry.

7.2. Changes to the vCard Parameters Registry

IANA has added the following entries to the "vCard Parameters" registry, as defined in Section 10.3.2 of [RFC6350].

Namespace	Parameter	Reference
	AUTHOR	RFC 9554, Section 4.1
	AUTHOR-NAME	RFC 9554, Section 4.2
	CREATED	RFC 9554, Section 4.3
	DERIVED	RFC 9554, Section 4.4
	LABEL	[RFC6350], Section 6.3.1 and RFC 9554, Section 4.5
	PHONETIC	RFC 9554, Section 4.6
	PROP-ID	RFC 9554, Section 4.7

Namespace	Parameter	Reference
	SCRIPT	RFC 9554, Section 4.8
	SERVICE-TYPE	RFC 9554, Section 4.9
	USERNAME	RFC 9554, Section 4.10

Table 2: New vCard Parameters

7.3. Changes to the vCard Property Values Registry

IANA has added the following entries to the "vCard Property Values" registry, as defined in Section 10.3.4 of [RFC6350].

Property	Value	Reference
GRAMGENDER	animate	RFC 9554, Section 3.2
GRAMGENDER	common	RFC 9554, Section 3.2
GRAMGENDER	feminine	RFC 9554, Section 3.2
GRAMGENDER	inanimate	RFC 9554, Section 3.2
GRAMGENDER	masculine	RFC 9554, Section 3.2
GRAMGENDER	neuter	RFC 9554, Section 3.2

Table 3: New vCard Property Values

7.4. Changes to the vCard Parameter Values Registry

IANA has added the following entries to the "vCard Parameter Values" registry, as defined in Section 10.3.4 of [RFC6350].

Property	Parameter	Value	Reference
ADR	TYPE	billing	RFC 9554, Section 5.1
ADR	TYPE	delivery	RFC 9554, Section 5.2
ADR, N	PHONETIC	ipa	RFC 9554, Section 4.6
ADR, N	PHONETIC	jyut	RFC 9554, Section 4.6
ADR, N	PHONETIC	piny	RFC 9554, Section 4.6

Property	Parameter	Value	Reference
ADR, N	PHONETIC	script	RFC 9554, Section 4.6

Table 4: New vCard Property and Parameter 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>.

[RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[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>.

[RFC5646]   Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

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

[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, March 2024, <https://www.rfc-editor.org/info/rfc9553>.

[RFC9555]   Loffredo, M. and R. Stepanek, "JSContact: Converting from and to vCard", RFC 9555, DOI 10.17487/RFC9555, March 2024, <https://www.rfc-editor.org/info/rfc9555>.

9. Informative References

[CALCONNECT-VOBJECT]   Tse, R., Tam, P., and M. Douglass, "vObject Internationalization", Work in Progress, Internet-Draft, draft-calconnect-vobject-i18n-00, 7 June 2018, <https://datatracker.ietf.org/doc/html/draft-calconnect-vobject-i18n-00>.

[CLDRPersonName]   Davis, M., Edberg, P., Gillam, R., Kolisnychenko, A., McKenna, M., and
                    other CLDR committee members, "Unicode Locale Data Markup Language
                    (LDML) Part 8: Person Names", Unicode Technical Standard #35, Version 44.1,
                    July 2023, <https://www.unicode.org/reports/tr35/tr35-personNames.html>.

          [IPA]   IPA, "International Phonetic Alphabet", <https://
                    www.internationalphoneticalphabet.org/>.

Acknowledgements

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