

Workgroup: Calendaring Extensions
Internet-Draft:
draft-ietf-calext-vcard-jscontact-
extensions-10
Updates: [6350](#) (if approved)
Published: 31 August 2023
Intended Status: Standards Track
Expires: 3 March 2024
Authors: R. Stepanek M. Loffredo
 Fastmail IIT-CNR

vCard Format Extension 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).

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 3 March 2024.

Copyright Notice

Copyright (c) 2023 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. [Introduction](#)
 - 1.1. [Notational Conventions](#)
 - 1.2. [ABNF Notations](#)
2. [Updated Properties](#)
 - 2.1. [ADR](#)
 - 2.2. [N](#)
3. [New Properties](#)
 - 3.1. [CREATED](#)
 - 3.2. [GRAMGENDER](#)
 - 3.3. [LANGUAGE](#)
 - 3.4. [PRONOUNS](#)
 - 3.5. [SOCIALPROFILE](#)
4. [New Parameters](#)
 - 4.1. [AUTHOR](#)
 - 4.2. [AUTHOR-NAME](#)
 - 4.3. [CREATED](#)
 - 4.4. [DERIVED](#)
 - 4.5. [LABEL](#)
 - 4.6. [PHONETIC](#)
 - 4.7. [PROP-ID](#)
 - 4.8. [SCRIPT](#)
 - 4.9. [SERVICE-TYPE](#)
 - 4.10. [USERNAME](#)
5. [New Values](#)
 - 5.1. [Billing Address Type Value](#)
 - 5.2. [Delivery Address Type Value](#)
6. [Security Considerations](#)
7. [IANA Considerations](#)
 - 7.1. [Changes to the "vCard Properties" registry](#)
 - 7.1.1. [New property definitions](#)
 - 7.1.2. [Updated vCard properties](#)

- [7.2. Changes to the "vCard Parameters" registry](#)
- [7.3. Changes to the "vCard Property Values" registry](#)
- [7.4. Changes to the "vCard Parameter Values" registry](#)
- [8. Acknowledgements](#)
- [9. References](#)
 - [9.1. Normative References](#)
- [10. Informative References](#)
- [Authors' Addresses](#)

1. Introduction

The JSContact [[I-D.ietf-calexext-jscontact](#)] 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 [[I-D.ietf-calexext-jscontact-vcards](#)] with the goal of not losing any semantics during conversion. To do so, 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 either are defined in [[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 that currently typically are 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 **SHOULD** ignore the street component during read if the ADR property value contains any of the new components.

The following change is made to the first paragraph in the "Special Notes" section, originally specified in [Section 6.3.1](#) of [[RFC6350](#)]. All remaining paragraphs of that section in the original specification still apply.

Special notes: 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;
- the extended address (e.g., apartment or suite number);
- the street address;
- the locality (e.g., city);
- the region (e.g., state or province);
- the postal code;
- the country name (full name in the language specified in [Section 5.1](#) of [[RFC6350](#)]);
- the room or suite number or identifier
- the apartment number, extension designation or box number.
- the building floor or level;
- the street number;
- the street name;
- the building, tower, condominium;
- the block name or number;
- the subdistrict;
- the district;
- the landmark or another publicly known prominent feature that can substitute the street name and number, e.g., "White House", "Taj Mahal";
- the cardinal direction or quadrant, e.g., "North"

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

ABNF

```

ADR-value = ADR-component-pobox ";" ADR-component-ext ";"
            ADR-component-street ";" ADR-component-locality ";"
            ADR-component-region ";" ADR-component-code ";"
            ADR-component-country ";"
            ; above components are defined in RFC 6350, section 6.3.1
            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 to the "Example" section, originally specified in [Section 6.2.2](#) of [\[RFC6350\]](#).

Example On this example, the post office box and the extended address components are, as shown, street number and name added both as separate components, as well as combined in the street component for backwards-compatibility.

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. Doing so, this also facilitates formatting N property values using the [Unicode CLDR Person Name](#) [\[CLDRPersonName\]](#) formatting standard.

One new component is for secondary surnames, as in some cultures, such secondary surname kinds are used to indicate the paternal and maternal family names or generational names indicating father,

grandfather. Another new component indicates a generation ("II", "XVI") or parental relation ("Jr.", "Sr.").

Currently, implementations typically place secondary surnames in the family name components, 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 backward-compatible component if an equal value is set in the according new component. 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 in the "Special Notes" section, originally specified in [Section 6.2.2](#) of [\[RFC6350\]](#). All remaining text of this and the following paragraphs of that section in the original specification still apply.

Special notes: 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 to the "ABNF" section, originally specified in [Section 6.2.2](#) of [\[RFC6350\]](#).

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

The following change is made to the "Example" section, originally specified in [Section 6.2.2](#) of [\[RFC6350\]](#).

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

Format definition: This property is defined by the following notation:

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

3.2. GRAMGENDER

Property name: GRAMGENDER

Purpose: This property defines which grammatical gender to use in salutations and other grammatical constructs.

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

Cardinality: *

Property parameters:

LANG

Description: This property defines the grammatical gender that the contact prefers to be addressed by or referred to 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 LANG parameter.

Format definition: This property is defined by the following notation:

```
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: GRAMGENDER:neuter

3.3. LANGUAGE

Property name: LANGUAGE

Purpose: This property defines the default language that human-readable text values in this vCard should be assumed 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 in which property values of type TEXT shall be assumed to be written 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: This property is defined by the following notation:

```
language-prop          = "LANGUAGE" any-param ":" Language-Tag  
                        ; Language-Tag is defined in RFC6350, Section 4.
```

Example: LANGUAGE:de-AT

3.4. PRONOUNS

Property name: PRONOUNS

Purpose: This property 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: LANG, PREF, TYPE

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:

This property is defined by the following notation:

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

3.5. SOCIALPROFILE

Property name: SOCIALPROFILE

Purpose: To specify 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, 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: This property is defined by the following notation:

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

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

```
socialpr-value   = URI / text
```

Example (As IPR):
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: This parameter 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. Note that as an alternative or in addition to this parameter, the AUTHOR-NAME parameter allows naming an author as free-text value (see [Section 4.2](#)).

Format definition: = "AUTHOR" "=" DQUOTE URI DQUOTE

Example (As IPR): AUTHOR="mailto:john@example.com":This is some note.

4.2. AUTHOR-NAME

Parameter name: AUTHOR-NAME

Purpose: This parameter 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 would not be a valid param-value (see [Section 3.3](#) of [\[RFC6350\]](#)). Note that as an alternative or in addition to this parameter, the AUTHOR parameter allows identifying an author by URI (see [Section 4.1](#)).

Format definition: param = "AUTHOR-NAME" "=" param-value ; not empty

EXAMPLE: (AUTHOR-NAME=John Doe:This is some note.
NOTE;AUTHOR-NAME="_:133tHckr:_":A note by an unusual author name.

4.3. CREATED

Parameter name: CREATED

Purpose: This parameter 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: param = "CREATED" "=" param-value ;
; a valid **TIMESTAMP** of [Section 4.3.5](#) of [\[RFC6350\]](#)

EXAMPLE: (CREATED=20221122T151823Z:This is some note.

4.4. DERIVED

Parameter name: DERIVED

Purpose: This parameter 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 an 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.

For 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: 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: This parameter is 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 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.

Formatted definition:="LABEL" "=" param-value

Example(s): The LABEL parameter as illustrated 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: This parameter 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 a same-named other property in the same vCard. Exemplary uses are to define how pronounce a Japanese name, or for romanization of Mandarin or Cantonese name and address components.

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

*ipa: denotes the [International Phonetic Alphabet](#) [[IPA](#)].

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

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

*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 parameter ([Section 5.4](#) of [RFC6350]) **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=" phonetic-value`

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

Example(s):
`N;ALTID=1;PHONETIC=jyut;
SCRIPT=Latn;LANGUAGE=yue:syun1;zung1saan1;man4,jat6sin1;;;`

4.7. PROP-ID

Parameter name: PROP-ID

Purpose: This parameter 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 only has the purpose 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 is assigned, is application-specific.

Format definition:="PROP-ID" "=" 1*255(ALPHA / DIGIT / "-" / "_")

Example(s): PROP-ID=p827:
<...remainder of base64-encoded data...>

4.8. SCRIPT

Parameter name: SCRIPT

Purpose: This parameter defines the script in which 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:4ALPHA

Example(s): Latin

4.9. SERVICE-TYPE

Parameter name: SERVICE-TYPE

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

Description: This parameter **MAY** be specified on a IMPP or 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: To define a username, such as the user of a messaging or social media service.

Description: This parameter **MAY** be specified on a IMPP or 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-value

Example(s): SOCIALPROFILE;USERNAME="The Foo":https://example.com/@foo

5. New Values

5.1. Billing Address Type Value

Value: billing

Purpose: This indicates to use 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): TYPE=billing;;;123 Main Street;Any Town;CA;91921-1234;U.S.A.

5.2. Delivery Address Type Value

Value: delivery

Purpose: This indicates to use 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): TYPE=delivery;;;123 Main Street;Any Town;CA;91921-1234;U.S.A.

6. Security Considerations

This specification extends the vCard Format Specification. 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 property definitions

IANA is requested to add the following entries to the "vCard Properties" registry, defined in Section 10.3.1. of [[RFC6350](#)].

Namespace	Property	Reference
	CREATED	This document, Section 3.1
	GRAMGENDER	This document, Section 3.2
	LANGUAGE	This document, Section 3.3
	PRONOUNS	This document, Section 3.4
	SOCIALPROFILE	This document, Section 3.5

Table 1: New vCard Properties

7.1.2. Updated vCard properties

IANA is requested to add [Section 2.1](#) of this document as reference for the ADR property.

IANA is requested to add [Section 2.2](#) of this document as reference for the N property.

7.2. Changes to the "vCard Parameters" registry

IANA is requested to add the following entries to the "vCard Parameters" registry, defined in Section 10.3.2. of [[RFC6350](#)].

Namespace	Parameter	Reference
	AUTHOR	This document, Section 4.1
	AUTHOR-NAME	This document, Section 4.2
	CREATED	This document, Section 4.3
	DERIVED	This document, Section 4.4
	LABEL	Section 6.3.1 of [RFC6350] and this document, Section 4.5
	PHONETIC	This document, Section 4.6
	PROP-ID	This document, Section 4.7
	SCRIPT	This document, Section 4.8
	SERVICE-TYPE	This document, Section 4.9
	USERNAME	This document, Section 4.10

Table 2: New vCard Parameters

7.3. Changes to the "vCard Property Values" registry

IANA is requested to add the following entries to the "vCard Property Values" registry, defined in Section 10.3.4. of [RFC6350].

Property	Value	Reference
GRAMGENDER	animate	This document, Section 3.2
GRAMGENDER	common	This document, Section 3.2
GRAMGENDER	feminine	This document, Section 3.2
GRAMGENDER	inanimate	This document, Section 3.2
GRAMGENDER	masculine	This document, Section 3.2
GRAMGENDER	neuter	This document, Section 3.2

Table 3: New vCard Property Values

7.4. Changes to the "vCard Parameter Values" registry

IANA is requested to add the following entries to the "vCard Parameter Values" registry, defined in Section 10.3.4. of [RFC6350].

Property	Parameter	Value	Reference
ADR	TYPE	billing	This document, Section 5.1
ADR	TYPE	delivery	This document, Section 5.2
ADR	TYPE	delivery	This document, Section 5.2
ADR, N	PHONETIC	ipa	This document, Section 4.6
ADR, N	PHONETIC	jyut	This document, Section 4.6
ADR, N	PHONETIC	piny	This document, Section 4.6
ADR, N	PHONETIC	script	This document, Section 4.6

Table 4: New vCard Property Values

8. Acknowledgements

The definition and examples of the [PHONETIC](#) ([Section 4.6](#)) and [SCRIPT](#) ([Section 4.8](#)) parameters are based on the initial version of [[I-D.calconnect-vobject-i18n](#)].

9. References

9.1. Normative References

[[I-D.ietf-calext-jscontact](#)] Stepanek, R. and M. Loffredo, "JSContact: A JSON representation of contact data", Work in Progress, Internet-Draft, draft-ietf-calext-jscontact-13, 24 July 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-calext-jscontact-13>>.

[I-D.ietf-calext-jscontact-vcard]

Loffredo, M. and R. Stepanek,
"JSContact: Converting from and to vCard", Work in
Progress, Internet-Draft, draft-ietf-calext-jscontact-
vcard-11, 24 July 2023, <[https://datatracker.ietf.org/
doc/html/draft-ietf-calext-jscontact-vcard-11](https://datatracker.ietf.org/doc/html/draft-ietf-calext-jscontact-vcard-11)>.

[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](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](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](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](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](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>>.

10. Informative References

[CLDRPersonName] Davis, M., Edberg, P., Gillam, R., Kolisnychenko,
A., McKenna, M., and others, "Technical Standard #35:
Unicode Locale Data Markup Language (LDML) Part 8: Person
Names, Version 43.1", July 2023, <[https://
www.unicode.org/reports/tr35/tr35-personNames.html](https://www.unicode.org/reports/tr35/tr35-personNames.html)>.

[I-D.calconnect-vobject-i18n] Tse, R. H., 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>>.

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

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