vCard

The Electronic Business Card

Version 2.1

A ***versit*** Consortium Specification

September 18, 1996

Trademarks

versit, the ***versit*** logo, versitcard, vCard, and vCalendar are trademarks of Apple Computer, Inc., AT&T Corp., International Business Machines Corp., and Siemens.

Apple, is a trademarks of Apple Computer, Inc. registered in the U.S. and other countries.

AT&T and ATTMail are registered trademarks of AT&T Corp.

IBM, IBM Mail, and OS/2 are registered trademarks of International Business Machines Corporation.

America Online is a registered trademark of America Online, Inc.

CompuServe, CompuServe Information Services are registered trademarks of Compuserve Incorporated.

MCIMail is a registered trademark of MCI Communications Corporation.

Microsoft is a registered trademark, and Microsoft Windows is a trademark of Microsoft Corporation.

Prodigy is a registered trademark of Prodigy Services Company.

Unicode is a registered trademark of Unicode, Inc.

Contributors

Roland Alden

Greg Ames, Ames & Associates

Masanari Arai, Puma Technologies

Stephen W. Bartlett

Donal Carroll

Liang-Jye Chang, Starfish Software

Frank Dawson, IBM Corporation

Ken Dobson, IntelliLink Inc.

Scott Feldstein, Nimble Software, Inc.

Anik Ganguly, OnTime/Division of FTP Software.

Beijing Goo, Microsoft

Arvind K. Goyal, Lotus Development Corporation

Gary Hand, IBM Corporation

Tim Howes, Netscape Communications Corporation

Mark Joseph, Attachmate Corporation

Kerry Kelly, Now Software, Inc.

Phac Letuan, Apple Computer, Inc.

Pat Megowan, Counterpoint Sytems Foundry Inc.

Tohri Mori, IBM Japan/Salutation Consortium

Ravi Pandya, NetManage, Inc.

Geoff Ralston, Four11 Corporation

Steven Rummel, Lucent Technologies

Michael Santullo, Four11 Corporation

Vinod Seraphin, Lotus Development Corporation

Dexter Seely, Corex Technologies, Inc.

Vlad Shmunis, Ring Zero Systems Inc.

Dean Stevens, Now Software, Inc.

Michelle Watkins, Netscape Communications Corporation

Horst Widlewski, Siemens

Reference Information

The cited references contain provisions which, through reference in this specification, constitute provisions of this specification. At the time of publication, the indicated versions in the following references were valid. Parties to agreements based on this specification are encouraged to research the possibility of revised standards.

- ANSI X3.4-1977, *Code for Information Interchange*, American National Standards Institute, 1977.
- CCITT (ITU) Recommendation E.163, *Numbering Plan for The International Telephone Service*, CCITT Blue Book, Fascicle II.2, pp. 128-134, November, 1988.
- CCITT (ITU) Recommendation G.721, *32 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)*, CCITT Red Book, Fascicle III.4, November, 1988.
- CCITT (ITU) Recommendation X.121, *International Numbering Plan for Public Data Networks*, CCITT Blue Book, Fascicle VIII.3, pp. 317-332, November, 1988.
- CCITT (ITU) Recommendations X.500-X.521, *Data Communication Networks: Directory*, CCITT Blue Book, Fascicle VIII.8, November, 1988.
- CCITT Recommendation X.520, *The Directory—Selected Attribute Types,* 1988.
- CCITT Recommendation X.521, *The Directory—Selected Object Classes,* 1988.
- IETF RFC 1738, *Universal Resource Locator*, December 1994.
- IETF Network Working Group RFC 1766, *Tags for the Identification of Languages,* March 1995.
- IETF Network Working Group Draft, *A MIME Content-Type for Directory Information,* January 1996. Available from the University of Michigan, 535 W. William St., Ann Arbor, MI 48103-4943, FTP://ds.internic.net/Internet-Drafts/draft-ietf-asid-mime-direct-01.txt.
- IETF Network Working Group Draft, *An Application/Directory MIME Content-Type Electronic Business Card Profile,* May 1996. Available FTP://ds.internic.net/Internet-Drafts/draft-ietf-asid-mime-vcard-00.txt.
- IETF Network Working Group Draft, *UTF-8, A Transformation Format of UNICODE and ISO 10646,* July 1996. Available from FTP://ds.internic.net/Internet-Drafts/draft-yergeau-utf8-01.txt.
- ISO 639, *Code for The Representation of names of languages*, International Organization for Standardization, April, 1988.

- ISO 3166, *Codes for The Representation of names of countries*, International Organization for Standardization, December, 1993.
- ISO 8601, *Data elements and interchange formats—Information interchange—Representation of dates and times*, International Organization for Standardization, June, 1988.
- ISO 8601, *Technical Corrigendum 1, Data elements and interchange formats—Information interchange—Representation of dates and times*, International Organization for Standardization, May, 1991.
- ISO 8859-1, *Information Processing—8-Bit single-byte coded graphic character sets—Part 1: Latin Alphabet No. 1,* International Organization for Standardization, February, 1987.
- ISO 9070, *Information Processing—SGML support facilities—Registration Procedures for Public Text Owner Identifiers,* 1990-02-01.
- ISO/IEC 9070, *Information Technology—SGML Support Facilities—Registration Procedures for Public Text Owner Identifiers,* Second Edition, International Organization for Standardization, April, 1991.
- ISO/IEC 11180, *Postal addressing*, International Organization for Standardization, 1993.
- Apple's Representation of a Canonical Static DeviceID in *The Telephony Suite,* version 1.0, Apple Computer, Inc., 1993.
- Microsoft TAPI in *Microsoft Windows 3.1 Telephony Programmers' Guide,* version 1.0, Microsoft Corporation, 1993.
- RFC1521, *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies,* Network Working Group, September, 1993.
- *The Unicode Standard, Version 1.1: Version 1.0, Volume 1* (ISBN 0-201-56788-1), version 1.0, volume 2 (ISBN 0-20-60845-6) and *Unicode Technical Report #4, The Unicode Standard*, version 1.1, The Unicode Consortium, October, 1991. Both references to be published by Addison-Wesley.

versit Update

versit is a multivendor development initiative of the communication and computer industries, founded by Apple, AT&T, IBM and Siemens. The ***versit*** parties believe that great potential exists in improving the nature of communications in the business world—permitting companies to better manage their quality, productivity, customer satisfaction and cost of operations, while expanding the market opportunities for a variety of product and service vendors. ***versit*** parties will jointly define and support open specifications that facilitate and promote the interoperability of advanced personal information and communication devices, networks and services.

The ***versit*** vision is to enable diverse communication and computing devices, applications and services from competing vendors to interoperate in all environments. Through developing a series of specifications for interoperability among diverse communications and computing devices, applications, networks and services, ***versit*** 's vision will become a reality.

versit 's primary development areas are in:

- Personal Data Interchange (PDI)
- Computer Telephone Integration (CTI)
- Conferencing and Messaging (C&M)
- Wired and Wireless connectivity

versit specifications are directed at both the decision makers and the implementation teams of:

- Equipment Manufacturers
- Independent Software Vendors
- Information Service Providers
- Online Service Providers
- Software Houses
- Users

versit specifications are made available to any interested party. In turn, ***versit*** encourages the support of our goals by soliciting feedback on ***versit*** specifications.

All comments relating to ***versit*** or the material within this specification should be submitted to:

versit
(800) 803-6240
+1 (201) 327-2803 (Outside USA)
pdi@versit.com
http://www.versit.com/pdi

Contents

Section 1 : Introduction

Personal Data Interchange (**PDI**) occurs every time two or more individuals communicate, in either a business or personal context, face-to-face, or across space and time. Such interchanges frequently include the exchange of informal information, such as business cards, telephone numbers, addresses, dates and times of appointments, etc. Augmenting PDI with electronics and telecommunications can help ensure that information is quickly and reliably communicated, stored, organized and easily located when needed.

Personal information, by nature, is complex and diverse. Currently, proprietary standards exist to structure some types of PDI information, but no single, open specification comprehensively addresses the needs of collecting and communicating PDI information across many common communication channels such as telephones, voice-mail, e-mail, and face-to-face meetings. ***versit*** is developing a comprehensive family of PDI technologies based on open specifications and interoperability agreements to help meet this technology need.

1.1 Overview

This specification defines a format for an electronic business card, or ***vCard***. The format is suitable as an interchange format between applications or systems. The format is defined independent of the particular method used to transport it. The transport for this exchange might be a file system, point-to-point asynchronous communication, wired-network transport, or some form of unwired transport.

A vCard is a data stream consisting of one or more vCard objects. The individual vCard definitions can be identified and parsed within the datastream. The vCard data stream may exist as a persistent form in a file system, document management system, network connection between two network endpoints, or in any other digital transport that has an abstraction of a stream of bytes.

Conceptually, a *vCard Writer* creates vCard data streams and a *vCard Reader* interprets vCard data streams. The vCard Reader and Writer may be implemented as a single application or as separate applications. It is not the intent of this specification to define the implementation of these processes beyond some fundamental capabilities related to the format of the vCard data stream and a common set of conformance requirements .

This specification provides for a clear-text encoding that is intended to be based on the syntax used by the MIME specification (RFC 1521).

The encoding of this specification can be used in environments which are constrained to 7-bit transfer encodings, short line lengths, and low bandwidth. In addition, the encoding is simple in order to facilitate the implementation of reader and writer applications on small platforms, such as Personal Digital Assistants (PDA), cellular telephones, or alphanumeric pagers.

1.2 Scope

The vCard is intended to be used for exchanging information about people and resources. In today's business environment, this information is typically exchanged on business cards. It is appropriate, then that this specification define this information in terms of a paradigm based on an electronic business card object.

The ultimate destination for this information is often a collection of business cards, Rolodex® file, or electronic contact manager. Prior to the introduction of the vCard specification, users of such applications typically had to re-key the original information, often transcribing it from paper business cards. With the advent of the vCard specification, this information can be exchanged in an automated fashion.

The basis for the data types supported by this specification have their origin in openly defined, international standards and in additional capabilities based on enhancements suggested by the demonstration of the exchange of prototypical vCards using the Internet based World-Wide-Web, Infra-red data transport, and simultaneous voice and data (SVD) modems.

The "person" object defined by the CCITT X.500 Series Recommendation for Directory Services was the primary reference for the properties that are defined by this specification. Every attempt was made to make it possible to map the X.520/X.521 attributes and objects into and out of an instance of a vCard. The vCard specification has extended the capabilities that have been defined within the CCITT X.500 Series Recommendation to allow the exchange of additional information often recorded on business cards and electronic contact managers. For example, this specification provides support for exchanging graphic images representing company logos, photographs of individuals, geo-positioning information, and other extensions to properties defined by the X.500 Recommendation.

The specification of all date and time values are defined in terms of the ISO 8601 standard for representation of dates and times. ISO 8601 supersedes all other international standards defined at the time this specification was drafted.

The paradigm of an electronic business card is related to the concepts of an entry in a LAN/WAN directory or an electronic mail address book or distribution list. However, the requirements of the electronic business card go beyond the definitions of a "person" object found in either the CCITT X.500 Series Recommendation, network directory services, or electronic mail address book products. The vCard specification is needed to address the requirements for an interchange format for the "person" personal data type or object.

Personal data applications such as Personal Information Managers (PIM) often provide an import/export capability using Comma Separated Value (CSV) or Tab Delimited Files (TDF) formats. However, these solutions do not preserve the intent of the originating application. When a CSV and TDF format is used by a PIM, the meta-data or semantics of the originating object are only apparent to a similar version of the originating application. Exchange of data between such applications is another important application of an industry-standard specification for an electronic business card interchange format, such as the vCard specification.

1.3 Contents

This specification is separated into eight sections:

- "Section 1 : Introduction" introduces PDI and the vCard specification with an overview, scope statement and section on definitions and abbreviations.
- "Section 2 : vCard Specification" defines the semantics and syntax for the vCard.
- "Section 3 : Internet Recommendations" specifies a set of guidelines to facilitate the exchange of vCard objects over Internet protocols such as HTTP using HTML and SMTP using MIME.
- "Section 4 : UI Support Recommendations" specifies a set of guidelines to facilitate the exchange of vCard objects at the desktop user interface using the file system, clipboard and drag/drop capabilities of the operating system.
- "Section 5 : Conformance" defines minimum conformance requirements to consider while developing support for this vCard specification.

1.4 Definitions and Abbreviations

Definitions and abbreviations used within this specification follow.

Electronic Business Card: Also known as vCard.

FPI: Formal Public Identifier. A string expression that represents a public identifier for an object. FPI syntax is defined by ISO 9070.

GUID: Globally Unique IDentifier

Internet: A WAN connecting thousands of disparate networks in industry, education, government, and research. The Internet uses TCP/IP as the standard for transmitting information.

ISO: Organization for International Standardization; a worldwide federation of national standards bodies (ISO Member bodies).

MIME: Multipurpose Internet Mail Extensions, as defined in RFC1521.

PDA: Personal Digital Assistant computing device

PDI: Personal Data Interchange, a collaborative application area which involves the communication of data between people who have a business or personal relationship, but do not necessarily share a common computing infrastructure.

PIM: Personal Information Manager

RFC#### documents: Internet “Request For Comment” documents (i.e., RFC822, RFC1521, etc.).

URL: Uniform Resource Locator; a string expression that can represent any resource on the Internet or local system. RFC 1738 defines the syntax for an URL.

UTC: Universal Time Coordinated; also known as UCT, for Universal Coordinated Time.

vCard: The generic term for an electronic, virtual information card that can be transferred between computers, PDAs, or other electronic devices through telephone lines, or e-mail networks, or infrared links. How, when, why, and where vCard are used depends on the applications developed utilizing a vCard.

versitcard: a vCard.

WAN: Wide-Area Network

Section 2 : vCard Specificiation

This section defines the semantics and syntax for the vCard.

A vCard is a collection of one or more properties. A property is a uniquely named value. A set of properties can be grouped within a vCard. For example, the properties for a telephone number and comment can be grouped in order to preserve the coupling of the annotation with the telephone number. In addition to property groupings, a vCard can include other, nested vCard objects. This allows for the recording of information about a secondary person or object associated with a given person or object. Additionally, this allows for the specification of a distribution list or work group of multiple vCard objects.

2.1 Encoding Characteristics

The following are encoding characteristics specific to this specification.

2.1.1 vCard Object

A vCard data stream may include one or more vCard objects. An individual vCard object is identified within a data stream by the appearance of the Begin vCard Delimiter:

```
BEGIN:VCARD
```

The sentinel string must appear as the first characters in the data stream or the first characters on a line.

The vCard object is terminated with either the logical end of the data stream or the appearance of the End vCard Delimiter as the first character on a line:

```
END:VCARD
```

2.1.2 Property

A *property* is the definition of an individual attribute describing the vCard. A property takes the following form:

PropertyName [‘;‘ PropertyParameters] ’:‘ PropertyValue

as shown in the following example:

```
TEL;HOME:+1-919-555-1234
```

A property takes the form of one or more lines of text. The specification of property names and property parameters is case insensitive. The property name can be one of a set of pre-defined strings. The property name, along with an optional grouping label, must appear as the first characters on a line. In the previous example, “TEL” is the name of the Telephone Number property. Property values are specified as strings. In the previous example, “+1-919-555-1234” is the formatted value for the Telephone Number property.

A property value can be further qualified with a property parameter expression. Property parameter expressions are delimited from the property name with a Semi-colon character (ASCII 59). A Semi-colon in a property parameter value must be escaped with a Backslash character (ASCII 92). The property parameter expressions are specified as either a name=value or a value string. The value string can be specified alone in those cases where the value is unambiguous. For example a complete property parameter specification might be:

```
NOTE;ENCODING=QUOTED-PRINTABLE:Don't remember to order Girl=
  Scout cookies from Stacey today!  //Full property parameter expression
```

A valid short version of the same property parameter specification might be:

```
NOTE;QUOTED-PRINTABLE:Don't remember to order Girl=
  Scout cookies from Stacey today!  //Full property parameter expression
```

2.1.3 Delimiters

Individual lines within the vCard data stream are delimited by the (RFC 822) line break, which is a CRLF sequence (ASCII decimal 13, followed by ASCII decimal 10). Long lines of text can be split into a multiple-line representation using the RFC 822 "folding" technique. That is, wherever there may be linear white space (NOT simply LWSP-chars), a CRLF immediately followed by at least one LWSP-char may instead be inserted. For example the line:

```
NOTE:This is a very long description that exists on a long line.
```

Can be represented as:

```
NOTE:This is a very long description

  that exists on a long line.
```

The process of moving from this folded multiple-line representation of a property definition to its single line representation is called "unfolding". Unfolding is accomplished by regarding CRLF immediately followed by a LWSP-char as equivalent to the LWSP-char.

It is recommended that folding be limited to higher-level syntactic breaks in structured components of the property definition.

A formatted text line break in a property value, must also be represented by a (RFC 822) line break, which is a CRLF sequence. However, since the CRLF sequence is used to delimit a line, property values with formatted line breaks (i.e., multiple lines) must be encoded using an alternate encoding of either Quoted-Printable or Base64, as defined in RFC 1521.

For example, in the Quoted-Printable encoding the multiple lines of formatted text are separated with a Quoted-Printable CRLF sequence of "=0D" followed by "=0A" followed by a Quoted-Printable softline break sequence of "=". Quoted-Printable lines of text must also be limited to less than 76 characters. The 76 characters does not include the CRLF (RFC 822) line break sequence. For example a multiple line LABEL property value of:

```
123 Winding Way
Any Town, CA 12345
USA
```

Would be represented in a Quoted-Printable encoding as:

```
LABEL;ENCODING=QUOTED-PRINTABLE:123 Winding Way=0D=0A=
  Any Town, CA 12345=0D=0A=
```

```
USA
```

Property parameter substrings are delimited by a field delimiter, specified by the Semi-colon character (ASCII decimal 59). A Semi-colon in a property parameter value must be escaped with a Backslash character (ASCII 92).

Compound property values are property values that also make use of the Semi-colon, field delimiter to separate positional components of the value. For example, the Name property is made up of the Family Name, Given Name, etc. components. A Semi-colon in a component of a compound property value must be escaped with a Backslash character (ASCII 92).

2.1.4 Grouping

There are two forms of grouping or collections supported within the vCard. A collection of vCard objects can be grouped and a collection of properties within an individual vCard can be grouped.

2.1.4.1 vCard Grouping

The vCard data stream can consist of multiple vCard objects. The vCard data stream can, sequentially, contain one or more vCard objects., In addition, the vCard data stream can contain a property whose value is a nested vCard. In both of these cases, each vCard object will be delimited by the vCard Delimiters. The vCard Reader conforming to this specification must be able to parse and process any of these combinations of vCard Groupings. The support for vCard Grouping is optional for a vCard Writer conforming to this specification.

2.1.4.2 Property Grouping

A *Property Grouping* is the definition of a method for specifying a collection of related properties within a vCard object. There is no requirement on a vCard reader that it preserve the property group name. However, the vCard reader is required to preserve the grouping of the properties.

The Property Grouping is identified by a character string prefix to the property name; separated by the Period character (ASCII decimal 46).

The grouping of a comment property with a telephone property is shown in the following example:

```
A.TEL;HOME:+1-213-555-1234
A.NOTE:This is my vacation home.
```

The vCard Reader conforming to this specification must be able to parse and process the property grouping. The support for Property Grouping is optional for a vCard Writer conforming to this specification.

2.1.5 Encodings

The default encoding for the vCard object is 7-Bit. The default encoding can be overridden for an individual property value by using the “ENCODING” property parameter. This parameter value can be either “BASE64”, “QUOTED-PRINTABLE”, or “8BIT”. This parameter may be used on any property.

Some transports (e.g., MIME based electronic mail) may also provide an encoding property at the transport wrapper level. This property can be used in these cases for transporting a vCard data stream that has been defined using a default encoding other than 7-bit (e.g., 8-bit).

2.1.6 Character Set

The default character set is ASCII. The default character set can be overridden for an individual property value by using the “CHARSET” property parameter. This property parameter may be used on any property. However, the use of this parameter on some properties may not make sense.

Any character set registered with the Internet Assigned Numbers Authority (IANA) can be specified by this property parameter. For example, ISO 8859-8 or the Latin/Hebrew character set is specified by:

```
ADR;CHARSET=ISO-8859-8:...
```

Some transports (e.g., MIME based electronic mail) may also provide a character set property at the transport wrapper level. This property can be used in these cases for transporting a vCard data stream that has been defined using a default character set other than ASCII (e.g., UTF-8).

2.1.7 Language

The default language is “en-US” (US English). The default language can be overridden for an individual property value by using the “LANGUAGE” property parameter. The values for this property are a string consistent with RFC 1766, Tags for the Identification of Languages. This property parameter may be used on any property. However, the use of this parameter on some properties, such as PHOTO, LOGO, SOUND, TEL, may not make sense. Canadian French would be specified by this parameter by the following:

```
ADR;LANGUAGE=fr-CA:...
```

2.1.8 Value Location

The default location of the property value is inline with the property. However, for some properties, such as those that specify multimedia values, it is efficient to organize the property value as a separate entity (e.g., a file out on the network). The property parameter “VALUE” can be specified to override the “INLINE” location of the property value. In the case of the vCard being transported within a MIME email message, the property value can be specified as being located in a separate MIME entity with the “Content-ID” value, or “CID” for short. In this case, the property value is the Content-ID for the MIME entity containing the property value. In addition, the property value can be specified as being located out on the network within some Internet resource with the “URL” value. In this case, the property value is the Uniform Resource Locator for the Internet resource containing the property value. This property parameter may be used on any property. However, the use of this parameter on some properties may not make sense; for example the Version, Time Zone, Comment, Unique

Identifier, properties . The following specifies a value not located inline with the vCard but out in the Internet:

```
PHOTO;VALUE=URL;TYPE=GIF:http://www.abc.com/dir_photos/my_photo.gif
SOUND;VALUE=CONTENT-ID:<jsmith.part3.960817T083000.xyzMail@host1.com
```

2.1.9 Binary Values

The vCard format supports inclusion of binary information, such as computer graphic images, digital audio, or video graphic images. The binary information may either be referenced with a Uniform Reference Locator (URL) or placed inline in the vCard as the value of a property. Inline binary information is included as a property value after being encoded into clear-text with a Base 64 (default) or Quoted-Printable encoding

2.2 Identification Properties

These property types are concerned with information associated with the identification and naming of the individual or resource associated with the vCard object.

2.2.1 Formatted Name

This property specifies the formatted name string associated with the vCard object. This is the way that the name is to be displayed. It can contain desired honorific prefixes, suffixes, titles, etc. For example, “Mr. John Q. Public, Jr.”, Dr. Ann Tyler, or Hon. Judge Blackwell. This property is based on the semantics of the X.520 Common Name attribute.

This property is identified by the property name **FN**. The following is an example of the Formatted Name property:

```
FN:Mr. John Q. Public, Esq.
```

Support for this property is optional for vCard Writers conforming to this specification.

2.2.2 Name

This property specifies a structured representation of the name of the person, place or thing associated with the vCard object.

This property is identified by the property name **N**. This property is defined to encapsulate the individual components of an object’s name. The property value consists of the components of the name specified as positional fields separated by the Field Delimiter character (ASCII decimal 59). The property value is a concatenation of the Family Name (first field), Given Name (second field), Additional Names (third field), Name Prefix (fourth field), and Name Suffix (fifth field) strings. The following is an example of the Name property for a person:

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

The following is an example of the Name property for a resource or place:

```
N:Veni, Vidi, Vici;The Restaurant.
```

Support for this property is mandatory for vCard Writers conforming to this specification. All vCard data streams should include this property to facilitate a common property for collating and sorting of vCard objects.

2.2.3 Photograph

This property specifies an image or photograph of the individual associated with the vCard.

The property is identified by the property name **PHOTO**. For example, the following syntax is an example of a referenced image file:

```
PHOTO;VALUE=URL:file:///jqpublic.gif
```

The following example is the syntax for including an inline GIF image file, using the Base 64 encoding:

```
PHOTO;ENCODING=BASE64;TYPE=GIF:
    R0lGODdhfgA4AOYAAAAAAK+vr62trVIxa6WlpZ+fnzEpCEpzlAha/0Kc74+PjyGM
    SuecKRhrtX9/fzExORBSjCEYCGtra2NjYyF7nDGE50JrhAg51qWtOTl7vee1MWu1
    50o5e3PO/3sxcwAx/4R7GBgQOcDAwFoAQt61hJyMGHuUSpRKIf8A/wAY54yMjHtz
...
```

Support for this property is optional for vCard Writers conforming to this specification.

2.2.3.1 Photo Format Type

This property parameter is provided to specify the graphics format for the Photo property value. The property parameter includes the following values:

Description	Property Parameter Value
TYPE=	
Indicates Graphics Interchange Format	GIF
Indicates ISO Computer Graphics Metafile	CGM
Indicates MS Windows Metafile	WMF
Indicates MS Windows Bitmap	BMP
Indicates IBM PM Metafile	MET
Indicates IBM PM Bitmap	PMB
Indicates MS Windows DIB	DIB
Indicates an Apple Picture format	PICT
Indicates a Tagged Image File Format	TIFF
Indicates Adobe PostScript format	PS
Indicates Adobe Page Description Format	PDF

Indicates ISO JPEG format	JPEG
Indicates ISO MPEG format	MPEG
Indicates ISO MPEG version 2 format	MPEG2
Indicates Intel AVI format	AVI
Indicates Apple QuickTime format	QTIME

2.2.4 Birthdate

This property specifies the date of birth of the individual associated with the vCard. The value for this property is a calendar date in a complete representation consistent with ISO 8601.

This property is identified by the property name **BDAY**. The property value is a string conforming to the ISO 8601 calendar date, complete representation, in either basic or extended format. The following example is in the basic format of ISO 8601:

```
BDAY:19950415
```

The following example is in the extended format of ISO 8601:

```
BDAY:1995-04-15
```

Support for this property is optional for vCard Writers conforming to this specification.

2.3 Delivery Addressing Properties

2.3.1 Delivery Address

This property specifies a structured representation of the physical delivery address for the vCard object. The property is made up of components that are based on the X.500 Post Office Box attribute, the X.520 Street Address geographical attribute, the X.520 Locality Name geographical attribute, the X.520 State or Province Name geographical attribute, the X.520 Postal Code attribute, and the X.520 Country Name geographical attribute.

This property is identified by the property name **ADR**. The property value consists of components of the address specified as positional fields separated by the Field Delimiter character (ASCII decimal 59). The property value is a concatenation of the Post Office Address (first field) Extended Address (second field), Street (third field), Locality (fourth field), Region (fifth field), Postal Code (six field), and Country (seventh field) strings. An example of this property follows:

```
ADR;DOM;HOME:P.O. Box 101;Suite 101;123 Main Street;Any Town;CA;91921-1234;
```

Support for this property is optional for vCard Writers conforming to this specification.

2.3.1.1 Delivery Address Type

This property parameter specifies the sub-types of physical delivery that is associated with the delivery address. For example, the label may need to be differentiated for Home, Work, Parcel, Postal, Domestic, and International physical delivery. One or more sub-types can be specified for a given delivery address.

The property parameter can have one or more of the following values:

Description	Property Parameter Value
TYPE=	
Indicates a domestic address	DOM
Indicates an international address (Default)	INTL
Indicates a postal delivery address (Default)	POSTAL
Indicates a parcel delivery address (Default)	PARCEL
Indicates a home delivery address	HOME
Indicates a work delivery address (Default)	WORK

The default property parameter is overridden to some other set of values by specifying one or more alternate values. For example, the default of a delivery for INTL, WORK, POSTAL and PARCEL can be reset to DOM, POSTAL, WORK and HOME in the following example:

```
ADR;DOM;WORK;HOME;POSTAL:P.O. Box 101;;;Any Town;CA;91921-1234;
```

2.3.2 Delivery Label

This property specifies the addressing label for physical delivery to the person/object associated with the vCard. The property is intended to include the information necessary to create a formatted delivery address label. Typical information includes the name, street address, possibly a Post Office or mail drop, city, state or province, zip or postal code. An international delivery label would also include the country name.

This property is based on the semantics of the X.520 Postal Address attribute. This specification has added semantics to those defined by the X.500 Series standard for differentiating Home, Work, Parcel, Postal, Domestic, and International delivery label types.

This property is identified by the property name **LABEL**. This property specifies the formatted delivery address label for the vCard object. An example of a domestic delivery label follows:

```
LABEL;DOM;POSTAL;ENCODING=QUOTED-PRINTABLE:P. O. Box 456=0D=0A=
123 Main Street=0D=0A=
Any Town, CA 91921-1234
```

An example of an international delivery label follows:

```
LABEL;INTL;PARCEL,ENCODING=QUOTED-PRINTABLE:Suite 101=0D=0A=
123 Main Street=0D=0A=
Any Town, CA 91921-1234=0D=0A=
U.S.A.
```

Support for this property is optional for vCard Writers conforming to this specification. A vCard Reader supporting this property and conforming to this specification should support a minimum of four lines of text for this property.

2.3.2.1 Delivery Label Type

This property parameter specifies the sub-types of physical delivery that is associated with the delivery label. For example, the label may need to be differentiated for Home, Work, Parcel, Postal, Domestic, and International physical delivery. One or more sub-types can be specified for a given delivery label.

The property parameter can have one or more of the following values:

Description	Property Parameter Value
TYPE=	
Indicates a domestic address	DOM
Indicates an international address (Default)	INTL
Indicates a postal delivery address (Default)	POSTAL
Indicates a parcel delivery address (Default)	PARCEL
Indicates a home delivery address	HOME
Indicates a work delivery address (Default)	WORK

The default property parameter is overridden to some other set of values by specifying one or more alternate values. For example, the default of a delivery for INTL, WORK, POSTAL and PARCEL can be reset to DOM and HOME in the following example:

```
LABEL;DOM;HOME,ENCODING=QUOTED-PRINTABLE:Suite 101=0D=0A=
123 Main Street=0D=0A=
Any Town, CA 91921-1234
```

2.4 Telecommunications Addressing Properties

These property types are concerned with information associated with the telecommunications addressing of the vCard object.

2.4.1 Telephone Number

This property specifies the canonical number string for a telephone number for telephony communication with the vCard object. The value of this property is specified in a canonical form in order to specify an unambiguous representation of the globally unique telephony endpoint. This property is based on the X.520 Telephone Number attribute.

The canonical form cannot be dialed without first being transformed by a dialing algorithm. The dialing algorithm combines the canonical number string with knowledge of the local dialing procedures, in effect at the time of call placement to produce actual dialing instructions. The actual dialing algorithm is outside the scope of this specification.

Two important canonical forms allowed by this specification are:

- Apple Computer's Representation of a Canonical Static DeviceID in *The Telephony Suite*, version 1.0,
- Microsoft TAPI in the Microsoft Windows 3.1 *Telephony Programmer's Guide*, version 1.0.

Software which creates this property can store a string in these allowed formats. Dialing software should be prepared to parse numbers from either of the supported formats; as neither format is considered to be technically costly to support.

This property is identified by the property name **TEL**. An example of this property follows:

```
TEL;PREF;WORK;MSG;FAX:+1-800-555-1234
```

Support for this property is optional for vCard Writers conforming to this specification.

2.4.1.1 Telephone Type

This property parameter specifies the sub-type of telephone that is associated with the telephone number (e.g., Home, Work, Cellular, Facsimile, Video, Modem, Message Service, or Preferred). One or more sub-type values can be specified for a given telephone number.

The property parameter can have one or more of the following values:

Description	Property Parameter Value
TYPE=	
Indicates preferred number	PREF
Indicates a work number	WORK
Indicates a home number	HOME
Indicates a voice number (Default)	VOICE
Indicates a facsimile number	FAX
Indicates a messaging service on the number	MSG
Indicates a cellular number	CELL
Indicates a pager number	PAGER
Indicates a bulletin board service number	BBS

Indicates a MODEM number	MODEM
Indicates a car-phone number	CAR
Indicates an ISDN number	ISDN
Indicates a video-phone number	VIDEO

The default property parameter is overridden to some other set of values by specifying one or more alternate values. For example, the default of a VOICE telephone number can be reset to a WORK and HOME, VOICE and FAX telephone number in the following example:

```
TEL;WORK;HOME;VOICE;FAX:+1-800-555-1234
```

2.4.2 Electronic Mail

This property specifies the address for electronic mail communication with the vCard object. The address is in the form of a specific addressing type. For example, the Internet mail address for John Public might be “John.Public@abc.com” or the CompuServe Information Service address might be “71234,5678”.This property is identified by the property name **EMAIL**.

An example of this property follows:

```
EMAIL;INTERNET:john.public@abc.com
```

Support for this property is optional for vCard Writers conforming to this specification.

2.4.2.1 Electronic Mail Type

This property parameter specifies the type of electronic mail address. The following are some example values for this property parameter:

Description	Property Parameter Value
TYPE=	
Indicates America On-Line	AOL
Indicates AppleLink	AppleLink
Indicates AT&T Mail	ATTMail
Indicates CompuServe Information Service	CIS
Indicates eWorld	eWorld
Indicates Internet SMTP (default)	INTERNET
Indicates IBM Mail	IBMMail
Indicates MCI Mail	MCIMail
Indicates PowerShare	POWERSHARE
Indicates Prodigy information service	PRODIGY

Indicates Telex number	TLX
Indicates X.400 service	X400

2.4.3 Mailer

This property parameter specifies the type of electronic mail software that is in use by the individual associated with the vCard object. This information may provide assistance to a correspondent regarding the type of data representation which can be used, and how they may be packaged. This property parameter is based on currently accepted practices within the Internet MIME community with the “X-Mailer” header field.

This property is identified by the property name **MAILER**. Support for this property is optional for vCard Writers conforming to this specification. An example of this property follows:

```
MAILER:ccMail 2.2
```

2.4.4 Geographical Properties

These property types are concerned with geographical positions or region information associated with the vCard object.

2.4.5 Time Zone

This property specifies information related to the standard time zone of the vCard object. The time zone is a string as specified in a manner consistent with ISO 8601. It is an offset from Coordinated Universal Time (UTC). An ISO 8601 UTC offset, in basic format, is specified as a positive or negative difference in units of hours and minutes (e.g., +hhmm). If minutes are zero, then they may be omitted and the format would be specified in units of hours (e.g., +hh). The time is specified as a 24-hour clock. Hour values are from 00 to 24, and minute values are from 00 through 59. Hour and minute values are 2-digits with high-order zeroes required to maintain digit count. The extended format for ISO 8601 makes use of a colon (i.e., “:”) character as a separator of the hour and minute substrings.

This is not a property for capturing the daylight savings time rules observed by the vCard object.

This property is identified by the property name **TZ**. This property is specified in a manner consistent with ISO 8601. The property value is a signed numeric indicating the number of hours and possibly minutes from UTC. Time zones east of UTC are positive numbers. Time zones west of UTC are negative numbers. Support for this property is optional for vCard Writers conforming to this specification. An example of the Eastern Standard Time (EST) zone value for this property follows using the basic format:

```
TZ:-0500
```

An example of the Pacific Standard Time (PST) zone value for this property follows using the extended format:

```
TZ:-08:00
```

2.4.6 Geographic Position

This property specifies information related to the global positioning of the vCard object. The property specifies a longitude and latitude. The latitude represents the location north and south of the equator as a positive or negative number, respectively. The longitude represents the location east and west of the prime meridian as a positive or negative number, respectively.

The rationale behind providing support for this property is that it is relatively simple for a vCard object to provide this information compared with how difficult it would be for a receiver of a vCard to determine the global location through some other means.

This property is identified by the property name **GEO**. An example of this property follows:

```
GEO:37.24,-17.87
```

Support for this property is optional for vCard Writers conforming to this specification.

2.5 Organizational Properties

These property types are concerned with information associated with characteristics of the organizations or organizational unites associated with the vCard object.

2.5.1 Title

This property specifies the job title, functional position or function of the individual associated with the vCard object within an organization. This property is based on the X.520 Title attribute. For example, “Vice President, Research and Development”.

This property is identified by the property name **TITLE**. An example of this property follows:

```
TITLE:V.P., Research and Development
```

Support for this property is optional for vCard Writers conforming to this specification.

2.5.2 Business Category

This property specifies information concerning the role, occupation, or business category of the vCard object within an organization. This property is based on the X.520 Business Category explanatory attribute. For example, “Programmer”. This property is included as an organizational property to avoid confusion with the Title property and to avoid incorrect use of the Title property to record Business Category information.

This property is identified by the property name **ROLE**. An example of this property follows:

```
ROLE:Executive
```

Support for this property is optional for vCard Writers conforming to this specification.

2.5.3 Logo

This property specifies an image or graphic of the logo of the organization that is associated with the individual to which the vCard belongs.

This property is identified by the property name **LOGO**. An example of a value for a GIF formatted image of a logo in Base 64 encoding for this property follows:


```
LOGO;ENCODING=BASE64;TYPE=GIF:
    R01GODdhfgA4AOYAAAAAAK+vr62trVIxa6WlpZ+fnzEpCEpzlAha/0Kc74+PjyGM
    SuecKRhrtX9/fzExORBSjCEYCGtra2NjYyF7nDGE50JrhAg51qWtOTl7vee1MWu1
    50o5e3PO/3sxcwAx/4R7GBgQOcDAwFoAQt61hJyMGHuUSpRKIf8A/wAY54yMjHtz
...
```


Support for this property is optional for vCard Writers conforming to this specification.

2.5.3.1 Logo Format Type

This property parameter is provided to specify the graphics format for the Logo property value, if it is different than the default value for the vCard object. The property parameter includes the following values:

Description	Property Parameter Value
TYPE=	
Indicates Graphics Interchange Format	GIF
Indicates ISO Computer Graphics Metafile	CGM
Indicates MS Windows Metafile	WMF
Indicates MS Windows Bitmap	BMP
Indicates IBM PM Metafile	MET
Indicates IBM PM Bitmap	PMB
Indicates MS Windows DIB	DIB
Indicates an Apple Picture format	PICT
Indicates Tagged Image File Format	TIFF
Indicates Adobe Page Description Format	PDF
Indicates Adobe PostScript	PS
Indicates ISO JPEG format	JPEG
Indicates ISO MPEG format	MPEG
Indicates ISO MPEG version 2 format	MPEG2
Indicates Intel AVI format	AVI

Indicates Apple QuickTime format	QTIME

2.5.4 Agent

This property specifies information about another person who will act on behalf of the vCard object. Typically this would be an area administrator, assistant, or secretary for the individual. A key characteristic of the Agent property is that it represents somebody or something which is separately addressable. For example, if all phone calls or e-mail messages are normally screened by an agent, this property may not be needed. On the other hand, if an agent can act as a proxy, and may otherwise need to be contacted separately, then an Agent property is useful.

This property is equivalent to nesting another vCard with the specified vCard.

This property is identified by the property name **AGENT**. The value of this property is a string containing another vCard object. An example of this property follows:

```
AGENT:
BEGIN:VCARD
VERSION:2.1
N:Friday;Fred
TEL;WORK;VOICE:+1-213-555-1234
TEL;WORK;FAX:+1-213-555-5678
END:VCARD
```

Support for this property is optional for vCard Writers conforming to this specification.

2.5.5 Organization Name and Organizational Unit

This property specifies the name and optionally the unit(s) of the organization associated with the vCard object. This property is based on the X.520 Organization Name attribute and the X.520 Organization Unit attribute. For example, “The AB Corporation” and the “North American Division”.

This property is identified by the property name **ORG**. This property is defined to encapsulate the Organization Name and Organization Unit properties as sub-properties. The property value consists of the components of the organization specified as positional fields separated by the Field Delimiter (ASCII decimal 59). The property value is a concatenation of the Organization Name (first field), Organizational Unit (second field) strings. Additional positional fields, if specified, contain additional Organizational Units. The following is an example of the Organization property:

```
ORG:ABC, Inc.;North American Division;Marketing
```

Support for this property is optional for vCard Writers conforming to this specification.

2.6 Explanatory Properties

These property types are concerned with additional explanations, such as that related to national language support, annotation, or encoding of binary information about the vCard object.

2.6.1 Comment

This property specifies supplemental information or a comment that is associated with the vCard. With the use of property grouping, the association can be limited to a group of properties. The property is based on the X.520 Description attribute.

This property is identified by the property name **NOTE**. An example of this property follows:

```
NOTE;ENCODING=QUOTED-PRINTABLE:This facsimile machine if operational=
 0830 to 1715 hours=0D=0A=
Monday through Friday. Call +1-213-555-1234 if you have problems=0D=0A=
with access to the machine.
```

Support for this property is optional for vCard Writers conforming to this specification.

2.6.2 Last Revision

This property specifies the combination of the calendar date and time of day of the last update to the vCard object. The property value is a character string conforming to the basic or extended format of ISO 8601. The value can either be in terms of local time or UTC.

This property is identified by the property name **REV**. Valid values for this property are a character string representing a combination of the calendar date and time of day conforming to the basic or extended format of ISO 8601. The time of day can be either local time or UTC. The following example is in the basic format and local time of ISO 8601:

```
REV:19951031T222710
```

The following example is in the extended format and UTC time of ISO 8601:

```
REV:1995-10-31T22:27:10Z
```

Support for this property is optional for vCard Writers conforming to this specification.

2.6.3 Sound

This property specifies a sound annotation for the vCard object. By default, if this property is not grouped with other properties it specifies the pronunciation of the Formatted Name property of the vCard object. Such information may be in the form of a string of characters representing a phonetic sound or in the form of a digitized sound, or both; subject to the limitations imposed by the encoding used to communicate the vCard.

This property is identified by the property name **SOUND**. Valid values for this property are either a string representation, a reference to a digital audio representation, or an inline digital audio representation of the phonetic pronunciation of the Formatted Name property. The following example shows the string based phonetic representation:

```
SOUND:JON Q PUBLIK
```

The following example shows the digtial sound representation and URL based value:

```
SOUND;VALUE=URL:file///multimed/audio/jqpublic.wav
```

The following example shows the digtial sound representation and INLINE value:

```
SOUND;WAVE;BASE64:
    UklGRhAsAABXQVZFZm10IBAAAAABAAEAESsAABErAAABAAgAZGF0YesrAACAg4eC
    eXR4e3uAhoiIiYmKjIiDfnx5eX6CgoKEhYWDenV5fH6BhISGiIiDfHZ2eXt/hIiK
    jY2IhH12d3Vyc3uDiIiFf3l7fn18eXl+houFf319fnyAgHl5eoCIiISChIeAfnt2
...
```

Support for this property is optional for vCard Writers conforming to this specification.

2.6.3.1 Sound Digital Audio Type

This property parameteris provided to specify the type of the digital audio Pronunciation for the vCard object. The property parameter can have the following values:

Description	Property Parameter Value
TYPE=	
Indicates Wave format	WAVE
Indicates MIME basic audio type	PCM
Indicates AIFF format	AIFF

2.6.4 Uniform Resource Locator

This property specifies a value that represents a Uniform Resource Locator (URL). An URL is a representation of an Internet location that can be used to obtain real-time information about the vCard object. Application of this property might be to specify the location of a publicly accessible directory where up-to-date or additional information on the individual or resource associated with a vCard can be found.

This property is identified by the property name **URL**. Valid values for this property are a string conforming to the IETF RFC 1738, *Uniform Resource Locators*. The following is an example of this property:

```
URL:http://abc.com/pub/directory/northam/jpublic.ecd
```

Support for this property is optional for vCard Writers conforming to this specification.

2.6.5 Unique Identifier

This property specifies a value that represents a persistent, globally unique identifier associated with the object. The property can be used as a mechanism to relate different vCard objects.

Some examples of valid forms of unique identifiers would include ISO 9070 formal public identifiers (FPI), X.500 distinguished names, machine-generated “random” numbers with a statistically high likelihood of being globally unique and Uniform Resource Locators (URL). If an URL is specified, it is suggested that the URL reference a service which will produce an updated version of the vCard.

This property is identified by the property name **UID**. This property is provided to enable a vCard Reader and Writer to uniquely identify either a vCard object instance or properties within a vCard object. Valid values for this property are a unique character string. The following is an example of this property:

```
UID:19950401-080045-40000F192713-0052
```

Support for this property is optional for vCard Writers conforming to this specification.

2.6.6 Version

This property specifies the identifier corresponding to the highest version number of the vCard Specification supported by the implementation that created the vCard object. The value of this property must be 2.1 to correspond to this specification..

This property is identified by the property name **VERSION**. The following is an example of this property:

```
VERSION:2.1
```

Support for this property is mandatory for implementations conforming to this specification. This property must appear within the vCard data stream.

2.7 Security Properties

These property types are concerned with the security of the information in the vCard object.

2.7.1 Public Key

This property specifies the public encryption key associated with the vCard object.

This property is identified by the property name **KEY**. Valid values for this property are a public key that conforms to a bilaterally agreed to representation. If the representation is a binary format, then the public key must be further encoded. The default format is clear-text. If a binary format is used, then it is specified by the property parameter. Support for this property is optional for vCard Writers conforming to this specification.

2.7.2 Key Type

This property parameter is provided to specify the type of the public key for the vCard object. The property parameter can have the following values:

Description	Property Parameter Value
TYPE=	
Indicates a X.509 public key certificate type of key	X509
Indicates an IETF PGP type of key	PGP

2.8 Miscellaneous Properties

2.8.1 Extensions

The vCard provides a “standard mechanism for doing non-standard things”. This extension support is provided for implementers to “push the envelope” on the existing version of the specification. Extension properties are specified by property and/or property parameter names that have the initial sub-string of **X-** (the two character sequence: Capital X character followed by the Dash character. It is recommended that vendors concatenate onto this sentinel an added short sub-string to identify the vendor. This will facilitate readability of the extensions and minimize possible collision of names between different vendors. For example, the following might be the ABC vendor’s extension for a video-clip form of identification property:

```
X-ABC-VIDEO;MPEG2:http://lonestar.bubbas.org/billibob.mpg
```

or, the following example might be an extension for grouping vCard objects into a distribution list for the Design Work Group.

```
BEGIN:VCARD
VERSION:2.1
X-DL;Design Work Group:List Item 1;List Item 2;List Item 3
BEGIN:VCARD
UID:List Item 1
N:John Smith
TEL:+1-213-555-1111
END:VCARD
BEGIN:VCARD
UID:List Item 2
N:I. M. Big
TEL:+1-213-555-9999
END:VCARD
BEGIN:VCARD
UID:List Item 3
N:Jane Doe
TEL:+1-213-555-5555
END:VCARD
END:VCARD
```

At present, there is no registration authority for names of extension properties.

Support for this property is mandatory for implementations conforming to this specification. However, an implementation may not be able to act on the extension property. Conformance only requires that an implementation be able to parse vCard data streams with extensions. The implementation need not act on them.

2.9 Formal Definition

The following modified Backus-Naur Notation (BNF) is provided to assist developers in building parsers for the vCard.

```
This syntax is written according to the form described in RFC 822, but it
references just this small subset of RFC 822 literals:

  CR         =  <ASCII CR, carriage return>  ; (     15,      13.)
  LF         =  <ASCII LF, linefeed>         ; (     12,      10.)
  CRLF       =  CR LF
  SPACE      =  <ASCII SP, space>            ; (     40,      32.)
  HTAB       =  <ASCII HT, horizontal-tab>   ; (     11,       9.)

All literal property names are valid as upper, lower, or mixed case.

ws    = 1*(SPACE / HTAB)
  ; "whitespace," one or more spaces or tabs

wsls      = 1*(SPACE / HTAB / CRLF)
  ; whitespace with line separators

word      = <any printable 7bit us-ascii except []=:., >

groups        = groups "." word
      / word

vcard_file    = [wsls] vcard [wsls]

vcard     = "BEGIN" [ws] ":" [ws] "VCARD" [ws] 1*CRLF
       items *CRLF "END" [ws] ":" [ws] "VCARD"

items     = items *CRLF item
      / item
  ; these may be "folded"

item      = [groups "."] name
        [params] ":" value CRLF
      / [groups "."] "ADR"
        [params] ":" addressparts CRLF
      / [groups "."] "ORG"
        [params] ":" orgparts CRLF
      / [groups "."] "N"
        [params] ":" nameparts CRLF
      / [groups "."] "AGENT"
        [params] ":" vcard CRLF
  ; these may be "folded"

name      = "LOGO" / "PHOTO" / "LABEL" / "FN" / "TITLE"
      / "SOUND" / "VERSION" / "TEL" / "EMAIL" / "TZ" / "GEO" / "NOTE"
      / "URL" / "BDAY" / "ROLE" / "REV" / "UID" / "KEY"
      / "MAILER" / "X-" word
  ; these may be "folded"

value     = 7bit / quoted-printable / base64

7bit      = <7bit us-ascii printable chars, excluding CR LF>

8bit      = <MIME RFC 1521 8-bit text>

quoted-printable = <MIME RFC 1521 quoted-printable text>
```

```
base64        = <MIME RFC 1521 base64 text>
   ; the end of the text is marked with two CRLF sequences
   ; this results in one blank line before the start of the next property

params        = ";" [ws] paramlist

paramlist = paramlist [ws] ";" [ws] param
      / param

param     = "TYPE" [ws] "=" [ws] ptypeval
      / "VALUE" [ws] "=" [ws] pvalueval
      / "ENCODING" [ws] "=" [ws] pencodingval
      / "CHARSET" [ws] "=" [ws] charsetval
      / "LANGUAGE" [ws] "=" [ws] langval
      / "X-" word [ws] "=" [ws] word
      / knowntype

ptypeval  = knowntype / "X-" word

pvalueval = "INLINE" / "URL" / "CONTENT-ID" / "CID" / "X-" word

pencodingval  = "7BIT" / "8BIT" / "QUOTED-PRINTABLE" / "BASE64" / "X-" word

charsetval    = <a character set string as defined in Section 7.1 of
      RFC 1521>

langval       = <a language string as defined in RFC 1766>

addressparts  = 0*6(strnosemi ";") strnosemi
   ; PO Box, Extended Addr, Street, Locality, Region, Postal Code,
   Country Name

orgparts  = *(strnosemi ";") strnosemi
   ; First is Organization Name, remainder are Organization Units.

nameparts = 0*4(strnosemi ";") strnosemi
   ; Family, Given, Middle, Prefix, Suffix.
   ; Example:Public;John;Q.;Reverend Dr.;III, Esq.

strnosemi = *(*nonsemi ("\;" / "\" CRLF)) *nonsemi
   ; To include a semicolon in this string, it must be escaped
   ; with a "\" character.

nonsemi       = <any non-control ASCII except ";">

knowntype = "DOM" / "INTL" / "POSTAL" / "PARCEL" / "HOME" / "WORK"
      / "PREF" / "VOICE" / "FAX" / "MSG" / "CELL" / "PAGER"
      / "BBS" / "MODEM" / "CAR" / "ISDN" / "VIDEO"
      / "AOL" / "APPLELINK" / "ATTMAIL" / "CIS" / "EWORLD"
      / "INTERNET" / "IBMMAIL" / "MCIMAIL"
      / "POWERSHARE" / "PRODIGY" / "TLX" / "X400"
      / "GIF" / "CGM" / "WMF" / "BMP" / "MET" / "PMB" / "DIB"
      / "PICT" / "TIFF" / "PDF" / "PS" / "JPEG" / "QTIME"
      / "MPEG" / "MPEG2" / "AVI"
      / "WAVE" / "AIFF" / "PCM"
      / "X509" / "PGP"
```

Section 3 : Internet Recommendations

3.1 Recommended Practice with SMTP/MIME

The vCard information can be transported through SMTP/MIME based electronic mail services. Interoperability of vCard information over SMTP/MIME transports can be better assured by following a common set of recommended practices for encapsulation of the vCard.

3.1.1 Text/Plain Content Type

Without any change to existing SMTP or MIME compliant user agents, a vCard can be included within Internet email messages. This might be the case for an existing, simple user agent such as a legacy SMTP mail system. While this approach provides for transport of vCards over SMTP services, it does not allow for the end user to take advantage of the full capabilities of either the vCard or Internet email (i.e., MIME) functionality.

The following demonstrates how a vCard can be included as an epilog to a SMTP message made up of a RFC 822 message. This may be an initial method for incorporating vCard objects into SMTP messages.

```
Date: Thr, 25 Jan 96 0932 EDT
From: john.smith@host.com
Subject: Re: RFC822 vCard Example
Sender: john.smith@host.com
To: smartin@host2.com
Message-ID: <JOHNSMITH.960125T091020.xyzMail@host3.com>

Steve:  Thanks for the call earlier today. I am unable to
use your material at this time. Please feel free to contact
me in the future.
BEGIN:VCARD
VERSION:2.1
N:Smith;John;M.;Mr.;Esq.
TEL;WORK;VOICE;MSG:+1 (919) 555-1234
TEL;WORK;FAX:+1 (919) 555-9876
ADR;WORK;PARCEL;POSTAL;DOM:Suite 101;1 Central St.;Any Town;NC;27654
END:VCARD
```

The following example demonstrates how a vCard can be included as a separate text/plain content portion within current MIME user agents.

```
Date: Fri, 26 Jan 1996 07:53:00 -0500
From: smartin@host2.com
Subject: RE: Text/Plain MIME vCard Example
To: fdawson@VNET.IBM.COM
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary=vcard
Message-ID: <ABC-1.00-Note-martin-steve-0824475754>

--vcard
```

```
Content-Type:text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

John:  I have looked over my material and feel that you may
have over looked a couple of appropriate pieces. Please give
me a call so that we can discuss further.

--vcard
Content-Type:text/plain; charset=us-ascii; name="MARTIN.VCF"


BEGIN:VCARD
VERSION:2.1
N:Martin;Stephen
TEL;HOME;VOICE:+1 (210) 555-1357
TEL;HOME;FAX:+1 (210) 555-0864
ADR;WORK;PARCEL;POSTAL;DOM:123 Cliff Ave.;Big Town;CA;97531
END:VCARD

--vcard--
```

3.1.2 Text/X-vCard Content Type

A vCard object may also be transferred in a (RFC 1521) MIME entity as a non-standard "text/x-vCard" content-type. This (RFC 1521) MIME type maybe useful in those cases where the MIME compliant messaging service does not yet support the "application/directory" and "multipart/related" MIME content-types and yet the specificity of a calendaring and scheduling media type is required.

The following example demonstrates how a vCard can be included as a separate non-standard text/x-vCard content portion within current MIME user agents.

```
Date: Fri, 26 Jan 1996 07:53:00 +0000
From: smartin@host2.com
Subject: RE: Text/x-vCard MIME vCard Example
To: fdawson@VNET.IBM.COM
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary=vcard
Message-ID: <ABC-1.00-Note-martin-steve-0824475754>


--vcard
Content-Type:text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

John:  I have looked over my material and feel that you may
have over looked a couple of appropriate pieces. Please give
me a call so that we can discuss further.

--vcard
Content-Type:text/x-vCard; charset=us-ascii; name="MARTIN.VCF"


BEGIN:VCARD
VERSION:2.1N:Martin;Stephen
TEL;HOME;VOICE:+1 (210) 555-1357
TEL;HOME;FAX:+1 (210) 555-0864
ADR;WORK;PARCEL;POSTAL;DOM:123 Cliff Ave.;Big Town;CA;97531
END:VCARD
```

```
--vcard--
```

3.1.3 Application/Directory Content Type

The Internet Engineering Task Force (IETF) Access and Searching of Internet Directories (ASID) working group has produced an Internet Draft defining the “application/directory” MIME content type. The current draft name is draft-ietf-asid-mime-direct-01.txt. This specification is intended to be aligned with this work. Internet Drafts are working documents of an IETF working group, valid for at most six months, and should be considered “works in progress”.

This MIME content type was designed to be used to transport directory information across MIME based electronic mail services. The internet draft is directly applicable to the exchange of business card data, such as that defined by the vCard specification.

The ***versit*** **PDI Team** has worked within the IETF ASID Working Group to draft an application/directory profile that registers the method for transporting a vCard as an application/directory Content-Type. The current draft name is draft-ietf-asid-mime-vcard-00.txt. This work is expected to be progressed to a Request For Comment after the publication of this version of the vCard specification. In the interim, the following guidelines are provided to describe how a vCard might be conveyed using the application/directory draft specification.

A vCard should be included in a MIME message that has a Content-Type header field value of “multipart/related”. The vCard is included in the message as the primary body part. The position of the body part entity can also be specified with the “start=“ parameter. This MIME body part entity has a Content-Type body part header field value of “application/directory” with a “profile” parameter value of “vcard”. Any vCard binary information, such as a logo, picture, or digital audio pronunciation can be included inline within the vCard, as is specified by the vCard specification. Preferably, the binary information should be extracted from the vCard object and contained in the MIME message as secondary body part entities. The binary content in the secondary body part entities can be referenced from within the vCard object through the use of the “VALUE=” property parameter. In this latter case, the binary information should be transformed into a content type nominally supported by MIME user agents. For image content, this would be the Graphics Image Format (GIF) or Joint Picture Encoding Group (JPEG) formats. For audio content, this would be the 8-bit mu-law (PCM) format specified by the MIME specification.

The following example defines how this might be specified:

```
Date: Mon, 29 Jan 96 0830 EDT
From: john.smith@host.com
Subject: Re: MIME application/directory vCard Example
Sender: john.smith@host.com
To: smartin@host2.com
Message-ID: <JOHNSMITH.960129T083020.xyzMail@host3.com>
Content-Type: multipart/related; boundary="vcard";
          type=application/directory;
          start=<JOHNSMITH.part1.960129T083020.xyzMail@host3.com>

--vcard
Content-Type: application/directory; charset=us-ascii;
```

```
          source="file://versit.or2"; profile="vcard"
Content-ID: <<JOHNSMITH.part1.960129T083020.xyzMail@host3.com>

BEGIN:VCARD
VERSION:2.1
N:Smith;John;M.;Mr.;Esq.
TEL;WORK;VOICE;MSG:+1 (919) 555-1234
TEL;CELL:+1 (919) 554-6758
TEL;WORK;FAX:+1 (919) 555-9876
PHOTO;GIF;MIME:<<JOHNSMITH.part3.960129T083020.xyzMail@host3.com>
ADR;WORK;PARCEL;POSTAL;DOM:Suite 101;1 Central St.;Any Town;NC;27654
END:VCARD

--vcard
Content-Type: text/plain; charset=us-ascii
Content-ID: <<JOHNSMITH.part2.960129T083020.xyzMail@host3.com>

Steve:

I am not in the office today. You may want to try
reaching me either on my cellular telephone or fax your
new ideas to my office.

Let's setup a face-to-face meeting later this week, after I review
your updated material. I am including a picture in my business card
data, since we have not met yet.

-- John

--vcard
Content-Type: image/gif
Content-ID: <<JOHNSMITH.part3.960129T083020.xyzMail@host3.com>

...image data would go here...

--vcard--
```

3.2 Recommended Practice with HTTP/HTML

A vCard object should be transferred over HTTP with the non-standard MIME type/subtype value of "text/x-vCard". The non-standard subtype should be used because the vCard has not been registered as a MIME media type with the IANA.

The vCard information can be captured with a FORM type of HTML document. Interoperability of of vCard information can be better assured by following a common set of recommended practices for mapping vCard information into and out of HTML documents.

3.2.1 Form Element Usage

The HTML **FORM** element is a useful method for capturing data intended for input into individual vCard property values. The following recommended practices are provided for such use.

3.2.2 Mapping To INPUT Element Attribute Names

An HTML form data set is a useful mechanism for capturing vCard data within the Internet

WWW. The use of a consistent naming scheme for the name attributes within a form element will permit implementations to support automatic fill-in of forms with existing vCard data. In addition, such a consistent naming scheme will provide a greater assurance of interoperability between HTML based applications that use vCard data.

The following table provides a recommended mapping of vCard properties and name attributes within a form element.

Identification Properties

Description	Attribute Name	Comment
Formatted Name	FN	
Name	N	Individual components of name property are captured as separate input elements with the names N.Family, N.First, N.Middle, N.Prefix, N.Suffix.
Photograph	PHOTO	Only the URL based specification is supported by this mapping. Value is the URL for the graphic.
Photograph Format Type	PHOTO.Type	Where the value is one of the enumerated strings defined by the vCard specification.
Birthdate	BDAY	

Delivery Addressing Properties

Description	Attribute Name	Comment
Delivery Address	ADR	TYPE=TEXTAREA
Address Type	ADR.x	TYPE=CHECKBOX. Separate input elements are used to capture the possible delivery types. The elements are named ADR.x, where x is one of the enumerated strings defined by the vCard specification.
Delivery Label	LABEL	
Label Type	LABEL.x	TYPE=CHECKBOX. Separate input elements are used to capture the possible delivery types. The elements are named LABEL.x, where x is one of the enumerated strings defined by the vCard specification.

Telecommunications Addressing Properties

Description	Attribute Name	Comment
Telephone Number	TEL	
Telephone Type	TEL.x	TYPE=CHECKBOX. Separate input elements are used to capture the possible telephone types. The elements are named TEL.x, where x is one of the enumerated strings defined by the vCard specification.
Electronic Mail Address	EMAIL	
Electronic Mail Address Type	EMAIL.Type	Selection option from a list of alternatives.
Mailer	MAILER	

Geographical Properties

Description	Attribute Name	Comment
Time Zone	TZ	
Geographic Position	GEO	

Organizational Properties

Description	Attribute Name	Comment
Title	TITLE	
Business Category	ROLE	
Logo	LOGO	Only the URL based specification is supported by this mapping. Value is the URL for the graphic.
Logo Format Type	LOGO.Type	Where the value is one of the enumerated strings defined by the vCard specification.
Agent		Captured through a separate form element using the mapping defined in these tables.
Organization	ORG	TYPE=TEXT. Separate input elements for the organizational name and unit. The name ORG.Name is used to capture the organizational name. The name ORG.UNIT is used to capture the organizational unit. If there are multiple organizational units, it is captured in a form with name attributes ORG.UNIT1, ORG.UNIT2, etc.

Explanatory Properties

Description	Attribute Name	Comment
Comment	NOTE	TYPE=TEXT
Last Revision	REV	A hidden field.
Version	VERSION	A hidden field with the value set to the string "2.1".
Language	LANG	A hidden field with the value set to the string associated with the default language used in the form (e.g., US-eng).
Sound	SOUND	TYPE=TEXT
Sound Type	N/A	
Uniform Resource Locator	URL	TYPE=TEXT
Unique Identifier	UID	TYPE=TEXT
Binary Encoding	BE.x	Where x is one of the enumerated encoding types defined by the vCard specification.

Security Properties

Description	Attribute Name	Comment
Public Key	KEY	
Key Type	KEY.Type.x	Where x is one of the enumerated encoding types defined by the vCard specification.
MISCELLANEOUS PROPERTIES		
Extensions	X-x	Where x is a string defined by the extension author.

Where multiple properties (e.g., telephone numbers) appear, a label prefix should be used. For example, telephone #1 might have a name attribute of "A.TEL", telephone #2 might have a name attribute of "B.TEL", etc.

3.2.3 Example HTML Code

The following HTML code is an example of the use of the mapping of INPUT element attributes names to vCard property names. The code can be used to capture input data for creating a vCard on a Web homepage.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<head>
<title>Create Your Own Versitcard</title>
</head>
<IMG src="versit.gif">
<h1>Create Your Own Versitcard</h1>
<P> Fill out this form and we'll
create a <b>Versitcard</b> for you and send it to the email address of your
choice,
along with more information on the Versitcard format.</P>
<hr><!-- Identification And Organizational Properties -->
<FORM METHOD="POST" ACTION="/cgi-bin/vcard-maker">
Formatted Name:<INPUT name="FN" type=text size=32 maxlength=64
value=""><br>
Phoenetic Pronunciation:<INPUT name="SOUND" type=text size=32 maxlength=128
value=""><br>
Company Name:<INPUT name="ORG.Name" type=text  size=32 maxlength=64
value=""><br>
Company Unit:<INPUT name="ORG.Unit" type=text size=32
maxlength=64 value=""><br>
Title:<INPUT name="TITLE" type=text size=32 maxlength=64
value="">
<hr><!-- Name Property Component Values -->
Family Name:<INPUT name="N.Family" type=text size=32 maxlength=64
value=""><br>
Given Name:<INPUT name="N.Given" type=text size=32
maxlength=64 value=""><br>
Middle Name:<INPUT name="N.Middle" type=type size=32
maxlength=64 value=""><br>
Name Prefix:<INPUT name="N.Prefix" type=type size=32
maxlength=64 value=""><br>
Name Suffix:<INPUT name="N.Suffix" type=type size=32
maxlength=64 value=""><br>
<hr><!-- Delivery Addressing Properties -->
Delivery Label:<TEXTAREA name="LABEL" cols=64 ROWS=5>
</TEXTAREA><br><br>
Post Office Address:<INPUT name="ADR.POAddr" type=text size=32
maxlength=64 value=""><br>
Extended Address:<INPUT name="ADR.ExtAddr" type=text size=32
maxlength=64 value=""><br>
Street Address:<INPUT name="ADR.Street" type=text size=62
maxlength=128 value=""><br>
City:<INPUT name="ADR.Locality" type=text size=16 maxlength=32
value="">
Region:<INPUT name="ADR.Region" type=text size=16 maxlength=32
value="">
Postal Code:<INPUT name="ADR.PostalCode" type=text size=16 maxlength=32
value=""><br>
Country Name:<INPUT name="ADR.CountryName" type=text size=16 maxlength=32
value="USA">
<INPUT type=checkbox name="ADR.Work" value=WORK checked>Work
<INPUT type=checkbox name="ADR.Home" value=HOME>Home
<INPUT type=checkbox name="ADR.Parcel" value=PARCEL checked>Parcel <INPUT
type=checkbox name="ADR.Postal" value=POSTAL checked>Postal<br>
<hr><!-- Geographical Properties -->
TimeZone:<INPUT name="TZ" type=text size=3 maxlength=8
value="-06">
Location:<INPUT name="GEO" type=text size=16 maxlength=32 value=""><br>
```

```
<hr><!-- Telephony Addressing Properties -->
<!-- Telephone #1 -->
Telephone #1:<INPUT type=text name="A.TEL" size=20 maxlength=40 value="+1
(000) 000-0000"><br>
<INPUT type=checkbox name="A.TEL.Work" value=WORK checked>Work
<INPUT type=checkbox name="A.TEL.Home" value=HOME>Home
<INPUT type=checkbox name="A.TEL.Voice" value=VOICE checked>Voice
<INPUT type=checkbox name="A.TEL.Msg" value=MSG checked>Msg <INPUT
type=checkbox name="A.TEL.Fax" value=FAX>Fax <INPUT type=checkbox
name="A.TEL.Prefer" value=PREFER checked>Preferred<br>
<hr><!-- Telephone #2 -->
Telephone #2:<INPUT type=text name="B.TEL" size=20 maxlength=40 value="+1
(000) 000-0000"><br>
<INPUT type=checkbox name="B.TEL.Work" value=WORK checked>Work <INPUT
type=checkbox name="B.TEL.Home" value=HOME>Home
<INPUT type=checkbox name="B.TEL.Voice" value=VOICE>Voice <INPUT
type=checkbox name="B.TEL.Msg" value=MSG>Msg
<INPUT type=checkbox name="B.TEL.Fax" value=FAX checked>Fax
<INPUT type=checkbox name="B.TEL.Prefer" value=PREFER>Preferred<br>
<hr><!-- Telephone #3 -->
Telephone #3:<INPUT type=text name= "C.TEL" size=20 maxlength=40 value="+1
(000) 000-0000"><br>
<INPUT type=checkbox name="C.TEL.Work" value=WORK>Work
<INPUT type=checkbox name="C.TEL.Home" value=HOME checked>Home <INPUT
type=checkbox name="C.TEL.Voice" value=VOICE checked>Voice <INPUT
type=checkbox name="C.TEL.Msg" value=MSG checked>Msg
<INPUT type=checkbox name="C.TEL.Fax" value=FAX checked>Fax <INPUT
type=checkbox name="D.Prefer" value=PREFER>Preferred<br>
<hr><!-- Email D -->
EmailAddress: <select name="D.EMAILTYPE">
<option selected>INTERNET:
<option>CompuServe:
<option>AOL:
<option>Prodigy:
<option>eWorld:
<option>AppleLink:
<option>AppleTalk:
<option>PowerShare:
<option>IBMMail:
<option>ATTMail:
<option>MCIMail:
<option>X.400:
<option>TLX:
</select><INPUT type=text name="D.EMAIL" size=32 maxlength=64 value="">
<INPUT type=checkbox name="D.EMAIL.Work" value=WORK checked>Work <INPUT
type=checkbox name="D.EMAIL.Home" value=HOME checked>Home<br>
<hr><!-- End of vCard Input -->
Send my Versitcard to this <b>internet</b> email address:
<INPUT type=text name="SENDTOADDR" size=32 maxlength=64 value=""><br> Press
<INPUT TYPE=SUBMIT value="Send"> to send the form now. Or, press <INPUT
TYPE=RESET value="Reset"> to reset values to the form defaults.
</form>
</body>
```

Section 4 : UI Support Recommendations

When integrating vCard support into an application, an implementor needs to consider a number of user interface (UI) implications. Most applications provide some levels of support for interacting with other applications. This is usually accomplished in three ways. These include the File System, Clipboard, and Drag/Drop. The full potential of the vCard technology can be better utilized if an application supports the vCard in each of these UI actions.

4.1 File System

It is recommended that applications integrating support for vCard specification provide support for importing and exporting vCard objects from the operating system's file system. In operating systems that support file types, it is recommended that a file type of ***VCF*** be used to distinguish the vCard objects. Applications should make use of the file system capabilities to support the FileOpen and FileSaveAs, or their equivalent function, of a vCard object.

4.2 Clipboard

It is recommended that applications integrating support for the vCard specification provide UI capabilities for exchanging vCard objects through the operating system's clipboard. In operating systems that provide support for registering clipboard format types, it is recommended that the vCard object be registered using the string ***+//ISBN 1-887687-00-9::versit::PDI//vCard***. This string is an ISO 9070 Formal Public Identifier (FPI). Applications should make use of the operating system's clipboard capability to support the Cut, Copy, and Paste, or their equivalent function, of a vCard object. Applications copying a vCard to the clipboard should put the vCard object on to the clipboard in both the vCard registered format and a plain text format.

4.3 Drag/Drop

It is recommended that applications integrating support for the vCard specification provide UI capabilities for exchanging vCard objects through the operating system's drag/drop capability. In operating systems that provide support for registering drag/drop object types, it is recommended that the vCard object be registered using the string ***+//ISBN 1-887687-00-9::versit::PDI//vCard***. This string is an ISO 9070 Formal Public Identifier (FPI). Applications should make use of the operating system's drag/drop capability to enable the application to act as either a Drag Source and Drag Target, or their equivalent function, of a vcard object. Applications acting as a Drag Source should advertise their ability to render the vCard in both the vCard registered format and a plain text format

Where an operating system environment provided multiple drag/drop protocols (e.g., file specification or clipboard based), it is recommended that an implementation provide negotiated support for both. For example, the file specification based drag/drop protocol is useful when dragging a desktop file object or a web based URL to a target application. In addition, the clipboard based drag/drop protocol is useful when dragging an event or todo from a source within an application to a target in another application. Supporting just one of these mechanisms will unnecessarily lead to a lack of interoperability between applications supporting this specifications.

.

Section 5 : Conformance

In order for a vCard Reader or Writer to conform to this specification it must meet the following criteria:

- Implement at least one of the syntaxes defined by this specification.
- All properties must be implemented as defined. Statements elsewhere in the specification which describe features as optional or with exceptions take precedence over this criterion.
- Character set support is up to the underlying implementation. However, support for the default character set (i.e., US ASCII) is required. Optionally, other character sets may be supported.
- All extensions are optional. It is requested that any vendor-specific extensions include the vendor identification sub-string in the extension name. For example, the extension name *X-ABC-* for an extension created by the ABC organization.
- All vendor defined extensions must declare the minimum conformance for that extension.

Additionally, in order for a vCard Reader to conform to this specification it must meet the following additional criteria:

- Must be able to parse all properties.
- All forms of vCard Grouping must be able to be parsed and processed.
- Property Grouping must be able to be parsed and processed.

Additionally, in order for a vCard Writer to conform to this specification it must meet the following additional criteria:

- Must be able to send at least the Version, Formatted Name, Name, Address, Telephone, Email, and Mailer properties.