vCalendar

The Electronic Calendaring and Scheduling Exchange Format

Version 1.0

A ***versit*** Consortium Specification

September 18, 1996

Copyrights

Trademarks

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

Apple and the Apple Logo are trademarks of Apple Computer, Inc. registered in the U.S. and other countries.

AT&T is a registered trademarks of AT&T Corp.

IBM is a registered trademarks of International Business Machines Corporation.

Contributors

Roland H. Alden

Stephen J. Bartlett

Jay Batson, ON Technology

John Binici, Iris Associates

Steve Carter, Novell

Liang-Jye Chang, Starfish Software

Andre Courtemanche, CS&T

Jim Cunnie, AT&T EasyCommerce

Frank Dawson, IBM Corporation

Rik Drummond, The Drummond Group

Gavin Eadie, University of Michigan

Pat Egen, Provident Life and Accident

Randell Flint, Sundial Systems

Ben Forta, OnTime/Division of FTP

Anik Ganguly, OnTime/Division of FTP.

Arvind Goyal, Lotus Development Corporation

David Goodhand, Microsoft

Steve Hanna, ON Technology

John Hansen, Starfish Software

Niraj Jain, Oracle Corporation

Del Jensen, Novell

Bruce M. Johnston, Lotus Development Corporation

Dr. Mark K. Joseph, Attachmate Corporation

Bruce Kahn, Iris Associates

Don Lavange, Novell

Larry Mason, Microsystems Software, Inc.

Skip Montanaro, Automatrix, Inc.

Pete O'Leary, Clear Blue Networking Systems, Inc.

Ron Rassner, Creative Networks, Inc.

Vinod Seraphin, Lotus Development Corporation

Uppili Srivivasan, Oracle Corporation

Tom Steppe, OnTime/Divison of FTP Software

Dean Stevens, Now Software, Inc.

Budi Sutardja

Robert Tatar, Automatrix, Inc.

Yvonne Tso, SunSoft

Mike Weston, Netscape Communications Corporation.

Steve Wincor, Lockheed Martin

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.
- IETF RFC 1738, *Universal Resource Locator*, December 1994.
- IETF Network Working Group RFC 1766, *Tags for the Identification of Languages,* March 1995.
- 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 tmes*, 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/IEC 9070, *Information Technology—SGML Support Facilities—Registration Procedures for Public Text Owner Identifiers,* Second Edition, International Organization for Standardization, April, 1991.
- RFC1521, *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies,* Network Working Group, September, 1993.
- XAPIA CSA, *Calendaring and Scheduling Application Programming Interface (CSA) Version 1.0,* X.400 API Association, November 15, 1994.

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
(201) 327-2803 (Outside USA)
pdi@versit.com
http://www.versit.com

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 calendaring and scheduling (***vCalendar***) format. The vCalendar format allows for the capture of information normally stored within a calendaring and scheduling application; such as a Personal Information Manager or a Group Scheduling product.

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 vCalendar is a data stream consisting of one or more vCalendar objects. The individual vCalendar definitions can be identified and parsed within the data stream. The vCalendar 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 *vCalendar Writer* creates vCalendar data streams and a *vCalendar Reader* interprets vCalendar data streams. The vCalendar 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 vCalendar data stream and a common set of conformance requirements.

This specification provides for a clear-text encoding. The specification also includes a formal grammar for the clear-text encoding to aid in the implementation of parsers and to serve as the definitive reference when ambiguities or questions arise in interpreting the descriptive prose definition of the specification.

The clear-text 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 vCalendar is intended to be used for exchanging information about ***event*** and ***todo*** types of calendaring and scheduling entities. An event is a calendaring and scheduling entity that represents a scheduled amount of time on a calendar. For example, it may be an activity; such as a one-hour, department meeting from 8 AM to 9 AM, tomorrow. A todo is a calendaring and scheduling entity that represents an action-item or assignment. For example, it may be an item of work assigned to an individual; such as “turn in travel expense today”.

In today’s business environment, this information is typically kept on a paper-based day-planner or calendar. More and more, this type of information is being also managed within electronic Personal Information Manager or Group Scheduling products. It is appropriate, then that this specification define this information in terms of a paradigm based on a calendaring and scheduling event and todo entities.

Prior to the introduction of the vCalendar specification, users of such applications typically had to re-key the original information, often transcribing it from paper day-planners, scraps of paper or electronic mail messages. With the advent of the vCalendar specification, this information can be exchanged in an automated and consistent fashion.

The basis for this specification have their origin in openly defined, industry specifications; such as the X.400 API Association’s Calendaring and Scheduling API (CSA). In addition, this specification has capabilities that were derived from the experience of multi-vendor demonstrations of this capability.

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

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 calendaring and scheduling interchange format, such as the vCalendar specification.

This specification is intended to be used as a format for exchange of calendaring and scheduling information from one product to another. This exchange may take place using desktop application interaction techniques; such as a file system FILE-OPEN or FILE-SAVE-AS functions, an operating systems clipboard CUT or COPY or PASTE operations, or a user interface DRAG and DROP interaction. In addition, this exchange may take place using a wired or wireless network transport; such as LAN or WAN protocols, switched telephone circuits, IrDA-based infra-red “beaming” of data, or emerging cellular data services. In any of these example cases, the vCalendar format is intended to be a transport- and platform-independent format for exchanging calendaring and scheduling personal data.

1.3 Contents

This specification is separated into eight sections:

- “Section 1 : Introduction” introduces PDI and the vCalendar specification with an overview, scope statement and section on definitions and abbreviations.
- “Section 2 : vCalendar” defines the semantics and syntax for a clear-text encoding of the vCalendar.
- “Section 3: Internet Recommendations” specifies a set of guidelines to facilitate the exchange of vCalendar 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 vCalendar 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 vCalendar specification.
- “Section 6 : Extended Recurrence Rule Grammar” defines reference information on an extended recurrence rule grammar, copied from the XAPIA CSA Specification.

1.4 Definitions and Abbreviations

Definitions and abbreviations used within this specification follow.

API: Application Programming Interface

Electronic Calendar: Also know as vCalendar.

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.

vCalendar: The generic term for an electronic, virtual collection of calendaring and scheduling information 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 vCalendar are used depends on the applications developed utilizing a vCalendar.

WAN: Wide-Area Network

Section 2 : vCalendar

This section defines the semantics and syntax for encoding the vCalendar in a simple, clear-text encoding.

2.1 Encoding Characteristics

The following characteristics are specific to this encoding.

2.1.1 vCalendar Object

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

```
BEGIN:VCALENDAR
```

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

The vCalendar object is terminated by the appearance of the End vCalendar Delimiter as the first characters on a line:

```
END:VCALENDAR
```

The vCalendar object is a container for calendaring and scheduling entities. These can include either event or todo entities.

2.1.1.1 vEvent Object

A vEvent is a grouping of calendaring and scheduling properties that define an entity that represents a scheduled amount of time on a calendar. For example, it may be an activity; such as a one-hour, department meeting from 8 AM to 9 AM, tomorrow.

An individual vEvent entity is identified within a vCalendar object by the appearance of the delimiter:

```
BEGIN:VEVENT
```

The sentinel string must appear as the first characters on a line.

The vEvent entity is terminated with the appearance of the following delimiter string as the first characters on a line

```
END:VEVENT
```

The vEvent entity can not be nested within another vEvent or vTodo entity. If vEvent entities need to be related to each other or to a vTodo entity, they can specify relationship with the RELATED-TO property.

2.1.1.2 vTodo Object

A vTodo is a grouping of calendaring and scheduling properties that define an entity that represents an action-item or assignment. For example, it may be an item of work assigned to an individual; such as “turn in travel expense today”.

An individual vTodo entity is identified within a vCalendar object by the appearance of the delimiter:

```
BEGIN:VTODO
```

The sentinel string must appear as the first characters on a line.

The vTodo entity is terminated with the appearance of the following delimiter string as the first characters on a line

```
END:VTODO
```

The vTodo entity can not be nested within another vTodo or vEvent entity. If vTodo entities need to be related to each other or to a vEvent entity, they can specify a relationship with the RELATED-TO property.

2.1.2 Property

A *property* is the definition of an individual attribute describing the event or todo associated with the vCalendar. A property takes the following form:

PropertyName [‘;‘ PropertyParameters] ’:‘ PropertyValue

as shown in the following example:

```
DTSTART:19960415T083000
```

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 must appear as the first characters on a line. In the previous example, “DTSTART” is the name of the Start Date/Time property. Property values are specified as strings. In the previous example, “19960415T083000” is the ISO 8601 formatted value for the Start Date/Time 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:

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

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

```
DESCRIPTION;QUOTED-PRINTABLE:Don’t forget to order Girl=
  Scout cookies from Stacey today!  //Full property parameter expression
```

2.1.3 Delimiters

Individual lines within the vCalendar 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:

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

Can be represented as:

```
DESCRIPTION: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 specified 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 DESCRIPTION value of:

```
Project XYZ Final Review
Conference Room - 3B
Come Prepared.
```

Would be represented in a Quoted-Printable encoding as:

```
DESCRIPTION;ENCODING=QUOTED-PRINTABLE:Project XYZ Final Review=0D=0A=
   Conference Room - 3B=0D=0A=
   Come Prepared.
```

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

2.1.4 Encodings

The default encoding for the vCalendar 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 “8-bit”. 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 vCalendar data stream that has been defined using a default encoding other than 7-bit (e.g., 8-bit).

2.1.5 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:

```
DESCRIPTION;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 vCalendar data stream that has been defined using a default character set other than ASCII (e.g., UTF-8).

2.1.6 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:

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

2.1.7 Date and Time

The date and time values for all vCalendar properties are formatted as a string consistent with the ISO 8601 representation for combinations of dates and times. Either the basic or extended format is allowed. The use of UTC, rather than local time, should be used when ever possible in order to avoid time zone ambiguities. The format for the complete, basic representation of a date and time value is written in the following sequence of characters:

```
<year><month><day>T<hour><minute<second><type designator>
```

For example, 8:30 AM on April 15, 1996 local time would be written as:

```
19960415T083000
```

And the same time in UTC based time would be written as:

```
19960415T083000Z
```

Where a value needs to specify a sequence of date and time values, then the property value is a string made up of a list of date and time values, separated by the field separator. For example:

```
19960101T090000Z; 19960201T090000Z; 19960301T090000Z; 19960401T090000Z; ...
```

2.1.8 Time Duration

The values for time duration or periods of time for all vCalendar properties are formatted as a string conformant with the ISO 8601 basic representation for duration of time. A given duration of a period of time is represented by a character string consisting of the designator "**P**", optionally including the number of years followed by the designator "**Y**", optionally including the number of months followed by the designator "**M**", optionally including the number of weeks followed by the designator "**W**", optionally including the number of days followed by the designator "**D**". The sequence can also contain a time component preceded by the designator "**T**", optionally including the number of hours followed by the designator "**H**", optionally including the number of minutes followed by the designator "**M**", optionally including the number of seconds followed by the designator "**S**". For example:

```
P6W
```

A period of six weeks;

```
PT15M
```

A period of 15 minutes;

```
PT1H30M
```

A period of 1 hour and thirty minutes; or

```
P2Y10M15DT10H30M20S
```

A period of 2 years, 10 months, 15 days, 10 hours, 30 minutes, and 20 seconds.

2.1.9 Value Location

The default location of the property values 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 vCalendar 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 shorthand. 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 using 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, Status, Priority, Mail Reminder, etc. properties. The following specifies a value not located inline with the vCalendar but out in the Internet:

```
ATTACH;VALUE=URL:http://www.abc.com/dir_photos/my_photo.gif
```

2.1.10 Binary Values

The vCalendar specification supports inclusion of binary information, such as computer graphic images (e.g., JPEG), digital audio (e.g., WAVE), or video graphic images (e.g., MPEG). The binary information can either be referenced with a Uniform Reference Locator (URL), referenced with a message using a MIME Content-ID of the MIME part that contains the content, or placed inline in the vCalendar as the value of a property. Inline binary information is included as a property value after being character encoded using Base 64 (default) or Quoted-Printable encoding.

2.1.11 Basic Recurrence Rule Grammar

The specification of recurring events can be simplified by the use of a grammar or rule notation. This specification makes use of the Base Recurrence Rule Grammar from the XAPIA's CSA Specification.

A recurrence rule is a string or clear-text encoding of a recurrence specification. A recurrence rule is composed of several components. A rule begins with a **frequency** which describes the type of repeating event (e.g., daily, weekly, etc.). This is followed by an **interval** which indicates how often the frequency repeats (i.e., daily, every third day, etc.). This can be followed by optional **frequency modifier** information and either an **end date** or a **duration**.

Below is the form of a typical rule. This example causes events to be generated every other week on Tuesday and Thursday, for 8 occurrences:

```
W2 TU TH #4
```

Where, **`W`** is the **Frequency**, **`2`** is the **Interval**, **`TU`** and **`TH`** are the optional **Frequency Modifiers**, and **`#4`** is the **`Duration`**.

The basic recurrence rule grammar supports six types of repetition. The six types follow the same form with only the frequency name and optional modifier information changing from one type of frequency to the next.

2.1.11.1 Daily Rule

The daily rule is used for specifying repeating events based on an interval of a day or more. These can range from every day to every 200th day and beyond. The daily rule begins with the letter **`D`** followed by an interval (representing days) and an optional duration or end date.

Some examples follow:

```
Daily for 10 occurrences:
  D1 #10

Daily until 12/24/94:
  D1 19941224T000000Z

Every other day - forever:
```

```
D2 #0
```

```
Every 10 days, 5 occurrences:
  D10 #5
```

2.1.11.2 Weekly Rule

The weekly rule is used for specifying repeating events based on an interval of a week or more. The basic weekly rule has the same form as the daily rule except that the rule begins with a **W** and can contain an optional list of weekdays the events are generated on. For weekly rules, the interval represents weeks. Some examples follow:

```
Weekly for 10 occurrences:
  W1 #10

Weekly until 12/24/94:
  W1 19941224T000000Z

Every other week - forever:
  W2 #0

Weekly on Tuesday and
Thursday for 5 weeks:
  W1 TU TH #5

Every other week on Monday Wednesday and Friday until 12/24/94:
  W2 MO WE FR 19941224T000000Z
```

2.1.11.3 Monthly Rule

The monthly rule is used for specifying repeating events base on an interval of a month or more. There are two types of monthly recurrence rules. One for **by-position** and one for **by-day**. The by-position rule allows weekdays in the month to be specified in relation to their occurrence in the month. An example would be to specify the third Sunday of the month or the last Friday of the month. An occurrence specifier may be used in monthly by-position rules. The occurrence specifiers control which occurrence of a weekday in a month an event occurs on:

```
1+, 2+, ... 5+  for the first occurrence, second, ...fifth occurrence of the
month.

1-, 2-, ... 5-  for the last occurrence, second to last occurrence, etc.
```

A **2+ FR SA** would indicate the second occurrence of Friday and Saturday in the month. A **1- MO** would indicate the first occurrence of Monday working from the end of the month backwards (i.e., the last occurrence). A **2- MO** would be the second to the last Monday of the month.

A by-day rule allows actual day numbers to be specified such as the 12th day or 29th day.

The by-position rule begins with a **MP** and the by-day rule begins with a **MD**. The interval in monthly rules represents months. Some examples follow:

```
Monthly on the 1st Friday for ten occurrences:
  MP1 1+ FR #10

Monthly on the 1st Friday until 12/24/94:
  MP1 1+ FR 19941224T000000Z

Every other month on the 1st and last
```

```
Sunday of the month for 10 occurrences:
  MP2 1+ SU 1- SU #10

Every six months on the 2nd Monday
thru Friday for 10 occurrences:
  MP6 2+ MO TU WE TH FR #10

Monthly on the second last Monday of the month for 6 months:

  MP1 2- MO #6

Monthly on the third to the last day of the month, forever:

  MD1 3- #0

Monthly on the 2nd and 15th of the month for 10 occurrences:
  MD1 2 15 #10

In the next example LD refers to "LastDay" in a monthly recurrence rule.
Monthly on the 1st and last day of the month for 10 occurrences:
  MD1 1 LD #10 or MD1 1 1- #10

Every 18 months on the 10th thru 15th of the month for 10 occurrences:
  MD18 10 11 12 13 14 15 #10

Monthly on the second to the last day for 5 months. So, if the start date is
August 1996, the event would repeat on 8/30/96, 9/29/96, 10/30/96, 11/29/96,
and 12/30/96:
  MD1 2- #5
```

2.1.11.4 Yearly Rule

The yearly rule is used for specifying repeating events based on an interval of a year or more. There are two types of yearly recurrence rules. One for **by-month** and one for **by-day**. The by-month rule allows specific months out of the year to be specified. The by-day allows specific days to be specified. In the by-month rule, the day in the month the rule is to occur on is determined from the initial appointment.

The by-month rule begins with a **`YM`** and the by-day rule begins with a **`YD`**. The interval in yearly rules represents years. Some examples follow:

```
Yearly in June and July for 10 occurrences:
  YM1 6 7 #10

Every other year on January, Feb, and March for 10 occurrences:
  YM2 1 2 3 #10

Every 3rd year on the 1st, 100th and 200th day for 10 occurrences:
  YD3 1 100 200 #10
```

2.1.11.5 Grammar

```
{}    0 or more

[]    0 or 1

start ::= <daily> [<enddate>] |

       <weekly> [<enddate>] |

       <monthlybypos> [<enddate>] |

       <monthlybyday> [<enddate>] |

       <yearlybymonth> [<enddate>] |
```

```
       <yearlybyday> [<enddate>]
digit ::= <0|1|2|3|4|5|6|7|8|9>
digits ::= <digit> {<digits>}
enddate   ::= ISO 8601_date_time value(e.g., 19940712T101530Z)
interval  ::= <digits>
duration  ::= #<digits>
lastday   ::= LD
plus      ::= +
minus     ::= -
daynumber     ::= <1-31> [<plus>|<minus>]| <lastday>
daynumberlist ::= daynumber {<daynumberlist>}
month     ::= <1-12>
monthlist ::= <month> {<monthlist>}
day       ::= <1-366>
daylist       ::= <day> {<daylist>}
occurrence    ::= <1-5><plus> | <1-5><minus>
occurrencelist    ::= <occurrence> {<occurrencelist>}
weekday   ::= <SU|MO|TU|WE|TH|FR|SA>
weekdaylist   ::= <weekday> {<weekdaylist>}
daily     ::= D<interval> [<duration>]
weekly        ::= W<interval> [<weekdaylist>] [<duration>]
monthlybypos  ::= MP<interval> [<occurrencelist> <weekdaylist>] [<duration>]
monthlybyday  ::= MD<interval> [<daynumberlist>] [<duration>]
yearlybymonth ::= YM<interval> [<monthlist>] [<duration>]
yearlybyday   ::= YD<interval> [<daylist>] [<duration>]
```

2.1.11.6 Glossary

enddate	Controls when a repeating event terminates. The enddate is the last time an event can occur.
interval	Defines the frequency in which a rule repeats.
duration	Controls the number of events a rule generates.
lastday	Can be used as a replacement to daynumber to indicate the last day of the month.
daynumber	A number representing a day of the month.
month	A number representing a month of the year.
day	A number representing a day of the year.
occurrence	Controls which week of the month a particular weekday event occurs.
weekday	A symbol representing a day of the week.

daily	Defines a rule that repeats on a daily basis.
weekly	Defines a rule that repeats on a weekly basis.
monthlybypos	Defines a rule that repeats on a monthly basis on a relative day and week.
monthlybyday	Defines a rule that repeats on a monthly basis on an absolute day.
yearlybymonth	Defines a rule that repeats on specific months of the year.
yearlybyday	Defines a rule that repeats on specific days of the year.

2.1.11.7 Policies

1. The duration portion of a rule defines the total number of events the rule generates, including the first event.
2. Information, not contained in the rule, necessary to determine the next event time and date is derived from the Start Time entry attribute.
3. If an end date and a duration is specified in the rule, the recurring event ceases when the end date is reached or the number of events indicated in the duration occur; whichever comes first.
4. If the duration or an end date is not established in the rule (e.g., `D4`) the event occurs twice. That is `D4` is equivalent to `D4 #2`.
5. A duration of `#0` means repeat this event forever.
6. Using the occurrence specifier `5+` (e.g. 5th Friday) or `5-` (e.g. 5th from last Friday) in a month that does not contain 5 weeks does not generate an event and thus does not count against the duration. The same applies to providing a day of the month that does not occur in the month. For example the 30th or 31st .
7. The start time and date of an entry must be synchronized with one of the repeating events defined by its recurrence rule. The following is not allowed:

   Initial Appt Date: 7/1/94 (Friday)
   Recurrence Rule: `W1 MO TH #5`

   The following is acceptable:

   Initial Appt Date: 7/1/94 (Friday)
   Recurrence Rule: `W1 MO FR #5` or `W1 #5`
8. If the optional **<occurrencelist>** and **<weekdaylist>** information is missing from a **<monthlybypos>** occurrence the information is derived from the entry attributes. The **<occurrence>** used in the recurring event is a count from the beginning of the month to the entry date and the **<weekday>** used is the day of the week the entry is scheduled to occur on.

   If the **<monthlybypos>** occurrence or **<monthlybyday>** occurrence does not list a week day (e.g., `SU or day 10`) in the rule, the week day is established from the entry attribute information. As an example the rule **`MP1 #3`** used in an entry with a start date of 7/20/94 (which is the third Wednesday of the month) repeats on 8/17/94 which is the third Wednesday of the month.

2.2 vCalendar Properties

The following properties must appear after the **BEGIN:VCALENDAR** delimiter but before the occurrence of the **BEGIN:VEVENT** or **BEGIN:VTODO** delimiters. These properties apply to the vCalendar object as a whole; unless overridden by a property within the scope of an event or todo entity.

2.2.1 Daylight Savings Rule

This property is identified by the property name **DAYLIGHT**. This property defines the daylight savings time rule observed by the "home" calendar system that created the vCalendar entity.

Many locations adjust their standard time forward or backward by one hour, in order to accommodate seasonal changes in number of daylight hours. Standard time is also known as Winter Time. Daylight savings time is also known as Advanced Time, Summer Time, or Legal Time in certain countries.

The property value consists of a sequence of components that define the daylight savings time rule. The value consists of the daylight savings time flag, followed by the daylight savings time offset, followed by the date and time that the daylight savings time begins, followed by the date and time that the daylight savings time ends, followed by the standard time designation, followed by the daylight savings time designation. The daylight savings time flag is TRUE if daylight savings time is observed, otherwise it is FALSE and no other components are specified. The daylight savings time offset value 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. The date and time that the daylight savings time begins and ends is specified in a manner consistent with ISO 8601 date and time format. The standard time and daylight savings time designations correspond to the customary character designations.

The following are examples of this property:

```
DAYLIGHT:TRUE;-06;19960407T025959;19961027T010000;EST;EDT

DAYLIGHT:FALSE

DAYLIGHT:TRUE;-09;19960407T115959;19961027T100000;PST;PDT
```

Support for this property is optional for implementations conforming to this specification.

2.2.2 Geographic Position

This property is identified by the property name **GEO**. This property specifies information related to the global position of the "home"system that created the vCalendar object. The property value specifies longitude and latitude. The longitude represents the location east and west of the prime meridian as a positive or negative real number, respectively. The latitude represents the location north and south of the equator as a positive or negative real number, respectively. The following is an example of this property:

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

Support for this property is optional for implementations conforming to this specification.

2.2.3 Product Identifier

This property is identified by the property name **PRODID**. This property specifies the identifier for the product that created the vCalendar object. The vendor of the implementation should assure that this is a globally unique identifier; using some technique such as an ISO 9070 FPI value. The following is an example of this property:

```
PRODID:-//ABC Corporation//NONSGML My Product//EN
```

Support for this property is optional for implementations conforming to this specification.

2.2.4 Time Zone

This property is identified by the property name **TZ**. This property specifies the standard time zone of the "home" system that created the vCalendar object. The property value 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. The following are examples of this property:

```
TZ:-05
TZ:+05:30
```

Support for this property is optional for implementations conforming to this specification.

2.2.5 Version

This property specifies the identifier corresponding to the highest version number of the vCalendar Specification supported by the implementation that created the vCalendar object. The value of this property must be 1.0 to correspond to this specification..

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

```
VERSION:1.0
```

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

2.3 vEvent and vTodo Properties

The following properties may appear within an event or todo calendaring and scheduling entity.

2.3.1 Attachment

This property is identified by the property name **ATTACH**. The property defines an attached object to the vCalendar entity. For example, a document to be reviewed at a scheduled event or the process steps for a todo. The property value can be a text string, a reference to another message body part or a reference to a URL based document.

Multiple attachments may be specified by including multiple ATTACH properties within the vCalendar entity.

The following are examples of this property:

```
ATTACH;VALUE=CONTENT-ID:<jsmith.part3.960817T083000.xyzMail@host1.com>
ATTACH;VALUE=URL:file://xyzCorp.com/pub/reports/r-960812.ps
```

Support for this property is optional for implementations conforming to this specification.

2.3.2 Attendee

This property is identified by the property name **ATTENDEE**. The property defines an attendee to a group event or todo. The default property value is an (RFC 822) address. The property may include property parameters **ROLE,** for the role of the attendee in the event or todo; **STATUS**, for the status of the attendee's participation in the event or todo, **RSVP**, for indicating whether the favor of a reply is requested, and **EXPECT**, to indicate the expectation of the attendee's participation by the originator.

Multiple attendees may be specified by including multiple ATTENDEE properties within the vCalendar entity.

The property value may reference a vCard object. This provides a useful mechanism to allow more than just the address of the attendee to be referenced.

The ROLE property parameter for each attendee can have the following values:

Description	Property Value
Indicates an attendee at the event or todo	ATTENDEE
Indicates organizer of the event, but not owner	ORGANIZER
Indicates owner of the event or todo.	OWNER
Indicates a delegate of another attendee.	DELEGATE

The default value for this property parameter is ATTENDEE.

The STATUS property parameter for each attendee can have the following values:

Description	Property Value
Indicates todo was accepted by attendee	ACCEPTED
Indicates event or todo requires action by attendee	NEEDS ACTION
Indicates event or todo was sent out to attendee	SENT
Indicates event is tentatively accepted by attendee	TENTATIVE
Indicates attendee has confirmed their attendance at the event	CONFIRMED
Indicates event or todo has been rejected by attendee	DECLINED
Indicates todo has been completed by attendee	COMPLETED
Indicates event or todo has been delegated by the attendee to another	DELEGATED

The default value for this property parameter is NEEDS ACTION.

The RSVP property parameter for each attendee can have the following values:

Description	Property Value
Indicates a reply is requested	YES
Indicates a reply is not requested.	NO

The default value for this property parameter is NO.

The EXPECT property parameter for each attendee can have the following values:

Description	Property Value
Indicates request is for your information.	FYI
Indicates presence is definitely required.	REQUIRE
Indicates presence is being requested	REQUEST
Indicates an immediate response needed.	IMMEDIATE

The default value for this property parameter is FYI.

The following is an example of this property's use for a todo:

```
ATTENDEE;ROLE=OWNER;STATUS=COMPLETED:jsmith@host1.com
```

The following is an example of this property used for specifying multiple attendees to an event:

```
ATTENDEE;ROLE=OWNER;STATUS=CONFIRMED:John Smith <jsmith@host1.com>
ATTENDEE;ROLE=ATTENDEE;STATUS=TENTATIVE:Henry Cabot <hcabot@host2.com>
ATTENDEE;ROLE=DELEGATE;STATUS=CONFIRMED:Jane Doe <jdoe@host1.com>
```

The following is an example of this property with the value specified as an URL reference to a vCard that contains the information about the attendee:

```
ATTENDEE;ROLE=ATTENDEE;STATUS=CONFIRMED;VALUE=URL;TYPE=VCARD:
   http://www.xyz.com/~myvcard.vcf
```

Support for this property is optional for implementations conforming to this specification.

2.3.3 Audio Reminder

This property is identified by the property name **AALARM**. The property defines an audio reminder for the vCalendar entity. An audio reminder is an alarm that is sounded for the event.

The value for the audio reminder consists of the **Run Time**, or the date and time that the reminder is to be executed; **Snooze Time**, or the duration of time after the Run Time that the reminder is to be dormant prior to being repeated; **Repeat Count**, or the number of times that the reminder is to be repeated; and the **Audio Content**, or the digital sound to be played when the reminder is executed.

The following are some examples of this property:

```
AALARM;TYPE=WAVE;VALUE=URL:19960415T235959; ; ; file:///mmedia/taps.wav

AALARM;TYPE=WAVE;VALUE=CONTENT-ID:19960903T060000;PT15M;4;<jsmith.part2.=
  960901T083000.xyzMail@host1.com>
```

The property has the following additional property parameters:

Description	Property Parameter Values
TYPE	
Indicates the MIME basic audio content type.	PCM
Indicates the WAVE format for audio content.	WAVE
Indicates the AIFF format for audio content.	AIFF

The Reminder properties are primarily provided as a means for allowing the capture of alarm information when accessing a calendar system. It may not be an appropriate property to send in an event or todo request.

Support for this property is optional for implementations conforming to this specification.

2.3.4 Categories

This property is identified by the property name **CATEGORIES**. This property defines the categories for the vCalendar entity. More than one category may be specified as a list of categories separated by the Semi-Colon character (ASCII decimal 59).

The following are some examples of this property:

```
CATEGORIES:APPOINTMENT;EDUCATION
CATEGORIES:MEETING
```

Some of the possible values for this property might include the following:

Some Possible Property Values
APPOINTMENT
BUSINESS
EDUCATION
HOLIDAY
MEETING
MISCELLANEOUS
PERSONAL
PHONE CALL
SICK DAY
SPECIAL OCCASION
TRAVEL
VACATION

Support for this property is mandatory for implementations conforming to this specification.

2.3.5 Classification

This property is identified by the property name **CLASS**. This property defines the access classification for the vCalendar entity.

A calendar entity access classification is only one component of the general security system within a calendar application. It provides a method of capturing the scope of the access the calendar owner intends for information within an individual calendar entry. The access classification of an individual vCalendar entity is useful when measured along with the other security components of a calendar system (e.g., user authorization, access rights, access role, etc.). Hence, the semantics of the individual access classifications can not be completely defined by this specification. Additionally, due to the “blind” nature of most exchange processes using this specification, these entity classifications can not serve as an enforcement statement for a system receiving a vCalendar data stream. Rather, they provide a method for capturing the intention of the calendar owner for the access to the calendar entry.

The following is an example of this property:

```
CLASS:PUBLIC
```

The property can have the following values:

Description	Property Value
Indicates general, public access.	PUBLIC
Indicates restricted, private access.	PRIVATE
Indicates very restricted, confidential access.	CONFIDENTIAL

The default value for this property is PUBLIC.

Support for this property is optional for implementations conforming to this specification.

2.3.6 Date/Time Created

This property is identified by the property name **DCREATED**. This property specifies the date and time that the vCalendar entity was created within the originating calendar system. This is not generally the same date and time that the vCalendar object was created. The date and time value is the local or UTC based time expressed in the complete representation, basic format as specified in ISO 8601. The following is example of this property:

```
DCREATED:19960329T083000
```

Support for this property is optional for implementations conforming to this specification.

2.3.7 Date/Time Completed

This property is identified by the property name **COMPLETED**. This property defines the date and time that the todo was actually completed. The date and time value is expressed in the complete representation, basic format as specified in ISO 8601. The time can either be in local or UTC based time. The following is an example of this property:

```
COMPLETED:19960401T235959
```

Support for this property is mandatory for implementations conforming to this specification.

2.3.8 Description

This property is identified by the property name **DESCRIPTION**. This property provides a more complete description of the vCalendar entity, than that provided by the SUMMARY property. The following is an examples of the property with formatted line breaks in the property value:

```
DESCRIPTION;ENCODING=QUOTED-PRINTABLE:Meeting to provide technical=
   review for “Phoenix” design. =0D=0A=
   Happy Face Conference Room. Phoenix design team=
   must attend this meeting. RSVP to team leader.
```

The following is an examples of the property with folding of long lines:

```
DESCRIPTION:Last draft of the new novel is to be completed
   for the editor’s proof today.
```

Support for this property is mandatory for implementations conforming to this specification.

2.3.9 Display Reminder

This property is identified by the property name **DALARM**. The property defines a display reminder for the vCalendar entity. A display reminder is an alarm that is popped up into the user interface or otherwise visually displayed for the event.

The value for the display reminder consists of the **Run Time**, or the date and time that the reminder is to be executed; **Snooze Time**, or the duration of time after the Run Time that the reminder is to be dormant prior to being repeated; **Repeat Count**, or the number of times that the reminder is to be repeated; and the **Display String**, or the text to be displayed when the reminder is executed.

The following is an example of this property:

```
DALARM:19960415T235000;PT5M;2;Your Taxes Are Due !!!
```

The Reminder properties are primarily provided as a means for allowing the capture of alarm information when accessing a calendar system. It may not be an appropriate property to send in an event or todo request.

Support for this property is optional for implementations conforming to this specification.

2.3.10 Due Date/Time

This property is identified by the property name **DUE**. This property defines the date and time that the todo is due to be completed. The date and time value is expressed in the complete representation, basic format as specified in ISO 8601. The time can either be in local or UTC based time. The following is an example of this property:

```
DUE:19960401T235959Z
```

Support for this property is mandatory for implementations conforming to this specification.

2.3.11 End Date/Time

This property is identified by the property name **DTEND**. This property defines the date and time that the event will end. The date and time value is expressed in the complete representation, basic format as specified in ISO 8601. The time can either be in local or UTC based time. Events may have an end date/time but no start date/time. In that case, the event does not take up any time. The following is an example of this property:

```
DTEND:19960401T235959Z
```

Support for this property is mandatory for implementations conforming to this specification.

2.3.12 Exception Date/Times

This property is identified by the property name **EXDATE**. This property defines the list of date/time exceptions for a recurring vCalendar entity. The date and time values is expressed in the complete representation, basic format as specified in ISO 8601. The times can either be in local or UTC based time. The number of date/time exceptions is specified by the Number Exceptions property. The following is an example of this property:

```
EXDATE:19960402T010000Z;19960403T010000Z;19960404T010000Z
```

Support for this property is optional for implementations conforming to this specification.

2.3.13 Exception Rule

This property is identified by the property name **EXRULE**. This property defines a rule or repeating pattern for an exception to a recurring vCalendar entity, based on the Basic Recurrence Rule Grammar of XAPIA's CSA. The value for the property is a pattern specification for the recurrence exception. The following are some examples of this property:

```
XRULE:W2 TU TH #2        // Except every other week, on Tuesday and Thursday
                         // for 4 occurrences
XRULE:D1 #10             // Except daily for 10 occurrences
XRULE:YM1 6 7 #8         // Except yearly in June and July for 8 occurrences
```

Support for this property is optional for implementations conforming to this specification.

2.3.14 Last Modified

This property is identified by the property name **LAST-MODIFIED**. The property specifies the date and time that the vCalendar entity was last revised. The following is an example of this property:

```
LAST-MODIFIED:19960817T133000Z
```

Support for this property is optional for implementations conforming to this specification.

2.3.15 Location

This property is identified by the property name **LOCATION**. The property defines the intended location for a vCalendar entity.

The property value may reference a vCard object. This provides a useful mechanism to specify a location in terms of its electronic business card.

The following are some examples of this property:

```
LOCATION:Conference Room - F123, Bldg. 002     // or
LOCATION;VALUE=URL;TYPE=VCARD:http://www.xyzcorp.com/~jsmith.vcf
```

Support for this property is optional for implementations conforming to this specification.

2.3.16 Mail Reminder

This property is identified by the property name **MALARM**. The property defines an email address that is to be sent a reminder for the vCalendar entity. A mail reminder is an electronic mail address that will be sent a display string as an alarm for the event.

The value for the procedure reminder consists of the **Run Time**, or the date and time that the reminder is to be executed; **Snooze Time**, or the duration of time after the Run Time that the reminder is to be dormant prior to being repeated; **Repeat Count**, or the number of times that the reminder is to be repeated; **Email Address**, or the (RFC 822) email address that is to be sent the reminder and the **Note**, or the textual reminder string that is to be sent to the email address.

The following is an example of this property:

```
MALARM:19960416T000000;PT1H;24;IRS@us.gov;The Check Is In The Mail!
```

The Reminder properties are primarily provided as a means for allowing the capture of alarm information when accessing a calendar system. It may not be an appropriate property to send in an event or todo request.

Support for this property is optional for implementations conforming to this specification.

2.3.17 Number Recurrences

This property is identified by the property name **RNUM**. The property defines the number of times the calendar entry will reoccur. The value is equal to the number of recurrences that are specified by the union of the Recurrence Dates, Recurrence Rule, Exception Dates, and Exception Rule property values. The following is an example of this property:

```
RNUM:3
```

In the event that this value does not match the computed number of recurrences, it will be ignored and the computer number of recurrences will be used.

Support for this property is optional for implementation conforming to this specification.

2.3.18 Priority

This property is identified by the property name **PRIORITY**. The property defines the priority for the vCalendar entity. The value is an alphanumeric. A value of zero (ASCII decimal 48) specifies an undefined priority. A value of one is the highest priority. A value of two is the second highest priority. Subsequent numbers specify a decreasing ordinal priority. The following is an example of this property:

```
PRIORITY:2
```

Support for this property is mandatory for implementation conforming to this specification.

2.3.19 Procedure Reminder

This property is identified by the property name **PALARM**. The property defines a procedure reminder for the vCalendar entity. A procedure reminder is a procedure, or application executable that will be run as an alarm for the event.

While this property has many useful purposes, implementors should be aware of the security implications of sending a vCalendar data stream containing this property. The security implications are similar to those associated with active messages within electronic mail.

The value for the procedure reminder consists of the **Run Time**, or the date and time that the reminder is to be executed; **Snooze Time**, or the duration of time after the Run Time that the reminder is to be dormant prior to being repeated; **Repeat Count**, or the number of times that the reminder is to be repeated; and the **Procedure Name**, or the path to the procedure to be run when the reminder is executed.

The following is an example of this property:

```
PALARM;VALUE=URL:19960415T235000;PT5M;2;file:///myapps/shockme.exe
```

The Reminder properties are primarily provided as a means for allowing the capture of alarm information when accessing a calendar system. It may not be an appropriate property to send in an event or todo request.

Support for this property is optional for implementations conforming to this specification.

2.3.20 Related To

This property is identified by the property name **RELATED-TO**. The property is used to represent relationships or references between this vCalendar entity and another. The property value consists of the persistent, globally unique identifier of another vCalendar entity. This value would be represented in a vCalendar data stream by the UID property.

A linked relationship can be specified by a series of entities that each, in turn, refer to their parent entity. A group relationship can be specified by a number of entities that all refer to one common parent entity.

Changes to a calendar entity referenced by this property may impact the related calendar entity. For example, if a group event changes it start or end date or time, then the related, dependent events will need to have their start and end dates changed in a corresponding way. This property is intended only to provide information on the relationship of calendar entities. It is up to the target calendar system to maintain this relationship.

The following is an example of this property:

```
RELATED-TO:<jsmith.part7.19960817T083000.xyzMail@host3.com>

RELATED-TO:19960401-080045-4000F192713-0052
```

Support for this property is optional for implementations conforming to this specification.

2.3.21 Recurrence Date/Times

This property is identified by the property name **RDATE**. This property defines the list of date/times for a recurring vCalendar entity. The date and time values is expressed in the complete representation, basic format as specified in ISO 8601. The times can either be in local or UTC based time. The number of recurring date/times is specified by the Number Recurrences property. The following is an example of this property:

```
RDATE:19960402T010000Z;19960403T010000Z;19960404T010000Z
```

Support for this property is optional for implementations conforming to this specification.

2.3.22 Recurrence Rule

This property is identified by the property name **RRULE**. This property defines a rule or repeating pattern for a recurring vCalendar entity, based on the Basic Recurrence Rule Grammar of XAPIA's CSA. The value for the property is a pattern specification for the recurrence. The following is an example of this property:

```
RRULE:W2 TU TH          // Every other week, on Tuesday and Thursday

RRULE:D1 #10            // Daily for 10 occurrences

RRULE:YM1 6 7 #8        // Yearly in June and July for 8 occurrences
```

Support for this property is optional for implementations conforming to this specification.

2.3.23 Resources

This property is identified by the property name **RESOURCES**. This property defines the equipment or resources needed in the vCalendar event.

Some of the values that the property may have include the following:

Some Possible Property Values
CATERING
CHAIRS
COMPUTER PROJECTOR
EASEL
OVERHEAD PROJECTOR
SPEAKER PHONE
TABLE
TV
VCR
VIDEO PHONE
VEHICLE

The following is an example of this property:

```
RESOURCES:EASEL;PROJECTOR;VCR
```

Support for this property is optional for implementations conforming to this specification.

2.3.24 Sequence Number

This property is identified by the property name **SEQUENCE**. This property defines the instance of the vCalendar entity in a sequence of revisions. When a vCalendar entity is created its sequence number is zero (ASCII decimal 48). It is incremented each time it is revised. The following is an example of this property:

```
SEQUENCE:1
```

Support for this property is optional for implementations conforming to this specification.

2.3.25 Start Date/Time

This property is identified by the property name **DTSTART**. This property defines the date and time that the event will start. The date and time value is expressed in the complete representation, basic format as specified in ISO 8601. The time can either be in local or UTC based time. Events may have a start date/time but no end date/time. In that case, the event does not take up any time. The following is an example of this property:

```
DTSTART:19960401T235959
```

Support for this property is mandatory for implementations conforming to this specification.

2.3.26 Status

This property is identified by the property name **STATUS**. This property defines the status associated with the vCalendar entity. This property can be used when the ATTENDEE property is either not supported or not needed. The following is an example of this property:

```
STATUS:TENTATIVE
```

The property can have the following values:

Description	Property Value
Indicates todo was accepted	ACCEPTED
Indicates event or todo requires action	NEEDS ACTION
Indicates event or todo was sent out.	SENT
Indicates event is tentatively accepted	TENTATIVE
Indicates event is confirmed	CONFIRMED
Indicates event or todo has been declined	DECLINED
Indicates todo has been completed	COMPLETED
Indicates event or todo has been delegated	DELEGATED

The default value for this property is NEEDS ACTION.

Support for this property is mandatory for implementations conforming to this specification.

2.3.27 Summary

This property is identified by the property name **SUMMARY**. This property defines a short summary or subject of the vCalendar entity. The following is an example of this property:

```
SUMMARY:Department Party
```

Support for this property is mandatory for implementations conforming to this specification.

2.3.28 Time Transparency

This property is identified by the property name **TRANSP**. This property defines whether the event is transparent to free time searches. The value of this property is a number. A value of zero (ASCII decimal 48) guaranttes that the entry will blocks time and will be factored into a free time search. A value of one (ASCII decimal 49) specifies that the entry will not block time and will not be factored into a free time search. Any values greater than “1” will provide implementation specific transparency semantics. Some implementations may treat values greater than one as non-blocking or transparent events. Other implementations may use the numeric value to provide a layering of levels of transparency. The default value is zero (ASCII decimal 48), the event is not transparent and will block free time searches. The following is an example of this property:

```
TRANSP:0
```

Support for this property is optional for implementations conforming to this specification.

2.3.29 Uniform Resource Locator

This property is identified by the property name **URL**. This property defines a Uniform Resource Locator for an Internet location that can be used to obtain real-time information associated with the vCalendar entity. 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/calendars/jsmith/mytime.or3
```

Support for this property is optional for implementations conforming to this specification.

2.3.30 Unique Identifier

This property is identified by the property name **UID**. This property defines a persistent, globally unique identifier associated with the vCalendar entity. Some examples of 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 can render an updated version of the vCalendar for the object. The following is an example of this property:

```
UID:19960401-080045-4000F192713-0052
```

This property is an important method for group scheduling applications to match calendar entities with later modification or deletion requests. Calendaring applications that do not generate this property in vCalendar entities may be limiting their interoperability with other group scheduling applications.

Support for this property is optional for implementations conforming to this specification.

2.4 Miscellaneous Properties

2.4.1 Extensions

The clear-text encoding 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. All vCalendar Readers are expected to be able to interpret the extension properties and property parameters but may ignore them. The following might be the ABC vendor's extension for an audio-clip form of subject property:

```
X-ABC-MMSUBJ;TYPE=WAV; VALUE=URL: http://load.noise.org/mysubj.wav
```

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 vCalendar data streams with extensions. The implementation need not act on them.

2.5 Formal Definition

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

```
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

value     = 7bit / 8bit / quoted-printable / base64
   ; The value must be in the encoding type specified for the property value.
```

```
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

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

vcal_file = [wsls] vcal [wsls]

vcal      = "BEGIN" [ws] ":" [ws] "VCALENDAR" [ws] 1*CRLF
      calprop calentities [ws] *CRLF
      "END" [ws] ":" [ws] "VCALENDAR" [ws] 1*CRLF

calentities   = calentities *CRLF calentity
      / calentity

calentity = evententity
      / todoentity

evententity   = "BEGIN" [ws] ":" [ws] "EVENT" [ws] 1*CRLF
      entprops [ws] *CRLF
      "END" [ws] ":" [ws] "EVENT" [ws] 1*CRLF

todoentity    = "BEGIN" [ws] ":" [ws] "TODO" [ws] 1*CRLF
      entprops [ws] *CRLF
      "END" [ws] ":" [ws] "TODO" [ws] 1*CRLF

calprops  = calprops *CRLF calprop
      / calprop

calprop   = "DAYLIGHT"
        [params] ":" value CRLF
      / "GEO"
```

```
        [params] ":" value CRLF
      / "PRODID"
        [params] ":" value CRLF
      / "TZ"
        [params] ":" value CRLF
      / "VERSION"
        [params] ":" "1.0" CRLF
  ; The VERSION calendar property MUST appear in the vCalendar object.

entprops  = entprops *CRLF entprop
      / entprop

entprop   = [ws] simprop
        [params] ":" value CRLF
      / [ws] "AALARM"
        [params] ":" aalarmparts CRLF
      / [ws] "CATEGORIES"
        [params] ":" 1*catvals CRLF
      / [ws] "CLASS"
        [params] ":" classvals CRLF
      / [ws] "DALARM"
        [params] ":" dalarmparts CRLF
      / [ws] "EXDATE"
        [params] ":" xdatevals CRLF
      / [ws] "MALARM"
        [params] ":" malarmparts CRLF
      / [ws] "PALARM"
        [params] ":" palarmparts CRLF
      / [ws] "RDATE"
        [params] ":" rdatevals CRLF
      / [ws] "RESOURCES"
        [params] ":" 1*resvals CRLF
      / [ws] "STATUS"
        [params] ":" statvals CRLF

simprop   = "ATTACH" / "ATTENDEE" / "DCREATED" / "COMPLETED"
      / "DESCRIPTION" / "DUE" / "DTEND" / EXRULE / LAST-MODIFIED
      / "LOCATION" / "RNUM" / "PRIORITY" / "RELATED-TO" / "RRULE"
      / "SEQUENCE" / "DTSTART" / "SUMMARY" / "TRANSP" / "URL" / "UID"
```

```
      /"X-" word

aalarmparts  = 0*3(strnosemi ";") strnosemi
  ; runTime, snoozeTime, repeatCount, audioContent

catvals  = "APPOINTMENT" / "BUSINESS" / "EDUCATION" / "HOLIDAY" / "MEETING"
      / "MISCELLANEOUS" / "PERSONAL" / "PHONE CALL" / "SICK DAY"
      / "SPECIAL OCCASION" / "TRAVEL" / "VACATION" / "X-" word / value

classvals = "PUBLIC" / "PRIVATE" / "CONFIDENTIAL" / "X-" word / value

dalarmparts  = 0*3(strnosemi ";") strnosemi
  ; runTime, snoozeTime, repeatCount, displayString

xdatevals = 1*value
  ; One or more date/time values

malarmparts  = 0*4(strnosemi ";") strnosemi
  ; runTime, snoozeTime, repeatCount, addressString, noteString

palarmparts  = 0*3(strnosemi ";") strnosemi
  ; runTime, snoozeTime, repeatCount, procedureName

rdatevals = 1*value
  ; One or more date/time values

resvals  = "CATERING" / "CHAIRS" / "EASEL" / "PROJECTOR" / "VCR"
      / "VEHICLE" / "X-" word / value

statvals  = "ACCEPTED" / "NEEDS ACTION" / "SENT" / "TENTATIVE"
      / "CONFIRMED" / "DECLINED" / "COMPLETED" / "DELEGATED"
      / "X-" word / value

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
      / “ROLE” [ws] “=” [ws] roleval
      / “STATUS” [ws] “=” [ws] statuval
      / “X-” word [ws] “=“ [ws] word
      / knowntype

ptypeval  = knowntype / “X-” word

knowntype = “WAVE” / “PCM” / “VCARD” / “X-“ word / value

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>

roleval   = “ATTENDEE” / “ORGANIZER” / “OWNER” / “X-“ word

statusval = “ACCEPTED” / “NEEDS ACTION” / “SENT” / “TENTATIVE” / “CONFIRMED”
      / “DECLINED” / “COMPLETED” / “DELEGATED” / “X-“ word

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

nonsemi      = <any non-control ASCII except “;”>
```

Section 3 : Internet Recommendations

3.1 Recommended Practice With SMTP/MIME

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

3.1.1 Text/Plain Content Type

Without any change to existing SMTP or MIME compliant user agents, a vCalendar object 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 vCalendars over SMTP services, it does not allow for the end user to take advantage of the full capabilities of either the vCalendar or Internet email (i.e., MIME) functionality.

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

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

Steve:  Thanks for the call earlier today. Let’s get together
tomorrow at 8:30 AM EST to discuss your new proposal. Here is
the meeting notice for your PIM.
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VEVENT
CATEGORIES:MEETING
STATUS:TENTATIVE
DTSTART:19960401T033000Z
DTEND:19960401T043000Z
SUMMARY:Your Proposal Review
DESCRIPTION:Steve and John to review newest proposal material
CLASS:PRIVATE
END:VEVENT
END:VCALENDAR
```

The following example demonstrates how a vCalendar 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 vCalendar Example
```

```
To: john.smith@host.com
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary=vCalendar
Message-ID: <ABC-1.00-Note-martin-steve-0824475754>

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

John:  I can’t make that meeting at 8:30. How about doing it
over lunch at noon?  Here is an action item for the meeting.

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

BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VTODO
SUMMARY:John to pay for lunch
DUE:19960401T083000Z
STATUS:NEEDS ACTION
END:VTODO
END:VCALENDAR


--vCalendar—
```

3.1.2 Text/X-vCalendar Content Type

The vCalendar object can also be passed as a non-standard MIME media type. This would be useful in order to clearly identify the vCalendar object in an electronic mail message body part. A non-standard, vCalendar object should be identified as the MIME type/subtype “text/x-vCalendar”.

The following example demonstrates how a vCalendar containing both an event and a todo can be included as a separate text/x-vCalendar content portion within a MIME user agent.

```
Date: Fri, 26 Jan 1996 07:53:00 -0500
From: smartin@host2.com
Subject: RE: Text/X-vCalendar MIME vCalendar Example
To: john.smith@host.com
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary=vCalendar
Message-ID: <ABC-1.00-Note-martin-steve-0824475754>

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

John:  I can’t make that meeting at 8:30. How about doing it
over lunch at noon?  Here is an event for your PIM. I have
also given you an action item for the meeting.

--vCalendar
Content-Type:text/x-vCalendar; charset=us-ascii; name="MARTIN.VCS"
```

```
BEGIN:VCALENDAR
VERSION:1.0
BEGIN:VEVENT
CATEGORIES:MEETING
STATUS:NEEDS ACTION
DTSTART:19960401T073000Z
DTEND:19960401T083000Z
SUMMARY:Steve’s Proposal Review
DESCRIPTION:Steve and John to review newest proposal material
CLASS:PRIVATE
END:VEVENT
BEGIN:VTODO
SUMMARY:John to pay for lunch
DUE:19960401T083000Z
STATUS:NEEDS ACTION
END:VTODO
END:VCALENDAR


--vCalendar—
```

3.2 Recommended Practice With HTTP/HTML

The vCalendar specification provides a useful format for conveying calendaring and scheduling information between a Web browser and a HTTP server. Homepages can be used as a web-based document for publishing public events. The events can be easily formatted into vCalendar objects for transfer between the server and a requesting browser. The following examples are provided to illustrate possible scenarios where a non-standard “text/x-vCalendar” MIME type/subtype corresponding the vCalendar can be used to transfer calendaring and scheduling information across the World Wide Web.

The following example demonstrates how a vCalendar object can be included in an HTML document or Web page. This may be an initial method for incorporating vCalendar objects into Web pages. This sample assumes that the Web Browser is capable of handling the OBJECT HTML 3.2 element.

```
<html>
<head>
<title>HTTP/Web vCalendar Example</title>
</head>
<body bgcolor="#ffffff">
<h1>Special <i>New</i> Events</h1>
<h3>The latest events to be added to the calendar of activities are:</h3>
<hr>
<center>
<object data=martin.vcs type="text/x-vCalendar">
Your browser does not support OBJECT or the text/x-vCalendar MIME
type/subtype.  Get the events' data <a href=martin.vcs>here</a> and manually
use it.
</object>
</center>
<hr>
<center>
<A HREF="mailto:john.smith@host.com"><img src="bottomtext.gif" border=0></A>
```

```
</center>
<hr>
</body>
</html>
```

The following table demonstrates a simple HTTP transaction between client and server that retrieves a vCalendar object from the Web Server. The entries under Client and Server are the actual HTTP headers and data that might be exchanged.

Client	Direction	Server

`GET martin.vcs HTTP/1.0`
`User-Agent: MyBrowser/1.0`
`Accept: text/html, text/plain, image/gif, image/jpeg, */*`
	->	

	<-	`200 OK`
		`Server: YourServer/1.1`
		`Date: 25 Jan 96 0932 EDT`
		`Content-Type: text/x-vCalendar`
		`Content-Length: 257`
		
		`BEGIN:VCALENDAR`
		`BEGIN:VEVENT`
		`CATEGORIES:MEETING`
		`STATUS:TENTATIVE`
		`DTSTART:19960401T033000Z`
		`DTEND:19960401T043000Z`
		`SUBJECT:Your Proposal Review`
		`DESCRIPTION:Steve and John to review newest proposal material`
		`CLASS:PRIVATE`
		`END:VEVENT`
		`END:VCALENDAR`

The following example illustrates how a “month at a glance” type of information can be displayed on a homepage with the events or todos being links to vCalendar objects.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD><TITLE>HTTP/Web vCalendar Example</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1>Calendar/Events for August</H1>
<HR><CENTER>
<TABLE BORDER=1 >
<TR>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%" BGCOLOR="#0080FF">
<CENTER><P>Monday</P></CENTER></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%" BGCOLOR="#0080FF">
<CENTER><P>Tuesday</P></CENTER></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%" BGCOLOR="#0080FF">
<CENTER><P>Wednesday</P></CENTER></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%" BGCOLOR="#0080FF">
<CENTER><P>Thursday</P></CENTER></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%" BGCOLOR="#0080FF">
<CENTER><P>Friday</P></CENTER></TD>
<TD ALIGN=CENTER VALIGN=TOP WIDTH="17%" BGCOLOR="#0080FF">
<CENTER><P>Saturday - Sunday</P></CENTER></TD>
</TR>

<TR>
```

```
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#C0C0C0">29:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#C0C0C0">30:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#C0C0C0">31:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">1:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">2:</TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">3:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">4:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">5:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#FFFF00">6:
<P><A HREF="080696a.vcs">Meeting w/Martin</A></P></TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">7:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">8:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">9:</TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">10:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">11:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">12:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">13:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">14:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">15:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">16:</TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">17:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">18:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">19:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">20:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#FFFF00">21:
<P><A HREF="082196a.vcs">vCal/vCard Seminar</A></P></TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">22:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">23:</TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">24:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">25:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%">26:</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#FFFF00">27:
<P><A HREF="082796a.vcs">versit Conference</A></P></TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#FFFF00">28:
<P><A HREF="082796a.vcs">versit Conference</A></P>
</TD>
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#FFFF00">29:
<P><A HREF="082796a.vcs">versit Conference</A></P></TD>
```

```
<TD ALIGN=LEFT VALIGN=TOP ROWSPAN="2" WIDTH="17%" BGCOLOR="#FFFF00">30:
<P><A HREF="082796a.vcs">versit Conference</A></P></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%">31:</TD>
</TR>

<TR>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="17%" BGCOLOR="#C0C0C0">1:</TD>
</TR>
</TABLE>
</CENTER>
<HR><A HREF="mailto:john.smith@host.com"><IMG BORDER=2 HEIGHT=24
WIDTH=22></A>
</BODY></HTML>
```

Section 4 : UI Support Recommendations

When integrating vCalendar 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 vCalendar technology can be better utilized if an application supports the vCalendar in each of these UI actions.

4.1 File System

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

4.2 Clipboard

It is recommended that applications integrating support for the vCalendar specification provide UI capabilities for exchanging vCalendar objects through the operating system’s clipboard. In operating systems that provide support for registering clipboard format types, it is recommended that the vCalendar object be registered using the string ***+//ISBN 1-887687-00-9::versit::PDI//vCalendar***. 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 vCalendar object. Applications copying a vCalendar to the clipboard should put the vCalendar object on to the clipboard in both the vCalendar registered format and a plain text format.

4.3 Drag/Drop

It is recommended that applications integrating support for the vCalendar specification provide UI capabilities for exchanging vCalendar 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 vCalendar object be registered using the string ***+//ISBN 1-887687-00-9::versit::PDI//vCalendar***. 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 vCalendar object. Applications acting as a Drag Source should advertise their ability to render the vCalendar in both the vCalendar 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 vCalendar Reader or Writer to conform to this specification it must meet the following criteria:

- 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.

Section 6 : Extended Recurrence Grammar

The material in this section is included in this specification for reference information.It is copied, with permission of the XAPIA, from the XAPIA Calendaring and Sceduling API (CSA) Specification. This section defines an extended recurrence rule grammar that may be useful to implementations wishing to extend the capability of the basic recurrence rule defined by this specification. The material is equally applicable to extended support of the exception rules for repeating events.

6.1 Rule Introduction

A recurrence rule is made up of one or more recurrence **frequencies**. The frequencies express the granularity of the repeating event. The smallest granularity is based on minutes, the largest is based on years. Each frequency is immediately followed by an **interval**. The interval helps to define how often the frequency repeats (daily, every third day, etc):

```
D2
```

Where, where `D` is the Frequency and `2` is the Interval.

```
M5 Repeat every five minutes
D1 Repeat daily
D2 Repeat every other day
D3 Repeat every third day
W1 Repeat weekly
W2 Repeat every other week
W3 Repeat every third week
```

The meaning of the interval depends on the frequency. As an example, the **5** in `M5` is in minutes while the **3** in **D3** is in days.

A rule can end with the **duration symbol**, **#**, followed by a number. This defines the number of times the repetition occurs (including the first time).

```
D2 #5
```

Where, `#5` is the Duration. In this example, the event occurs every other day and the duration indicates it occur 5 times.

There may be other information between the frequency and the duration that supplements the meaning of the rule:

```
D2 1200 1600 #5
```

In this example, the event occurs every other day at 1200 and 1600 for a total of 10 occurrences. The duration controls the number of times the rule occurs. In this case the rule defines two occurrences (1200 and 1600) so a total of 10 (2 x 5) occurrences are generated.

A rule can be made up of several recurrence rules:

```
MP6 1+ MO #5 D2 1200 1600 #5 M5 #3
```

This recurrence rule is made up of three recurrence rules. Every time the first rule executes (every 6 months) it executes the next rule to the right. If there is not a rule to the right an event is generated. In this case there is a daily frequency rule to the right of the monthly frequency rule. It executes twice a day; starting on the first Monday of the month. The daily frequency rule executes a total of ten times. Since there is a rule following the daily rule it executes it each time the daily frequency rule executes. The minute frequency rule is executed three times, every time the daily frequency rule executes, for a total of six times a day. The above rules generate a total of 150 (5 * (2 * 5) * 3) occurrences. An occurrence specifier may be used in some rules. The occurrence specifier controls which occurrence of a weekday in a month an event occurs on:

```
1+, 2+, ... 5+ for the first occurrence, second, ...fifth occurrence of the
month.
```

1-, **2-**...**5-** for the last occurrence, second to last occurrence, etc.

```
MP2 1+ 2- FR #3
```

Where, **1+** and **2-** are Occurrence Specifiers.

This rule defines an event which happens every other month on the first Friday and second to the last Friday of the month. It generates a total of six occurrences. The end marker symbol $ can be used to terminate a rule early:

```
W1 0100$ 0200 0300 #4
```

The above rule generates occurrences weekly for 4 weeks at 0100, 0200 and 0300. On the last (fourth week) an event is generated at 0100 only.

Recurrence rules are written from largest granularity to smallest.

6.2 Grammar

The following grammar defines the extended recurrence rule syntax.

```
{}    0 or more

[]    0 or 1

start     ::= <minuteop> [<enddate>] |

      <daily> [<enddate>] |

      <weekly> [<enddate>] |

      <monthlybypos> [<enddate>] |

      <monthlybyday> [<enddate>] |

      <yearlybymonth> [<enddate>] |

      <yearlybyday> [<enddate>]

digit ::= <0|1|2|3|4|5|6|7|8|9>

digits ::= <digit> {<digits>}

endmarker ::= $
```

```
enddate       ::= ISO 8601 (clause 5.4.1) string(e.g. 19940712T101530Z) in
UTC

interval  ::= <digits>

duration  ::= #<digits>

lastday   ::= LD

plus  ::= +

minus ::= -

daynumber     ::= <1-31>[<plus>|<minus>][<endmarker>] | <lastday>

daynumberlist ::= daynumber {<daynumber>}

month         ::= <1-12>[<endmarker>]

monthlist ::= <month> {<monthlist>}

day       ::= <1-366>[<endmarker>]

daylist   ::= <day> {<daylist>}

occurrence    ::= <1-5><plus>[<endmarker>] | <1-5><minus>[<endmarker>]

occurrencelist    ::= <occurrence> {<occurrencelist>}

weekday   ::= <SU|MO|TU|WE|TH|FR|SA>[<endmarker>]

weekdaylist   ::= <weekday> {<weekdaylist>}

time      ::= <hhmm>[<endmarker>]

timelist  ::= <time> {<timelist>}

daytime       ::= <weekday> {<timelist>} {<daytime>}

weekdaytime   ::= <occurrencelist> <weekdaylist> {<timelist>} {<weekdaytime>}

minuteop  ::= M<interval> [<duration>]

daily     ::= D<interval> [<timelist>] [<duration>] [<minuteop>]

weekly    ::= W<interval> <daytime> [<duration>] [<minuteop>]

monthlybypos  ::= MP<interval> [<weekdaytime>] [<duration>][<weekly> |
<daily> | <minuteop>]

monthlybyday  ::= MD<interval> [<daynumberlist>] [<duration>] [<weekly> |
<daily> | <minuteop>]

yearlybymonth ::= YM<interval> [<monthlist>] [<duration>] [<monthlybyday> |
<monthlybypos> | <weekly> | <daily> | <minuteop>]

yearlybyday   ::= YD<interval> [<daylist>] [<duration>] [<monthlybyday> |
<monthlybypos> | <weekly> | <daily> | <minuteop>]
```

6.3 Glossary

enddate	Controls when a repeating event terminates. The enddate is the last time an event can occur.
enddate	Controls when a repeating event terminates.
interval	Defines the frequency in which a rule repeats.
duration	Controls the number of occurrences a rule generates.

lastday	Can be used as a replacement to daynumber to indicate the last day of the month.
daynumber	A number representing a day of the month.
month	A number representing a month of the year.
day	A number representing a day of the year.
occurrence	Controls which week of the month a particular weekday event occurs.
weekday	A symbol representing a day of the week.
time	The time in hours and minutes using a 24 hour clock.
daytime	Controls which days (and hours) of the week a weekly event occurs.
weekdaytime	Used in monthly occurrences to specify which weeks and days an event occurs.
minuteop	Defines a rule that repeats on a particular minute interval.
daily	Defines a rule that repeats on a daily basis.
weekly	Defines a rule that repeats on a weekly basis.
monthlybypos	Defines a rule that repeats on a monthly basis on a relative day and week.
monthlybyday	Defines a rule that repeats on a monthly basis on an absolute day.
yearlybymonth	Defines a rule that repeats on specific months of the year.
yearlybyday	Defines a rule that repeats on specific days of the year.

6.4 Policies

1. The duration portion of a rule defines the total number of occurrences the rule generates, including the first event. As an example, the rule **`MP1 #3 W1 #3`** starting on 1/1/94 would generate occurrences on 1/1/94, 1/8, 1/15, 2/5/94, 2/12, 2/19, 3/5/94, 3/12, 3/19.
2. The duration granularity is defined by the recurrence frequency immediately preceding the duration portion of the rule. For example, **`D1 #5 M15 #4`** establishes a repeating event which happens for five days, four times per day.
3. Information, not contained in the rule, necessary to determine the next event time and date is derived from the event start time.
4. If no specific time is indicated in the recurrence rule it is taken from the event.
5. If an end date and a duration for the first rule in a nested rule are specified in the rule, then the recurring event ceases when the end date is reached or the number of occurrences indicated in the duration occur; whichever comes first.
6. If the duration or and end date is not established in the rule (e.g. ``D2'') the event occurs twice. That is **`D2`** is equivalent to **`D2 #2`**.
7. If an endmark is used in a second or later rule of a nested rule, then the endmark is applied each time that rule is executed by the previous rule. **`YM1 1 6 #1 MD1 7$ 14`**

generates occurrences on 1/7 1/14 2/7 6/7 6/14 7/7.

8. If an endmark is used on a day of the week which is followed by several times (**`TU$ 1200 1300`**) or an endmark is used on a week occurrence that is followed by several weekdays (**`1+$ TU WE`**) the repeating event stops after the last time or week day in the list is executed.
9. If a rule has an ambiguity with respect to whether it will repeat on a specific day (12th of the month) vs on a relative day (2nd Friday of the month), the specific day takes precedence. The only exception to this policy is policy 14.
10. A duration of **`#0`** means repeat this event forever.
11. Nested rules can not have a duration of 0. These are not allowed:

```
YM1 6 #10 MP1 1+ SA #0
D5 0600 0800 #5 M5 #0
```

12. Using the occurrence specifier **`5+`** (e.g. 5th Friday) or **`5`**- (e.g. 5th from last Friday) in a month that does not contain 5 weeks does not generate an event and thus does not count against the duration. The same applies to providing a day of the month that does not occur in the month: 31st.
13. The start time and date of an event must be in-sync with one of the event slots defined by its ocurrence rule. The following are not allowed:

Initial Appt Time:	**`1300`**
Recurrence Rule:	**`D1 1400 #5`**
Initial Appt Date:	7/1/94 (Friday)
Recurrence Rule:	**`W1 MO TH #5`**

The following are acceptable:

Initial Appt Time:	**`1300`**
Recurrence Rule:	**`D1 #5`** or **`D1 1300 #5`**
Initial Appt Date:	7/1/94 (Friday)
Recurrence Rule:	**`W1 MO FR #5`** or **`W1 #5`**

14. If the optional **`<weekdaytime>`** information is missing from a **`<monthlybypos>`** frequency, the information is derived from the initial event. The **`<occurrence>`** used in the recurring event is a count from the beginning of the month to the event date and the **`<weekday>`** used is the day of the week the initial event is scheduled to occur on. If the **`<monthlybypos>`** frequency does not list a week day (e.g. **`SU`**) in the rule, then the week day is established from the initial event information. As an example, the rule **`MP1 #3`** used in an event with a start date of 7/20/94 (which is the third Wed of the month) will repeat on 8/17/94 which is the third Wed of the month.
15. The next event of a higher order rule does not execute until all the occurrences of a subrule are generated. If the next event of a higher order rule comes earlier in time than the last event of a subrule then the missed occurrences are not generated. In other words,

subrules can not interleave occurrences with other subrules. The following results in indeterminate results because the minute subrule which begins to execute at 0630 generates occurrences beyond 0700 which is when the daily subrule begins executing again:

```
D1 0630 0700 #4 M45 #5
```

Another incorrect rule:

```
MP1 1+ 1- #3 W2 TU TH #5
```

6.5 Examples

Hourly for 12 hours (12:00, 1:00,...10:00, 11:00):

```
M60 #12
```

Every 5 minutes for 1 hour (1:00, 1:05, 1:10,...1:50, 1:55):

```
M5 #12
```

Daily, for 5 days:

```
D1 #5
```

Daily, for 5 days repeating at 10 minute intervals for 1 hour. e.g. 6/1 at 12:00, 12:10, 12:20, ... 12:50; 6/2 at 12:00, 12:10, ...

```
D1 #5 M10 #6
```

Every other day, two times:

```
D2
```

Every other day at 6AM, 12noon and 3PM for a duration of two occurrences (span of three days). e.g. 6/1/94 at 6, 12 and 3PM and 6/3/94 at 6, 12 and 3PM.

```
D2 0600 1200 1500 #2
```

Every other day at 6AM, 12noon and 3PM for a duration of three occurrences (a span of 5 days) stopping at noon on the fifth day. e.g. 6/1/94 at 6, 12, and 3, 6/3/94 at 6, 12 and 3 and 6/5/94 at 6 and 12.

```
D2 0600 1200$ 1500 #3
```

Weekly at 6 AM (repeat every 15 minutes for an hour) for five weeks. e.g. 6:00, 6:15, 6:30, 6:45 on 6/1, 6/8, 6/15, 6/22 and 6/29.

```
D7 0600 #5 M15 #4
```

Weekly at 6 AM (repeat every 15 minutes for an hour) for four weeks stopping at 6AM on the last event day. e.g. 6:00, 6:15, 6:30, 6:45 on 6/1, 6/8, 6/15 and 6:00 on 6/22.

```
D7 0600$ #4 M15 #4
```

Weekly at 6 AM (repeat every 15 minutes for an hour) for 1 week stopping at 6:45AM. e.g.

6:00, 6:15, 6:30, 6:45 on 6/1.

```
D7 0600 #1 M15 #4 or
D7 #1 M15 #4 /* start time defined in appt entry */ or
M15 #4 /* start time defined in appt entry */
```

Weekly for four weeks:

```
W1 #4
```

Biweekly on Monday and Tuesday for 2 occurrences ending on a Monday:

```
W2 MO$ TU #2
```

Weekly on Tuesday and Thursday at the time specified in the appt and repeated at time + 5 minutes:

```
W1 TU TH #3 M5 #2
```

Weekly on Tuesday at 1200 and 1230 and Thursday at 1130 and 1200 for 10 weeks:

```
W1 TU 1200 TH 1130 #10 M30 or
W1 TU 1200 1230 TH 1130 1200 #10
```

Weekly on Tuesday at 1200 and 1230 and Thursday at 1130 and 1200 for 10 weeks stopping on the last TU at 1230:

```
W1 TU$ 1200 TH 1130 #10 M30 or
W1 TU$ 1200 1230 TH 1130 1200 #10
```

Weekly on Tuesday at 1200 and 1230 and Thursday at 1130 and 1200 for 10 weeks stopping on the last Tuesday at 1200:

```
W1 TU 1200$ 1230 TH 1130 1200 #10
```

Monthly for 1 year:

```
MP1 #12
```

Every other month on the first and last Friday of the month for 5 months:

```
MP2 1+ 1- FR #3
```

Monthly on the second to the last day of the month for 5 months:

```
MD1 2- #5
```

Monthly on the second to the last Monday of the month for 6 months:

```
MP1 2- MO #6
```

Monthly on the third to the last day of the month for forever:

```
MD1 3- #0
```

Monthly on the seventh to the last day of the month for 12 months:

```
MD1 7- #12
```

Every other month on the first and last Friday of the month for 5 months stopping on the first Friday in the fifth month:

```
MP2 1+$ 1- FR #3
```

Every six months on the first Monday of the month (repeat for 5 days) for 24 months:

```
MP6 1+ MO #5 D1 #5
```

Every six months on the first Monday of the month (repeat every other day at 0600, 1200 and 1500 for 20 days) for 24 months:

```
MP6 1+ MO #5 D2 0600 1200 1500 #10
```

Every six months on the first Monday of the month (repeat every other day at 0600, 1200 and 1500 for 20 days (repeat every 5 minutes for 3 times)) for 24 months:

```
MP6 1+ MO #5 D2 0600 1200 1500 #10 M5 #3
```

Every six months on the first Monday of the month and the second to last Thursday of the month (repeat five minutes later) for 24 months:

```
MP6 1+ MO 2- TH #5 M5 #2
```

Every six months on the first SU and MO at Noon, the second TU and WE at 1:00PM and the third TH and FR at 2:00PM:

```
MP6 1+ SU MO 1200 2+ TU WE 1300 3+ TH FR 1400 #4
```

Every month on the 7th for 12 months:

```
MD1 7 #12
```

Every month on the 7th, 14th, 21st, 28th for 12 months

```
MD1 7 14 21 28 #12
```

Every month on the 10th and 20th for 24 months - daily for 5 days at 0600, 1200 and 1600 - every 15 minutes for an hour:

```
MD1 10 20 #24 D1 0600 1200 1600 #5 M15 #4
```

Yearly on the 1st, 6th and 12 month on the first Monday and last Friday of the month:

```
YM1 1 6 12 #5 MP1 1+ MO 1- FR
```

Every other year on the 6th month (on the 12th day) for 5 years.

```
YM2 6 #3 MD1 12
```

Yearly on the 7th 14th 21st and 28th of the 1st 3rd and 8th month and on the 7th and 14th of the 2nd, 4th and 9th months ending on the 4th month, 14th day of the 5th year:

```
YM1 1 3$ 8 #5 MD1 7 14$ 21 28
```

Yearly on the 6th, 9th and 10th month on all weekends of the month:

```
YM1 6 9 10 #10 MP1 1+ 2+ 3+ 4+ 1- SA SU #1
```

Yearly on the 6th month for 10 years, weekly on Tuesday and Thursday at 1100 and 1300 for 4 weeks:

```
YM1 6 #10 W1 TU TH 1100 1300 #4
```

Yearly on the 1st, 100th, 200th and 300th day for 4 years:

```
YD1 1 100 200 300 #4
```

Yearly on the 1st - 5th days and 100th - 104th days:

```
YD1 1 100 #5 D1 #5
```

Yearly on the 1st - 5th days and 100th - 104th days stopping on 1/2/99:

```
YD1 1 100 D1 #5 19990102T000000Z
```