

Network Working Group
Internet Draft
<ietf-calsch-ical-02.txt>
Expires January 1998

Frank Dawson, Lotus
Derik Stenerson, Microsoft
July 29, 1997

Internet Calendaring and Scheduling Core Object Specification (iCalendar)

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months. Internet-Drafts may be updated, replaced, or made obsolete by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as a "working draft" or "work in progress".

To learn the current status of any Internet-Draft, please check the `1id-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on `ds.internic.net` (US East Coast), `nic.nordu.net` (Europe), `ftp.isi.edu` (US West Coast), or `munniari.oz.au` (Pacific Rim).

Distribution of this document is unlimited.

Abstract

There is a clear need to provide and deploy interoperable calendaring and scheduling services for the Internet. Current group scheduling and Personal Information Management (PIM) products are being extended for use across the Internet, today, in proprietary ways. This document has been defined to provide the a definition of a common format for openly exchanging calendaring and scheduling information across the Internet.

This memo is formatted as a registration for a MIME media type per [\[RFC 2048\]](#). However, the format in this memo is equally applicable for use outside of a MIME message content type.

The proposed media type value is 'TEXT/CALENDAR'. This string would label a media type containing calendaring and scheduling information encoded as text characters formatted in a manner outlined below.

This MIME media type provides a standard content type for capturing calendar event and to-do information. It also can be used to convey free/busy time information. The content type is suitable as a MIME message entity that can be transferred over MIME based email systems or using HTTP. In addition, the content type is useful as an object for interactions between desktop applications using the operating system clipboard, drag/drop or file systems capabilities.

Dawson/Stenerson

1

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

This document is based on the earlier work of the vCalendar specification for the exchange of personal calendaring and scheduling information. In order to avoid confusion with this referenced work, this document is to be known as the iCalendar specification. This document is based on the calendaring and scheduling model defined in [[ICMS](#)]. The document is also the basis for the calendaring and scheduling interoperability protocol defined in [ITIP-1], [[ITIP-2](#)] and [[ITIP-3](#)].

This document also includes the format for defining content type profiles. A content type profile is a document that defines a set of usage constraints for the iCalendar object. For example, a profile might be defined to specify how the iCalendar object can be used to provide for a set of interpersonal scheduling messages. Such a profile might define scheduling messages that request an event be scheduled, reply to an event request, send a cancellation notice for an event, modify or replace the definition of an event, provide a counter proposal for an original event request, delegate an event request to another individual, request free or busy time, reply to a free or busy time request, or provide similar scheduling messages for a to-do or journal entry calendar component.

Dawson/Stenerson

2

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

Table of Contents

1. Introduction.....	5
2. Basic Grammar and Conventions.....	5
3. Definitions.....	6
3.1 Alarm.....	6
3.2 Busy Time.....	6
3.3 Calendar Component.....	6
3.4 Calendar Date.....	6
3.5 Calendar Object.....	7
3.6 Calendar Properties.....	7
3.7 Calendar Scale.....	7
3.8 Component Properties.....	7
3.9 Coordinate Universal Time (UTC).....	7
3.10 Daylight Saving Time (DST).....	7
3.11 Event.....	7
3.12 Free Time.....	7
3.13 Gregorian Calendar.....	8
3.14 Journal.....	8
3.15 Local Time.....	8
3.16 Period.....	8
3.17 Recurrence Rule.....	8
3.18 Reminder.....	8
3.19 Repeating Event or To-do.....	8
3.20 Standard Time.....	8
3.21 Time Zone.....	9
3.22 To-do.....	9

4. TEXT/CALENDAR Registration Information.....	9
5. iCalendar Object Specification.....	11
5.1 Syntax Considerations.....	11
5.1.1 Content Lines.....	13
5.1.2 List and Field Separators.....	14
5.1.3 Multiple Values.....	14
5.1.4 Character Set.....	15
5.1.5 Language.....	15
5.1.6 Content Encoding.....	15
5.1.7 Binary Content.....	15
5.1.8 Recurrence Set.....	15
5.1.9 Data Types.....	16
5.2 iCalendar Object.....	21
5.3 Property.....	21
5.4 Calendar Components.....	22
5.4.1 Event Component.....	22
5.4.2 To-do Component.....	23
5.4.3 Journal Component.....	23
5.4.4 Free/Busy Component.....	24
5.4.5 Alarm Component.....	25
5.4.6 Timezone Component.....	26
5.5 Calendar Properties.....	30
5.5.1 Calendar Scale.....	30
5.5.2 Product Identifier.....	30
5.5.3 Profile.....	31
5.5.4 Profile Version.....	31
5.5.5 Source.....	32
5.5.6 Source Name.....	32
5.5.7 Version.....	32

Dawson/Stenerson

3

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

5.6 Component Properties.....	33
5.6.1 Attachment.....	33
5.6.2 Attendee.....	33
5.6.3 Categories.....	36
5.6.4 Classification.....	36
5.6.5 Comment.....	37
5.6.6 Date/Time Completed.....	37
5.6.7 Date/Time Created.....	38
5.6.8 Date/Time Due.....	38
5.6.9 Date/Time End.....	38
5.6.10 Date/Time Stamp.....	39
5.6.11 Date/Time Start.....	39
5.6.12 Daylight.....	40
5.6.13 Description.....	40
5.6.14 Duration.....	41
5.6.15 Exception Date/Times.....	41

5.6.16	Exception Rule.....	42
5.6.17	Free/Busy Time.....	42
5.6.18	Geographic Position.....	43
5.6.19	Last Modified.....	44
5.6.20	Location.....	44
5.6.21	Priority.....	45
5.6.22	Recurrence Date/Times.....	45
5.6.23	Recurrence ID.....	46
5.6.24	Recurrence Rule.....	46
5.6.25	Related To.....	52
5.6.26	Repeat Count.....	53
5.6.27	Request Status.....	53
5.6.28	Resources.....	55
5.6.29	Response Sequence Number.....	56
5.6.30	Sequence Number.....	56
5.6.31	Status.....	57
5.6.32	Summary.....	57
5.6.33	Time Transparency.....	58
5.6.34	Time Zone Name.....	58
5.6.35	Time Zone Offset.....	59
5.6.36	Uniform Resource Locator.....	59
5.6.37	Unique Identifier.....	59
5.6.38	Non-standard Properties.....	60
6.	Recommended Practices.....	60
7.	Registration of Content Type Elements.....	61
7.1	Registration of New and ModifiedProfiles.....	61
7.2	Registration of New Properties.....	61
7.2.1	Define the property.....	61
7.2.2	Post the Property definition.....	62
7.2.3	Allow a comment period.....	62
7.2.4	Submit the property for approval.....	62
7.3	Property Change Control.....	63
8.	File extension.....	63
9.	Macintosh File Type Code.....	63
10.	References.....	63
11.	Acknowledgments.....	65
12.	Author's Address.....	65
13.	iCalendar Object Examples.....	66

Dawson/Stenerson

4

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

[1.](#) **Introduction**

The use of calendaring and scheduling has grown considerably in the last decade. Enterprise and inter-enterprise business has become dependent on rapid scheduling of events and actions using this information technology. However, the longer term growth of calendaring and scheduling, is currently limited by the lack of Internet standards for the message content types that are central to these groupware applications. This specification is intended to progress the level of interoperability possible between dissimilar calendaring and scheduling applications. This specification defines a MIME content type for exchanging electronic calendaring and scheduling information. The Internet Calendaring and Scheduling Core Object Specification, or iCalendar, allows for the capture and exchange of information normally stored within a calendaring and scheduling application; such as a Personal Information Manager or a Group Scheduling product.

The calendaring and scheduling model implemented by this specification is defined in the [[ICMS](#)].

The format is suitable as an exchange format between applications or systems. The format is defined in terms of a MIME content type. This will enable the object to be exchanged using several transports, including but not limited to SMTP, HTTP, a file system, desktop interactive protocols such as the use of a memory-based clipboard or drag/drop interactions, point-to-point asynchronous communication, wired-network transport, or some form of unwired transport such as infrared might also be used.

The definition of a calendaring and scheduling interoperability protocol is the subject of another specification [[ITIP-1](#)], [[ITIP-2](#)] and [[ITIP-3](#)].

The specification also provides for the definition of usage profiles that will map this content type to a set of messages for supporting calendaring and scheduling operations such as requesting, replying to, modifying, and canceling meetings or appointments, to-dos and journal entries. The usage profiles can be used to define other calendaring and scheduling operations such as requesting for and replying with free/busy time data.

The specification also includes a formal grammar for the content type 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.

[2.](#) Basic Grammar and Conventions

This document makes use of both a descriptive prose and a more formal notation for defining the calendaring and scheduling format.

The notation used in this document is the augmented BNF notation of [RFC 822]. Readers intending on implementing this format defined in

Dawson/Stenerson

5

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

this document should be familiar with this notation in order to properly interpret the specifications of this document.

All numeric and hexadecimal values used in this document are given in decimal notation. All names of properties, property parameters, enumerated property values and property parameter values are case-insensitive. However, all other property values are case-sensitive, unless otherwise stated.

Note: All indented editorial notes, such as this one, are intended to provide the reader with additional information that is not essential to the building of a conformant implementation of the specifications of this document. The information is provided to highlight a particular feature or characteristic of the specifications.

The format for the iCalendar object is based on the syntax of the [MIME DIR] content type. While the iCalendar object is not a profile of the [MIME DIR] content type, it does reuse a number of the elements from the [MIME DIR] specification.

3. Definitions

EDITORS' NOTE: This section may be removed if this text is added to the [ICSM].

Date and time, as well as, calendaring and scheduling terminology are used in every day conversations. However, there are precise definitions of many of these terms that are used by this memo.

3.1 Alarm

Also called a reminder. An activity that is an asynchronous mechanism for providing feedback for a pending or past event or to-do.

3.2 Busy Time

A period of time on a calendar where there is already scheduled one or more events or that is otherwise not available for scheduling.

3.3 Calendar Component

One of a number of entities that may be found within a calendar object. In particular, a calendar may be composed of calendar properties and event, to-do, journal, free/busy, time zone or alarm calendar components. Calendar components are identified by unique delimiters within a calendar object. Calendar components provide an organized collection of component properties.

[3.4](#) **Calendar Date**

A particular day of a calendar year identified by its position within the year.

Dawson/Stenerson	6	Expires January 1998
------------------	---	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

[3.5](#) **Calendar Object**

An entity consisting of an organized collection of calendar properties and calendar components. The calendar object is identified by unique delimiters.

[3.6](#) **Calendar Properties**

Attributes that apply to the calendar object as a whole. For example, the iCalendar version used to format the calendar object, an identifier of the product that created the calendar object, the calendar scale used to represent the calendar information and time zone information.

[3.7](#) **Calendar Scale**

The particular type of calendar in general use. For example, Gregorian, Buddhist Era, Japanese Emperor Era, Chinese Lunar, Islamic, and Jewish Calendars.

[3.8](#) **Component Properties**

Attributes that can only appear within one or more calendar components. For example, the due date can only appear within a to-do calendar component. The start date and time applies to both the event and the to-do component.

3.14 Journal

A calendar component that defines a collection of information intended for human presentation and is minimally specified by a calendar date and one or more descriptions.

3.15 Local Time

The clock time in public use in a locale. Local time is often referenced by the customary name for the time zone in which it is located. The relationship between local time and UTC is based on the offset that is in use for a particular time zone. In general, the formula is as follows:

$$\text{local time} = \text{UTC} + (\text{offset})$$

3.16 Period

A duration of time, specified as either a defined length of time or by its beginning and end points.

3.17 Recurrence Rule

A notation used to represent repeating occurrences, or the exceptions to such a repetition of an event or a to-do. The recurrence rule can also be used in the specification of a time zone description. This document defines a particular notation used in recurrence rules within this specification.

3.18 Reminder

See Alarm.

3.19 Repeating Event or To-do

An event or to-do that repeats for one or more additional occurrences. The recurrence may be defined with discrete dates and times and/or with a recurrence rule.

3.20 Standard Time

Introduced by Sir Sanford Fleming and others around 1870, standard time is a scheme for dividing the world into zones where the same time would be kept. The original proposal was to divide the world

into 24 zones, each zone having a width of 15 degrees of longitude. The center zone was originally the meridian passing through Greenwich, England, called Greenwich Mean Time (GMT). The time in the zones was decremented by one hour per zone going westwards and was incremented by one hour per zone going eastwards from GMT. Changes have been made to the original proposal to accommodate political boundaries. In addition, some countries and regions specify 30 or 45 minute offsets, rather than the full 60 minute offset. Standard time is also known as Winter Time in some regions.

GMT and UTC are generally equivalent. However, by international agreement, the GMT term is discouraged in favor of the term UTC for all general time keeping.

3.21 Time Zone

The particular time zone that time in a particular location is expressed in. A time zone is unambiguously defined by the set of time measurement rules determined by the governing body for the given location. These rules describe at a minimum the base offset from UTC, often referred to as the Standard Time offset. Optionally, if Daylight Savings Time is observed, the rules will specify the Daylight Savings Time offset and either a set of rules describing the transition to and from Daylight Savings Time or absolute dates describing the movement in and out of Daylight Savings Time. It is important to note that these rules are not static. Time zones may also have a local customary name. However, not all time zones have a special name for their time. The customary names for time zones are often abbreviated. However, not all time zone abbreviations are unique. For example, AST may mean Atlantic Standard Time, Alaska Standard Time, and even Aleutian Standard Time. Each of these are different offsets from UTC. Nevertheless, customary names for time zones are in use in various parts of the world.

3.22 To-do

A calendar component that defines an action item and is minimally specified by an effective calendar date and time of day, a due calendar date and time of day, a priority and a description.

4. TEXT/CALENDAR Registration Information

The Calendaring and Scheduling Core Object Specification is intended for use as a MIME content type. However, the implementation of the specification is in no way limited solely as a MIME content type.

The following text is intended to register this specification as the

MIME content type "text/calendar".

To: ietf-types@uninett.no

Subject: Registration of MIME content type text/calendar.

MIME media type name: text

MIME subtype name: calendar

Dawson/Stenerson

9

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

Required parameters: profile

The "profile" parameter is used to convey the scheduling usage to which the calendaring and scheduling information pertains. It also is an identifier for the set of properties that the iCalendar object will consist of. The parameter is intended to be used as a guide to applications interpreting the information contained within the body part. It should NOT be used to exclude or require particular pieces of information unless the identified profile definition specifically calls for this behavior. Unless specifically forbidden by a particular profile definition, a text/calendar content type may contain any set of properties permitted by the Calendaring and Scheduling Core Object Specification.

The value for the "profile" parameter is defined as follows:

profile = component "-" usage

component = "EVENT" / "event" / "TODO" / "todo"
/ "JOURNAL" / "journal" / "FREEBUSY"
/ "freebusy" / x-token / iana-comp

usage = "REQUEST" / "request" / "REPLY" / "reply"
/ "CANCEL" / "cancel" / x-token / iana-usage

x-token = <The two characters "X-" or "x-" followed, with no intervening white space, by any atom, where atom is from [section 3.3 of \[RFC 822\]](#)>

iana-comp = <A publicly defined extension component, registered with IANA, as specified by this document>

iana-usage = <A publicly defined extension usage, registered with IANA, as specified by this document>

Optional parameters: charset

The "charset" parameter is defined in [[RFC 2046](#)] for other body parts. It is used to identify the default character set used within the body part.

Optional content header fields: Any header fields defined by [RFC 2045].

Encoding considerations: This MIME content type can contain 8bit characters, so the use of quoted-printable or base64 MIME content-transfer-encodings may be necessary when iCalendar objects are transferred across protocols restricted to the 7bit repertoire. Note that each property in the content entity may also have an inline encoding for the body part as a whole (i.e., inline encoding is performed first, then Content-Transfer-Encoding is applied to the entire body part). This means that content values may end up encoded twice.

Dawson/Stenerson

10

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

Security considerations: The calendaring and scheduling information based on this MIME content type may include references to Uniform Resource Locators that may be programmed resources. In addition, this information may contain direct references to executable programs intended to be used as procedure-based alarms for an event or to-do. Implementers and users of this specification should be aware of the network security implications of accepting and parsing such information. In addition, the security considerations observed by implementations of electronic mail systems should be followed for this specification.

Interoperability considerations: This MIME content type is intended to provide interoperability between calendaring and scheduling products. It is heavily based on the earlier [[VCAL](#)] industry specification.

Intended Usage: COMMON

Published specification: This document.

Author/Change controllers:

Frank Dawson
6544 Battleford Drive
Raleigh, NC 27613-3502
919-676-9515 (Telephone)

919-676-9564 (Facsimile)
fdawson@earthlink.net (Internet Mail)

Derik Stenerson
One Microsoft Way
Redmond, WA 98052-6399
206-936-5522 (Telephone)
206-936-7329 (Facsimile)
deriks@microsoft.com (Internet Mail)

5. iCalendar Object Specification

The following sections define the details of a Calendaring and Scheduling Core Object Specification. This information is intended to be an integral part of the MIME content type registration. In addition, this information may be used independent of such content registration. In particular, this specification has direct applicability for use as a calendaring and scheduling exchange format in file-, memory- or network-based transport mechanisms.

5.1 Syntax Considerations

The content information associated with an iCalendar object is formatted using a syntax similar to that defined by [MIME DIR]. That is, the content information consists of one or more CRLF-separated lines in the following format:

```
contentline = name [";" paramlist] ":" value CRLF
;Folding permitted on content lines.
```

Dawson/Stenerson 11 Expires January 1998

Internet Draft C&S Core Object Specification July 29, 1997

LWSP = SPACE / HTAB

SPACE = <ASCII Decimal 32>

HTAB = <ASCII Decimal 9>

name = x-name / iana-name ;An iCalendar attribute/property

x-name = <The two characters "X-" or "x-" followed, with no intervening white space, by any atom>

iana-name = <A publicly defined name, registered with IANA>

paramlist = parameter / paramlist ";" parameter

parameter = encodingparm
/ valuetypeparm ;If not present => inline value
/ languageparm
/ [parmtype "="] parmvalues

encodingparm = "encoding" "=" encodetype

encodetype = "8bit" ;From [[RFC 2045](#)]
/ "7bit" ;From [[RFC 2045](#)]
/ "q" ;From [[RFC 2045](#)]
/ "b" ;From [[RFC 2045](#)]

valuetypeparm = "value" "=" valuetype

valuetype = "url"
/ "text"
/ "date"
/ "time"
/ "date-time"
/ "period"
/ "duration"
/ "boolean"
/ "integer"
/ "float"
/ "cal-address"
/ "utc-offset"
/ x-token
/ iana-value

iana-value = <A publicly defined extension value type, registered
with IANA, as specified by this document>

languageparm = "language" "=" language
;As defined in [[RFC 1766](#)]

parmtype = x-token / iana-ptype

iana-ptype = <A publicly defined extension parameter type,
registered with IANA, as specified by this document>

parmvalues = parmvalue / parmvalues "," parmvalue

parmvalue = x-name / iana-pvalue

iana-pvalue = <A publicly defined extension parameter value,

registered with IANA, as specified by this document>

value = url / text / date / time / date-time / period /
/ duration / boolean / integer / float / cal-address
/ utc-offset / x-token / iana-value

iana-value = <A publicly defined property value data type,
registered with IANA, as defined in this document>

5.1.1 Content Lines

Individual lines within the iCalendar object are delimited by a line break, which is a CRLF sequence (ASCII decimal 13, followed by ASCII decimal 10). Line should not be longer than 76 characters, excluding the line break.

Long lines of text can be split into a multiple-line representations using a line "folding" technique. That is, a long line may be split at any point by inserting a CRLF immediately followed by a single LWSP character (i.e., SPACE, ASCII decimal 32 or HTAB, ASCII decimal 9). Any sequence of CRLF followed immediately by a single LWSP character is ignored (i.e., removed) when processing the content type.

For example the line:

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

Can be represented as:

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

The process of moving from this folded multiple-line representation to its single line representation is called "unfolding". Unfolding is accomplished by removing the CRLF character and the LWSP character that immediately follows.

An intentional 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 imbedded formatted line breaks (i.e., hard line breaks) must be encoded using an alternate encoding of either the "Q" or "B" encodings, as defined in [[RFC 2047](#)]. These encodings are used directly and without any of the additional syntax elements of [[RFC 2047](#)] encoded-words.

Since neither the "Q" nor the "B" encodings ever produce LWSP characters (note that the "Q" encoding turns spaces into underscores), or CRLF character sequences as output, LWSP characters and CRLF character sequences can be freely inserted into encoded

material at any point to fold encoded field values. All LWSP characters and CRLF character sequences should be ignored when

Dawson/Stenerson

13

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

decoding an encoded field value. The "Q" encoding of multiple lines of formatted text are separated with a Q CRLF sequence of "=0D=0A". The length restrictions [[RFC 2047](#)] imposes on encoded-words do not apply in this context, but fields encoded with the "Q" or "B" encodings must be folded into lines of no longer than 76 characters.

For example a multiple line DESCRIPTION property value of:

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

Could be represented in "Q" encoding as:

```
DESCRIPTION;ENCODING=Q:Project_XYZ
_Final_Review=0D=0A
Conference_Room_-_3B=0D=0A
Come_Prepared.
```

And in the "B" encoding as:

```
DESCRIPTION;ENCODING=UHJvamVjdCBYWVogRmluYWw
gUmV2aWV3DQpDb25mZXJIbm NIIFJvb20gLSAzQg0KQ29tZ
SBQcmVwYXJIZC4NCg = =
```

5.1.2 List and Field Separators

Where a property parameter value consists of a list of values, each value must be separated by a COMMA character (ASCII decimal 44). A COMMA character in a property parameter value must be escaped with a BACKSLASH character (ASCII decimal 92).

Structured property values must have their components separated by a SEMICOLON character (ASCII decimal 59). In addition, lists of property parameters must be separated by a SEMICOLON character (ASCII decimal 59). A SEMICOLON character in a property value or property parameter value must be escaped with a BACKSLASH character (ASCII decimal 92).

For example, in the following properties a SEMICOLON is used to separate property parameters and property value fields. A COMMA is used to separate values.

ATTENDEE;RSVP=TRUE;ROLE=ATTENDEE:"J.Smith" <jsmith@host.com)
RDATE;VALUE=DATE:19970304,19970504,19970704,19970904

[5.1.3](#) Multiple Values

Each attribute or property defined in the iCalendar object may have multiple values, if allowed in the definition of the specific property. The general rule for encoding multi-valued items is to simply create a new content line for each value; including the property name. However, it should be noted that some properties support encoding multiple values in a single property by separating the values with a COMMA (ASCII decimal 44).

Dawson/Stenerson

14

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

[5.1.4](#) Character Set

The default character set is [[UTF-8](#)]. For transport in a MIME entity, the "charset" Content-Type parameter may be used to set the default character set for the entire MIME body part.

[5.1.5](#) Language

The "language" property parameter should be used to identify data in alternate languages. The default language is "us-EN". The value of the language property parameter is that defined in [[RFC 1766](#)].

Note: For transport in a MIME entity, the Content-Language header field may be used to set the default language for the entire body part.

[5.1.6](#) Content Encoding

The "encoding" property parameter should be used to specify an alternate encoding for a value. If the value contains a <CR> character (ASCII decimal 10) or <LF> character (ASCII decimal 13), it must be encoded using either "Q" or "B" encoding, since <CR><LF> is used to separate lines in the iCalendar object itself.

[5.1.7](#) Binary Content

There is no support for inline encoding of binary information in an iCalendar object. Binary information is associated with the iCalendar

object through the use of a uniform resource locator (URL) reference to the binary information.

5.1.8 Recurrence Set

Recurring events and to-dos are supported by this specification. The recurrence within the iCalendar object may be specified as either a list of discrete date and time values or as a recurrence rule. The full recurrence set is generated by considering the initial DTSTART along with the RRULE, RDATE, EXDATE and EXRULE properties contained within the iCalendar object. The DTSTART defines the first instance in the recurrence set. Multiple instances of the RRULE and EXRULE properties may also be specified to define more sophisticated recurrence sets. The final recurrence set is generated by gathering all of the start date-times generated by any of the specified RRULE and RDATE properties, and excluding any start date and times which fall within the union of start date and times generated by any specified EXRULE and EXDATE properties. This implies that start date and times within exclusion related properties (i.e., EXDATE and EXRULE) take precedence over those specified by inclusion properties (i.e., RDATE and RRULE). Where duplicate instances are generated by the RRULE and RDATE specification, only one recurrence is considered. Duplicate instances are ignored.

The recurrence rule used in the iCalendar object is defined in the RRULE component property.

Dawson/Stenerson

15

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

5.1.9 Data Types

The "value" property parameter is an optional property parameter. It is used to identify the data type and format of the property value. The values of a given instance of a property must only be of a single data type. For example, a RDATE property can not have a combination of DATE-TIME and TIME values. The following data types are used by the iCalendar object.

5.1.9.1 Boolean

The "boolean" data type is used to identify properties that contain either a "true" or a "false" boolean value. These values are case insensitive. The data type is defined by the following notation:

boolean = "TRUE" / "FALSE"

For example, any of the following are equivalent:

TRUE
true
TrUe

[5.1.9.2](#) Calendar User Address

The "cal-address" data type is used to identify properties that contain an address of a calendar user. The phrase component of the address may be used to match an unknown address with an otherwise known individual, group, or resource. The data type is as defined by the following notation:

```
cal-address      = addr-spec / [phrase] "<" addr-spec ">"

addr-spec = local-part "@" domain ;RFC 822 style address
local-part = WORD *("." WORD)
domain     = domain-ref *("." domain-ref)
domain-ref = ATOM

phrase      = 1*WORD
WORD        = ATOM / quoted-string
quoted-string = <"> *(qtext/quoted-pair) <"> ; Regular qtext or
                                                ;   quoted chars.

qtext       = <any CHAR excepting <">,           ; => may be folded
              "\" & CR, and including linear-white-space>

quoted-pair = "\" CHAR ; may quote any char
CHAR        = <any a character from the selected character set>
ATOM        = 1*<any CHAR except specials, SPACE and CTLs>
```

[5.1.9.3](#) Date

The "date" data type is used to identify values that contain a calendar date. The format is expressed as the [ISO 8601] complete representation, basic format for a calendar date. The text format specifies a four-digit year, two-digit month, and two-digit day of the month. There are no separator characters between the year, month

and day component text. The data type is defined by the following notation:

```

DIGIT      =<any ASCII decimal digit>      ;0-9

date-fullyear      = 4DIGIT
date-month         = 2DIGIT      ;01-12
date-mday          = 2DIGIT      ;01-28, 01-29, 01-30, 01-31
                                   ;based on month/year

date              = date-fullyear date-month date-mday

```

For example, the following represents July 14, 1997:

```
19970714
```

[5.1.9.4](#) **Date-Time**

The "date-time" data type is used to identify values that contain a precise calendar date and time of day. The format is expressed as the [ISO 8601] complete representation, basic format for a calendar date and time of day. The text format is a concatenation of the "date", followed by the LATIN CAPITAL LETTER T character (ASCII decimal 84) time designator, followed by the "time" format defined above. The data type is defined by the following notation:

```
date-time = date "T" time ;As specified above in date and time
```

The following represents July 14, 1997, at 1:30 PM in UTC and the equivalent time in New York (five hours behind UTC), expressed as a local time and local time with UTC offset:

```

19970714T133000Z
19970714T083000
19970714T083000-0500

```

[5.1.9.5](#) **Duration**

The "duration" data type is used to identify properties that contain a duration of time. The format is expressed as the [ISO 8601] basic format for the duration of time. The format can represent durations in terms of years, months, days, hours, minutes, and seconds. The data type is defined by the following notation:

```

DIGIT      =<any ASCII decimal digit>      ;0-9

dur-second = 1*DIGIT "S"
dur-minute = 1*DIGIT "M" [dur-second]
dur-hour   = 1*DIGIT "H" [dur-minute]
dur-time    = "T" (dur-hour / dur-minute / dur-second)

dur-week    = 1*DIGIT "W"
dur-day      = 1*DIGIT "D"

```

```
dur-month  = 1*DIGIT "M" [dur-day]
dur-year   = 1*DIGIT "Y" [dur-month]
dur-date   = (dur-day / dur-month / dur-year) [dur-time]
```

Dawson/Stenerson

17

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

```
duration   = "P" (dur-date / dur-time / dur-week)
```

For example, a duration of 10 years, 3 months, 15 days, 5 hours, 30 minutes and 20 seconds would be:

```
P10Y3M15DT5H30M20S
```

[5.1.9.6](#) Float

The "float" data type is used to identify properties that contain a real value number value. If the property permits, multiple "float" values may be specified using a COMMA character (ASCII decimal 44) separator character. The data type is defined by the following notation:

```
DIGIT      =<any ASCII decimal digit>      ;0-9

float      = ["+" / "-"] *DIGIT ["." *DIGIT]
```

For example:

```
1000000.0000001
1.333
-3.14
```

[5.1.9.7](#) Integer

The "integer" data type is used to identify properties that contain a signed integer value. The valid range for "integer" is -2147483648 to 2147483647. If the sign is not specified, then the value is assumed to be positive. If the property permits, multiple "integer" values may be specified using a COMMA character (ASCII decimal 44) separator character. The data type is defined by the following notation:

```
DIGIT      =<any ASCII decimal digit>      ;0-9

integer    = ["+" / "-"] *DIGIT
```

For example:

```
1234567890
```


The "text" data type is used to identify values that contain human readable text. The character set and language in which the text is represented is controlled by the charset and language property parameters. The data type is defined by the following notation:

CHAR = <Any character in the selected character set>

5.1.9.10 Time

The "time" data type is used to identify values that contain a time of day. The format is expressed as the [ISO 8601] complete representation, basic format for a time of day. The text format consists of a two-digit 24-hour of the day (i.e., values 0-23), two-digit minute in the hour (i.e., values 0-59), and two-digit seconds in the minute (i.e., values 0-59). If seconds of the minute are not supported by an implementation, then a value of "00" should be specified for the seconds component. Fractions of an hour, minute or second are not supported by this format. This format is used to represent local time, local time with UTC offset and UTC time. UTC time is identified by a LATIN CAPITAL LETTER Z suffix character (ASCII decimal 90), the UTC designator, appended to the time. The local time with UTC offset is expressed as a local time, suffixed with the signed offset from UTC. The UTC offset is expressed as the 2-digit hours and 2-digit minutes difference from UTC. It is expressed as positive, with an optional leading PLUS SIGN character (ASCII decimal 43), if the local time is ahead of UTC. It is expressed as a negative, with a leading HYPHEN-MINUS character (ASCII decimal 45), if

Dawson/Stenerson

19

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

the local time is behind UTC. Local time has neither the UTC designator nor the UTC offset suffix text. The data type is defined by the following notation:

DIGIT = <any ASCII decimal digit> ;0-9

time-hour = 2DIGIT ;00-23

time-minute = 2DIGIT ;00-59

time-second = 2DIGIT ;00-59

time-numzone = ("+" / "-") time-hour time-minute

time-zone = "Z" / time-numzone

time = time-hour time-minute time-second [time-zone]

For example, the following represents 8:30 AM in New York, five hours behind UTC, in local time and local time with UTC offset. In addition, 1:30 PM in UTC is illustrated:

083000
083000-0500
133000Z

There are cases when a floating time is intended within a property value. For example, an event may be defined that indicates that an individual will be busy from 11:00 AM to 1:00 PM every day. In these cases, a local time may be specified. The recipient of an iCalendar object with a property value consisting of a local time, without any relative time zone information, should interpret the value as being fixed to the recipient's locale and time zone. In most cases, a fixed time is desired. To properly communicate a fixed time in a property value, either UTC, local time with UTC offset, or local time with a time zone calendar component must be specified.

5.1.9.11 URL

The "url" data type is used to identify values that contain a uniform resource locator (URL) type of reference to the property value. This data type might be used to reference binary information, for values that are large, or otherwise undesirable to include directly in the iCalendar object.

The URL value formats in [RFC 1738](#), [RFC 2111](#) and any other IETF registered value format may be specified.

The data type is defined by the following notation:

```
url          = <As defined by any IETF RFC>
```

Any IANA registered URL type may be used. These include, but are not limited to, those for FTP and HTTP protocols, file access, content identifier and message identifier.

For example, the following is an URL for a local file:

```
file:///my-report.txt
```

Dawson/Stenerson

20

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

```
text      = <Any CHAR, including bare CR & bare LF but not
           including CRLF>
```

5.1.9.12 UTC Offset

The "utc -offset" data type is used to identify properties that contain an offset from UTC to local time. The data type is defined by the following notation:

```
utc-offset = time-numzone ;As defined above in time data type
```

For example, the following are UTC offsets for New York (five hours behind UTC) and Geneva (one hour ahead of UTC):

```
-0500      ;New York
+0100      ;Geneva
```

[5.2](#) iCalendar Object

The Calendaring and Scheduling Core Object is a collection of calendaring and scheduling information. Typically, this information will consist of a single iCalendar object. However, multiple iCalendar objects may be sequentially, grouped together. The first line and last line of the iCalendar object must contain a pair of iCalendar object delimiter strings. The syntax for an iCalendar object is as follows:

```
icalobject = "BEGIN" ":" "VCALENDAR" CRLF
            icalbody
            "END" ":" "VCALENDAR" CRLF [icalobject]
```

The following is a simple example of an iCalendar object:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
BEGIN:VEVENT
DTSTART:19970714T120000-0500
DTEND:19970714T235959-0500
DESCRIPTION:Bastille Day Party
END:VEVENT
END:VCALENDAR
```

[5.3](#) Property

A property is the definition of an individual attribute describing a calendar property or a calendar component. A property takes the following form:

```
property    = proptime [";" parmlist] ":" value CRLF

proptime    = <any properties defined in this document>
              / iana-prop / x-token

x-token     = <The two characters "X-" or "x-" followed, with no
```

intervening white space, by any atom>

Dawson/Stenerson

21

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

iana-prop = <A publicly defined extension property, registered
with IANA, as specified by this document>

The following is an example of a property:

DTSTART:19960415T083000-05:00

This document places no imposed ordering of properties within an
iCalendar object.

Property names, parameter names and parameter values (i.e.,
everything to the left of the ":" on a line) are case insensitive.
For example, the property name "DUE" is the same as "due" and "Due".

5.4 Calendar Components

The body of the iCalendar object consists of a sequence of calendar
properties and one or more calendar components. The calendar
properties are attributes that apply to the calendar as a whole. The
calendar components are collections of properties that with a
particular calendar semantic. For example, the calendar component may
specify a an event, a to-do, journal entry, time zone information, or
free/busy time information, or alarm.

The body of the iCalendar Object is defined by the following notation:

```
icalbody    = calprops 1*component

calprops    = [calscale] prodid [profile] [profile-version]
              [source] [name] version

component   = 1*(eventc / todoc / journalc / freebusyc /
              / timezonec)
```

5.4.1 Event Component

An Event Calendar Component is a grouping of component properties and
an optional alarm calendar component that represent a scheduled
amount of time on a calendar. For example, it may be an activity;
such as a one-hour, department meeting from 8:00 AM to 9:00 AM,
tomorrow.

An Event Component is defined by the following notation:

```

eventc      = "BEGIN" ":" "VEVENT" CRLF
              *eventprop *alarmc
              "END" ":" "VEVENT" CRLF

eventprop   = *attach *attendee *categories [class] [created]
              description [dtend] dtstart *exdate *exrule
              [geo] *last-mod [location] [priority] [rstatus]
              *related *resources *rdate *rrule dtstamp
              [resp-seq] [seq] [status] [summary] [transp]
              uid *url [recurid] *(comment)

```

Dawson/Stenerson

22

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

The Event Component can not be nested within another Calendar Component. Event components may be related to each other or to a To-do or Journal Calendar Component with the RELATED-TO property.

The following is an example of the Event Calendar Component:

```

BEGIN:VEVENT
DTSTART:19970903T083000-0800
DTEND:19970903T110000-0800
DESCRIPTION:Annual Employee Review
CLASS:PRIVATE
CATEGORIES:BUSINESS,HUMAN RESOURCES
END:VEVENT

```

[5.4.2](#) To-do Component

A To-do Calendar Component is a grouping of component properties and an optional alarm calendar component that represent 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".

A To-do Component is defined by the following notation:

```

todoc       = "BEGIN" ":" "VTODO" CRLF
              *todoprop *alarmc
              "END" ":" "VTODO" CRLF

todoprop    = *attach *attendee *categories [class] [completed]
              [created] description dtstamp dtstart due *exdate
              *exrule [geo] *last-mod [location] priority [rstatus]
              *related *resources *rdate *rrule [resp-seq] [recurid]
              [seq] [status] [summary] [transp] uid *url *(comment)

```

The To-do Component can not be nested within another Calendar Component. If To-do components need to be related to each other or to an Event or Journal Calendar Component, they can specify a relationship with the RELATED-TO property.

The following is an example of a To-do Calendar Component:

```
BEGIN:VTODO
DTSTART:19970415T083000-0500
DUE:19970415T235959-0500
DESCRIPTION:1996 Income Tax Preparation
CLASS:CONFIDENTIAL
CATEGORIES:FAMILY,FINANCE
PRIORITY:1
STATUS:NEEDS ACTION
END:VEVENT
```

5.4.3 Journal Component

A Journal Calendar Component is a grouping of component properties that represent one or more descriptive text on a particular calendar date. For example, it may be a journal entry of individual telephone

Dawson/Stenerson	23	Expires January 1998
Internet Draft	C&S Core Object Specification	July 29, 1997

contacts for the day or an ordered list of accomplishments for the day.

A Journal Component is defined by the following notation:

```
journalc  = "BEGIN" ":" "VJOURNAL" CRLF
           *jourprop
           "END" ":" "VJOURNAL" CRLF

jourprop  = *attach *attendee *categories [class] [created]
           description dtstart dtstamp *last-mod *related
           [rdate] [rrule] [rstatus] [resp-seq] [seq] uid *url
           [recurid] *(comment)
```

The Journal Component can not be nested within another Calendar Component. If Journal Components need to be related to each other or to an Event or To-Do Calendar Component, they can specify a relationship with the RELATED-TO property.

The following is an example of the Journal Calendar Component:

```

BEGIN:VJOURNAL
DTSTART:19970317T083000
DESCRIPTION:1. Staff meeting: Participants include Joe, Lisa
and Bob. Aurora project plans were reviewed. There is currently
no budget reserves for this project. Lisa will escalate to
management. Next meeting on Tuesday.
2. Telephone Conference: ABC Corp. sales representative called
to discuss new printer. Promised to get us a demo by Friday.
3. Henry Miller (Handsoff Insurance): Car was totaled by tree.
Is looking into a loaner car. 654-2323 (tel).
END:VJOURNAL

```

5.4.4 Free/Busy Component

A Free/Busy Calendar Component is a grouping of component properties that represent free or busy time information. Typically, this component exists in an iCalendar object that is being used to either request or return free or busy time information.

A Free/Busy Component is defined by the following notation:

```

freebusyc = "BEGIN" ":" "VFREEBUSY" CRLF
           *fbprop
           "END" ":" "VFREEBUSY" CRLF

fbprop    = *attendee [created] [duration] [dtend] [dtstart]
           dtstamp #freebusy *last-mod *related [rstatus]
           [resp-seq] [seq] uid *url *(comment)

```

The Free/Busy Component can not be nested within another Calendar Component. Free/Busy components may be related to each other with the RELATED-TO property. Multiple Free/Busy Calendar Components may be specified within a iCalendar object. This permits the grouping of Free/Busy information into logical collections, such as monthly groups of busy time information.

The Free/Busy Calendar Component is intended for use in profiles involving requests for free time, requests for busy time, requests for both free and busy, and the associated replies.

Free/Busy information can be expressed using the FREEBUSY property. This property provides a terse representation of time periods. One or more FREEBUSY properties may be specified in the FREE/BUSY Calendar Component to describe the Free/Busy information.

Optionally, the DTSTART and DTEND properties may be specified to express the start and end date and time for Free/Busy information in the Free/Busy Calendar Component. When present in a Free/Busy Calendar Component, they should be specified prior to any FREEBUSY properties. In a free time request, these properties may be used in combination with the DURATION property to express a request for a duration of free time within a given window of time.

The recurrence properties (RRULE, EXRULE, RDATE, EXDATE) are not permitted within a Free/Busy Calendar Component. Any recurring events are resolved into their individual busy time periods using the FREEBUSY property.

The following is an example of a Free/Busy Calendar Component:

```
BEGIN:VFREEBUSY
DTSTART:19971015T050000Z
DTEND:19971016T050000Z
FREEBUSY;VALUE=PERIOD-START:19971015T050000Z/PT8H30M,
  19971015T160000Z/PT5H30M, 19971015T223000Z/PT6H30M
END:VFREEBUSY
```

5.4.5 Alarm Component

An Alarm Calendar Component is a grouping of component properties that is a reminder or alarm for an event or a to-do. The Alarm Calendar Component may only be specified in an event or to-do calendar component. For example, it may define a reminder for a pending event or an overdue to-do.

The DTSTART property specifies the calendar date and time of day that the alarm will be triggered. The value may alternately be set to a period of time, before or after the event or to-do, that the alarm will be triggered.

An Alarm Component is defined by the following notation:

```
alarmc      = "BEGIN" ":" "VALARM" CRLF
              *alarmprop
              "END" ":" "VALARM" CRLF

alarmprop   = *attach [created] [description] dtstart duration
              / *last-mod *related repeat [summary] *(comment)
```

The Alarm Component can only appear within either an Event or To-Do Calendar Component. Alarm Components can not be nested.

The following is an example of the Alarm Calendar Component:

```
BEGIN:VALARM
DTSTART:19970317T133000Z
REPEAT:4
DURATION:PT15M
CATEGORIES:DISPLAY,AUDIO
ATTACH:file:///mmedia/sounds/bell1.wav
DESCRIPTION:Breakfast meeting with executive team at 8:30 AM
END:VALARM
```

[5.4.6](#) Timezone Component

A time zone is unambiguously defined by the set of time measurement rules determined by the governing body for a given geographic area. These rules describe at a minimum the base offset from UTC for the time zone, often referred to as the Standard Time offset. Many locations adjust their Standard Time forward or backward by one hour, in order to accommodate seasonal changes in number of daylight hours, often referred to as Daylight Saving Time. Some locations adjust their time by a fraction of an hour. Standard Time is also known as Winter Time. Daylight Saving Time is also known as Advanced Time, Summer Time, or Legal Time in certain countries. The following table shows the changes in time zone rules for the eastern United States.

Effective Date	Transition Rule (Date/Time)	Offset	Abbreviation
1920-1920	last Sun in Mar, 02:00	-0400	EDT
1920-1920	last Sun in Oct, 02:00	-0500	EST
1921-1966	last Sun in Apr, 02:00	-0400	EDT
1921-1954	last Sun in Sep, 02:00	-0500	EST
1955-1966	last Sun in Oct, 02:00	-0500	EST
1967-*	last Sun in Oct, 02:00	-0500	EST
1967-1973	last Sun in Apr, 02:00	-0400	EDT
1974-1974	Jan 6, 02:00	-0400	EDT
1975-1975	Feb 23, 02:00	-0400	EDT
1976-1986	last Sun in Apr, 02:00	-0400	EDT

1987-* first Sun in Apr, 02:00 -0400 EDT

Interoperability between two calendaring and scheduling applications, especially for recurring events and to-dos, is dependent on the ability to capture and convey date and time information in an unambiguous format. The specification of current time zone information is integral to this behavior.

Dawson/Stenerson

26

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

The Time Zone Calendar Component is a grouping of component properties that define a time zone description. The time zone description specifies the effective Standard Time or Daylight Savings Time rules for a particular time zone. The Timezone Component can not be nested within other Calendar Components. The Time Zone Component may be specified multiple times. If the Time Zone Component is missing, the recipient should assume all local times are relative to the recipient's time zone. The Time Zone Component should be specified in the iCalendar object before any other Calendar Components.

A Time Zone Component is defined by the following notation:

```
timezonec = "BEGIN" ":" "VTIMEZONE" CRLF
           *tzprop
           "END" ":" "VTIMEZONE" CRLF

tzprop    = [created] [daylight] dtstart [rdate / rrule]
           [tzname] tzoffset *(comment)
```

The Time Zone component is important for correct interpretation of individual as well as recurring calendar components that span a time zone transition. For example, from EST to EDT. The exception to this are calendar components that are considered floating (i.e., occurs at a particular local time no matter what time zone you are in). If the iCalendar object contains a non-floating calendar component that has a recurring date pattern (i.e., includes the RRULE property) or a list of date and local time values (i.e., includes the RDATE property), one or more Time Zone components must be specified, such that for the given range of the recurrence (i.e., the earliest instance to latest instance), there is valid time zone information for all instances. In other words, if all of the instances of the pattern is entirely within one offset observance, (e.g., all are in Standard Time), only one Time Zone Calendar Component need be present. If a time zone transition is crossed, then other Time Zone Components are needed. Further, if there are known changes to the rules for the time zone, even more Time Zone Components are needed.

Each Time Zone Component consists of several properties:

The `CREATED` property is a `DATE-TIME` value that indicates when the time zone description was created.

The `DAYLIGHT` property is a `BOOLEAN` value indicating Standard Time (`FALSE`) or Daylight Savings Time (`TRUE`). The default for `DAYLIGHT` is `FALSE` or Standard Time.

The `DTSTART` property in this usage is a fully specified `DATE-TIME` value, including the UTC offset, indicating the effective start date and time for the time zone information. For example, `19671029T020000-0400` represents the time at which the transition to Standard Time took effect in 1967 for the eastern United States.

The `TZOFFSET` property is a `UTC-OFFSET` value indicating the UTC offset for the time zone (Standard Time or Daylight Savings Time).

The `TZNAME` property is the customary name for the time zone.

Dawson/Stenerson

27

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

The `RRULE` property is a `TEXT` property indicating the recurrence rule for the transition to this time zone. For example, in the United States, the transition from Standard Time to Daylight Saving Time occurs on the first Sunday in April at 02:00. If a recurrence rule describing the transition is known to have an effective end date, the `UNTIL` recurrence rule parameter is used to specify that end date and time. If the recurrence rule for a particular observance (Daylight Saving Time) is changing, then the `UNTIL` of the first rule will be equal to the `DTSTART` for the replacement rule. See example below.

Alternatively, the `RDATE` property can be used. The `RDATE` property is a `DATE-TIME` property indicating the individual dates and times that the transition takes effect. The values supplied for `RDATE` for each Time Zone component must provide valid time zone information of all instances of the recurrence specified for the calendar component to which this time zone information is to be applied.

The following are examples of the Time Zone Calendar Component:

This is a simple example showing time zone information for the Eastern United States using `RDATE`. Note that this is only suitable for a recurring event that starts on or later than 1997, April 6, at 02:00:00 EST (i.e., the earliest effective transition date and time) and ends no later than 1998, April 7, 02:00:00 EST (i.e., latest valid date and time for EST in this scenario). For example, this can be used for a recurring event that occurs every Friday, 8am-9am,

starting June 1, 1997, ending Dec 31, 1997.

```
BEGIN:VTIMEZONE
DAYLIGHT:FALSE
RDATE:19971026T020000-0400
TZOFFSET:-0500
TZNAME:EST
END:VTIMEZONE
```

```
BEGIN:VTIMEZONE
DAYLIGHT:TRUE
RDATE:19970406T020000-0500
TZOFFSET:-0400
TZNAME:EDT
END:VTIMEZONE
```

This is a simple example showing the current time zone rules for the Eastern United States using a RRULE recurrence pattern. Note that there is no effective end date to either of the Standard Time or Daylight Time rules. This information would be valid for a recurring event starting today and continuing on into the known future.

```
BEGIN:VTIMEZONE
DAYLIGHT:FALSE
DTSTART:19671029T020000-0400
RRULE:BYDAY=-1SU;BYMONTH=10;FREQ=YEARLY
TZOFFSET:-0500
TZNAME:EST
END:VTIMEZONE
```

Dawson/Stenerson

28

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

```
BEGIN:VTIMEZONE
DAYLIGHT:TRUE
DTSTART:19870405T020000-0500
RRULE:BYDAY=1SU;BYMONTH=4;FREQ=YEARLY
TZOFFSET:-0400
TZNAME:EDT
END:VTIMEZONE
```

This is an example showing a fictitious set of rules for the Eastern United States, where the Daylight Time rule has an effective end date (i.e., after that date, Daylight Time is no longer observed).

```
BEGIN:VTIMEZONE
DAYLIGHT:FALSE
DTSTART:19671029T020000-0400
```

```
RRULE:BYDAY=-1SU;BYMONTH=10;FREQ=YEARLY
TZOFFSET:-0500
TZNAME:EST
END:VTIMEZONE
```

```
BEGIN:VTIMEZONE
DAYLIGHT:TRUE
DTSTART:19870405T020000-0500
RRULE:BYDAY=1SU;BYMONTH=4;FREQ=YEARLY;UNTIL=19981025T020000-0400
TZOFFSET:-0400
TZNAME:EDT
END:VTIMEZONE
```

This is an example showing a fictitious set of rules for the Eastern United States, where the first Daylight Time rule has an effective end date. There is a second Daylight Time rule that picks up where the other left off.

```
BEGIN:VTIMEZONE
DAYLIGHT:FALSE
DTSTART:19671029T020000-0400
RRULE:BYDAY=-1SU;BYMONTH=10;FREQ=YEARLY
TZOFFSET:-0500
TZNAME:EST
END:VTIMEZONE
```

```
BEGIN:VTIMEZONE
DAYLIGHT:TRUE
DTSTART:19870405T020000-0500
RRULE:BYDAY=1SU;BYMONTH=4;FREQ=YEARLY;UNTIL=19990404T020000-0500
TZOFFSET:-0400
TZNAME:EDT
END:VTIMEZONE
```

```
BEGIN:VTIMEZONE
DAYLIGHT:TRUE
DTSTART:19990404T020000-0500
RRULE:BYDAY=-1SU;BYMONTH=3;FREQ=YEARLY
TZOFFSET:-0400
TZNAME:EDT
END:VTIMEZONE
```

Dawson/Stenerson

29

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

5.5 Calendar Properties

The Calendar Properties are attributes that apply to the iCalendar

object, as a whole. These properties do not appear within a Calendar Component. They should be specified after the BEGIN:VCALENDAR properties and prior to any Calendar Component.

5.5.1 Calendar Scale

This property is identified by the property name CALSCALE. This property defines the calendar scale used for the calendar information specified in the iCalendar object. This specification is based on the Gregorian calendar scale. The Gregorian calendar scale is assumed if this property is not specified in the iCalendar object. It is expected that other calendar scales will be defined in other specifications or by future versions of this specification.

The property is defined by the following notation:

```
calscale    = "CALSCALE" ":" calvalue CRLF
calvalue    = "GREGORIAN" / iana-scale
iana-scale  = <Any other designator for a calendar scale
              registered with IANA>
```

The following is an example of this property:

```
CALSCALE:GREGORIAN
```

The data type for this property is TEXT.

5.5.2 Product Identifier

This property is identified by the property name PRODID. This property specifies the identifier for the product that created the iCalendar 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. This calendar property must be specified in the iCalendar object but can only appear once.

The property is defined by the following notation:

```
prodid      = "prodid" ":" pidvalue CRLF
pidvalue    = text
;Any text that describes the product and version
;and that is generally assured of being unique.
```

The following is an example of this property:

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

The data type for this property is TEXT.

Dawson/Stenerson

30

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

[5.5.3](#) Profile

This property is identified by the property name PROFILE. This property defines the usage profile associated with the calendar object. When used in a MIME message entity, the value of this property must be the same as the Content-Type profile parameter value. This property can only appear once within the iCalendar object.

The property is defined by the following notation:

```
profile      = "PROFILE" ":" profvalue CRLF
profvalue    = " component "-" action
component    = "EVENT" / "TODO" / "JOURNAL" / "FREEBUSY"
              / iana-component / x-token
action       = <Any IANA registered iCalendar action type.>
x-token      = <The two characters "X-" or "x-" followed, with no
              intervening white space, by any atom>
iana-component = <Any other component registered with IANA>
```

The following is an example of this property when the iCalendar object is used to request a meeting:

```
PROFILE:EVENT-REQUEST
```

In the event that this property is not specified, the usage profile is undefined. The data type for this property is TEXT.

[5.5.4](#) Profile Version

This property is identified by the property name PROFILE-VERSION. This property specifies the identifier corresponding to the highest version number or the minimum and maximum range of the usage profile that was used in constructing the iCalendar object. Values for this property are to be defined by registering an iCalendar usage profiles.

The property is defined by the following notation:

```
prof-version = "PROFILE-VERSION" ":" profvalue CRLF
profvalue    = iana-prfver
iana-prfver  = max-prfver / (min-prfver ";" max-prfver)
min-prfver   = <A IANA registered iCalendar profile identifier>
max-prfver   = <A IANA registered iCalendar profile identifier>
```

The following is an example of this property:

Dawson/Stenerson	31	Expires January 1998
------------------	----	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

```
PROFILE-VERSION:IPCS-1.0
```

The data type for this property is TEXT.

[5.5.5](#) Source

This property is identified by the property name SOURCE. This property is defined by the [MIME DIR] specification. In this specification, the property identifies the URL for the source of the iCalendar object. The URL is useful for accessing the iCalendar object using a calendar access protocol.

The property is defined by the following notation:

```
source      = "SOURCE" ":" url CRLF
```

The following are examples of this property:

```
SOURCE:http://xyz.corp.com/corp-cals/1997-events.or4
```

```
SOURCE:http://xyz.corp.com/calendars/~jdoe
```

The data type for this property is URL.

[5.5.6](#) Source Name

This property is identified by the property name NAME. This property is defined by the [MIME DIR] specification. The property identifies the displayable, presentation name for the source of the iCalendar object. The source name is a useful text to associate in the user-interface of an application with the value in the SOURCE property.

The property is defined by the following notation:

```
name          = "NAME" ":" text CRLF
```

The following is an example of this property:

```
NAME:1997 Events Calendar for XYZ Corporation
```

The data type for this property is TEXT.

[5.5.7](#) **Version**

This property is identified by the property name VERSION. This property specifies the identifier corresponding to the highest version number or the minimum and maximum range of the MIME Calendaring and Scheduling Content Type specification supported by the implementation that created the iCalendar object. A value of "2.0" corresponds to this specification. This calendar property must appear within the iCalendar object but can only appear once.

The property is defined by the following notation:

```
version       = "VERSION" ":" vvalue CRLF
```

Dawson/Stenerson	32	Expires January 1998
------------------	----	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

```
vvalue        = "2.0"           ;This specification  
               / maxver  
               / (minver ";" maxver)
```

```
minver        = <A IANA registered iCalendar profile identifier>  
               ;Minimum iCalendar version used to create the iCalendar object
```

```
maxver        = <A IANA registered iCalendar profile identifier>  
               ;Maximum iCalendar version used to create the iCalendar object
```

The following is an example of this property:

```
VERSION:2.0
```

The data type for this property is TEXT.

[5.6](#) **Component Properties**

The following properties apply to either an event or to-do calendar object component.

5.6.1 Attachment

This property is identified by the property name ATTACH. The property provides the capability to associate an external object with a calendar component. For example, a document to be reviewed at a scheduled event or the description of the process steps for a to-do. The property may only be specified within event, to-do, or journal calendar components. This property may be specified multiple times within an iCalendar object.

The property is defined by the following notation:

```
attach      = "ATTACH" ":" url CRLF
```

The following are examples of this property:

```
ATTACH:CID:jsmith.part3.960817T083000.xyzMail@host1.com
```

```
ATTACH:FTP://xyzCorp.com/pub/reports/r-960812.ps
```

The data type for this property is URL.

5.6.2 Attendee

This property is identified by the property name ATTENDEE. The property defines an attendee within a calendar component. The property may only be specified within the event, to-do and free/busy calendar components.

The property has the property parameters TYPE, for the type of attendee, ROLE, for the intended role of the attendee; STATUS, for the status of the attendee's participation; RSVP, for indicating whether the favor of a reply is requested; EXPECT, to indicate the expectation of the attendee's participation by the originator; MEMBER, to indicate the group that the attendee belongs to;

Dawson/Stenerson	33	Expires January 1998
------------------	----	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

DELEGATED-TO, to indicate the person that the original request was delegated to; and DELEGATED-FROM, to indicate whom the request was delegated from.

A recipient delegated a request MUST inherit the RSVP and EXPECT values from the attendee that delegated the request to them.

Multiple attendees may be specified by including multiple ATTENDEE

properties within the MIME calendaring entity.

The property data type default is CAL-ADDRESS. The property data type may also be set to URL. This provides a useful mechanism to allow more than just the address of the attendee to be referenced. For example, the property value may refer to a URL.

The property is defined by the following notation:

```
attendee    = "ATTENDEE" [";" attparamlist] ":"
              (cal-address / URL) CRLF
;Value must match default or explicit data type

attparamlist = attparam / attparamlist ";" attparam
              / paramlist / paramlist ";" attparam
              / paramlist ";" attparamlist ";"
              attparam

attparam     = typeparm / roleparm / statusparm / rsvpparm
              / expectparm / memberparm / deletoparm / delefromparm

typeparm     = "TYPE" "="
              ("INDIVIDUAL"   ; An individual
              / "GROUP"       ; A group of individuals
              / "RESOURCE"    ; A physical resource
              / "ROOM"        ; A room resource
              / "UNKNOWN")    ; Otherwise not known
;Default value is UNKNOWN

roleparm     = "ROLE" "="
              ("ATTENDEE"     ; Indicates a regular attendee
              / "OWNER"       ; Indicates owner of event or to-do
              / "ORGANIZER"   ; Indicates organizer of event or to-do
              / "DELEGATE")   ; Indicates delegate to event or to-do
;Default is ATTENDEE

statusparm   = "STATUS" "="
              ("NEEDS-ACTION" ; Indicates event or to-do needs action
              / "ACCEPTED"    ; Indicates event or to-do accepted
              / "DECLINED"    ; Indicates event or to-do not accepted
              / "TENTATIVE"   ; Indicates event or to-do tentatively
              ; accepted. Status may change in the future.
              / "COMPLETED"   ; Indicates to-do was completed.
              ; COMPLETED property has date/time completed.
              / "DELEGATED"   ; Indicates event or to-do delegated
              ; to another ATTENDEE
              / "CANCELED")   ; Indicates event or to-do canceled for
                              ; ATTENDEE
;Default is NEEDS-ACTION
```

```

rsvpparm    = "RSVP" "="
              ("TRUE"           ; Indicates response requested
              / "FALSE")       ; Indicates no response needed
;Default is FALSE

expectparm  = "EXPECT" "="
              ("FYI"           ; Indicates request is for your info
              / "REQUIRE"     ; Indicates presence is required
              / "REQUEST")     ; Indicates presence is requested
;Default is FYI

memberparm  = "MEMBER" "=" cal-address
; Indicates a group or mailing list

deletoparm  = "DELEGATED-TO" "=" cal-address
; Indicates who request delegated to

delefromparm = "DELEGATED-FROM" "=" cal-address
;Indicates who request delegated from

```

The following are examples of this property s use for a to-do:

```

ATTENDEE;ROLE=OWNER;STATUS=COMPLETED:jsmith@host1.com
ATTENDEE;MEMBER=DEV-GROUP:joecool@host2.com
ATTENDEE;DELEGATED-FROM=immud@host3.com:ildoit@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:
http://www.xyz.com/~myvcard.vcf

```

The following is an example of this property with "delegatee" and "delegator" information for an event:

```

ATTENDEE;ROLE=OWNER;STATUS=ACCEPTED:John Smith <jsmith@host1.com>
ATTENDEE;ROLE=DELEGATE;STATUS=TENTATIVE;DELEGATED-FROM=
  iamboss@host2.com:Henry Cabot<hcabot@host2.com>
ATTENDEE;ROLE=ATTENDEE;STATUS=DELEGATED;DELEGATED-TO=
  hcabot@host2.com=iamboss(The Big Cheese)@host2.com

```

ATTENDEE;ROLE=DELEGATE;STATUS=ACCEPTED:Jane Doe <jdoe@host1.com>

The default data type for this property is CAL-ADDRESS. The data type may be reset to URL; in which case the value is a location or message that contains the information that is to be used to specify the attendee address.

Dawson/Stenerson

35

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

5.6.3 Categories

This property is identified by the property name CATEGORIES. This property defines the categories for a calendar component. The property may be specified within the event, to-do or journal calendar component with an arbitrary text value. The property may also be specified within the alarm property with a value of the alarm category. More than one category may be specified as a list of categories separated by the COMMA character (ASCII decimal 44).

The properties is defined by the following notation:

```
categories = "CATEGORIES" [";" paramlist] ":"  
            catvalue CRLF
```

```
catvalue    = cat1value *["," cat1value]  
            / cat2value *["," cat2value]
```

```
cat1value    = "APPOINTMENT" / "BUSINESS" / "EDUCATION" / "HOLIDAY"  
            / "MEETING" / "MISCELLANEOUS" / "NON-WORKING HOURS"  
            / "NOT IN OFFICE" / "PERSONAL" / "PHONE CALL"  
            / "SICK DAY" / "SPECIAL OCCASION" / "TRAVEL"  
            / "VACATION" / word
```

;Used in event and to-do components only

```
cat2value    = "AUDIO" / "DISPLAY" / "EMAIL" / "PROCEDURE"  
            / x-token / iana-word
```

;Used in alarm component only

The following are examples of this property in an event, to-do or journal calendar component:

```
CATEGORIES:APPOINTMENT,EDUCATION
```

```
CATEGORIES:MEETING
```

The following are examples of this property in an alarm calendar

component:

CATEGORIES:AUDIO,DISPLAY

CATEGORIES:PROCEDURE

The data type for this property is TEXT.

5.6.4 Classification

This property is identified by the property name CLASS. This property defines the access classification for a calendar component. The property may only be specified in an event, to-do or journal calendar component. The property may only be specified once.

An 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

Dawson/Stenerson 36 Expires January 1998

Internet Draft C&S Core Object Specification July 29, 1997

classification of an individual iCalendar component 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 alone. Additionally, due to the "blind" nature of most exchange processes using this specification, these access classifications can not serve as an enforcement statement for a system receiving an iCalendar object . Rather, they provide a method for capturing the intention of the calendar owner for the access to the calendar component.

The property is defined by the following notation:

```
class      = "CLASS" [";" paramlist] ":"  
            classvalue CRLF
```

```
classvalue = "PUBLIC" / "PRIVATE" / "CONFIDENTIAL" / x-token  
;Default is PUBLIC
```

The following is an example of this property:

CLASS:PUBLIC

The data type for this property is TEXT.

5.6.5 Comment

This property is identified by the property name COMMENT. This property specifies non-processing information intended to provide a comment to the calendar user. The property may be specified in any of the calendar components. The property may be specified multiple times.

The property is defined by the following notation:

```
comment      = "COMMENT" ":" text CRLF
```

The following is an example of this property:

```
COMMENT:The meeting really needs to include both ourselves
and the customer. We can t hold this meeting without them.
As a matter of fact, the venue for the meeting ought to be at
their site. - - John
```

The data type for this property is TEXT.

[5.6.6](#) **Date/Time Completed**

This property is identified by the property name COMPLETED. This property defines the date and time that a to-do was actually completed. The property may be specified once in a to-do component. The date and time is a UTC value.

The property is defined by the following notation:

```
completed    = "COMPLETED" ":" date-time CRLF
```

Dawson/Stenerson	37	Expires January 1998
Internet Draft	C&S Core Object Specification	July 29, 1997

The following is an example of this property:

```
COMPLETED:19960401T235959Z
```

This property is optional for MIME entities conforming to this content type. The data type for this property is DATE-TIME.

[5.6.7](#) **Date/Time Created**

This property is identified by the property name CREATED. This property specifies the date and time that the calendar information was created. The property may be specified in any of the calendar components. The property may only be specified once. The date and time is an UTC value.

The property is defined by the following notation:

```
created      = "CREATED" ":" date-time CRLF
```

The following is an example of this property:

```
CREATED:19960329T133000Z
```

The data type for this property is DATE-TIME.

5.6.8 Date/Time Due

This property is identified by the property name DUE. This property defines the date and time that a to-do is expected to be completed. The value must be later in time than the value for the DTSTART property. The time can either be in local time, local time with UTC offset or UTC time. The property must be specified in a to-do calendar component, but may only be specified once. The DUE value must be a date/time after the DTSTART value.

The property is defined by the following notation:

```
due          = "DUE" ":" (date-time / duration) CRLF
;Value data type must match the value
```

The following is an example of this property:

```
DUE:19960401T235959Z
```

The default data type for this property is DATE-TIME. The data type may be reset to DURATION.

5.6.9 Date/Time End

This property is identified by the property name DTEND. This property may be specified within the event, free/busy, and time zone calendar components.

Within the event calendar component, this property defines the end date and time for the event. The property is required in event calendar components. The time can either be in local time, local time

Dawson/Stenerson 38 Expires January 1998

Internet Draft C&S Core Object Specification July 29, 1997

with UTC offset or UTC time. The local time is only to be used to specify date and time values that do not need to be fixed. A recipient must assume their own time zone for data and time values

that do not include time zone information. Events may have an end date/time but no start date/time. In that case, the event does not take up any time. The value must be later in time than the value of the DTSTART property.

Within the free/busy calendar component, this property defines the end date and time for the free or busy time information. The time must be specified in local time with UTC offset or UTC time. The value must be later in time than the value of the DTSTART property.

The property is defined by the following notation:

```
dtend      = "DTEND" ":" date-time CRLF
```

The following is an example of this property:

```
DTEND:19960401T235959Z
```

The data type for this property is DATE-TIME.

5.6.10 Date/Time Stamp

This property is identified by the property name DTSTAMP. This property specifies an UTC date/time stamp. The property indicates the date/time that the iCalendar object instance was created. This property SHOULD be included in every iCalendar object to permit the recipient to know when the iCalendar object was created.

This property is different than the CREATED and LAST-MODIFIED properties. These two properties are used to specify when the calendar service information was created and last modified. This is different than when the iCalendar object representation of the calendar service information was created or last modified.

The property is defined by the following notation:

```
dtstamp    = "DTSTAMP" ":" date-time CRLF
```

The value type for this property is DATE-TIME. The value must be a UTC date/time value.

5.6.11 Date/Time Start

This property is identified by the property name DTSTART. This property may be specified within the event, free/busy, and time zone calendar components.

Within the event calendar component, this property defines the start date and time for the event. The property is required in event calendar components. The time can either be in local time, local time

with UTC offset or UTC time. The local time is only to be used to specify date and time values that do not need to be fixed. A recipient must assume their own time zone for data and time values

Dawson/Stenerson

39

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

that do not include time zone information. Events may have a start date/time but no end date/time. In that case, the event does not take up any time.

Within the free/busy calendar component, this property defines the start date and time for the free or busy time information. The time must be specified in local time with UTC offset or UTC time.

Within the time zone calendar component, this property defines the effective start date and time for a time zone specification. This property is required within time zone calendar components. The time must be specified as a UTC time.

The property is defined by the following notation:

```
dtstart      = "DTSTART" ":" (date-time / date) CRLF
;Date data type only permitted on Journal calendar component.
```

The following is an example of this property:

```
DTSTART:19960401T235959-0600
```

The default data type for this property is DATE-TIME. For Journal calendar components, the data type may be overridden to be DATE.

[5.6.12](#) Daylight

This property is identified by the property name DAYLIGHT. This property may only be specified in a Time Zone Calendar Component. This property specifies whether Daylight Saving Time (i.e., value is TRUE) or Standard Time (i.e., value is FALSE) is in effect for the time zone. The default value is FALSE or Standard Time.

The property is defined by the following notation:

```
daylight     = "DAYLIGHT" ":" boolean CRLF
;Default value is FALSE
```

The following is an example of this property:

```
DAYLIGHT:TRUE           ;Specifies DST in effect in time zone
```

The data type for this property is `BOOLEAN`.

5.6.13 Description

This property is identified by the property name DESCRIPTION. This property provides a more complete description of the calendar component, than that provided by the SUMMARY property. The property must be specified in the event, to-do and journal calendar components. The property may be specified multiple times only within a journal calendar component.

The property is defined by the following notation:

Dawson/Stenerson	40	Expires January 1998
------------------	----	----------------------

Internet Draft C&S Core Object Specification July 29, 1997

```
Description      = "DESCRIPTION" [";" paramlist] ":"
                  text CRLF
```

The following is an example of the property with formatted line breaks in the property value:

```
DESCRIPTION;ENCODING=Q: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 example 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.

The data type for this property is TEXT.

5.6.14 Duration

This property is identified by the property name DURATION. The property specifies a duration of time. The property may be specified in an event calendar component in order to specify a duration of the event, instead of an explicit end date/time. The property may be specified in a free/busy calendar component in order to specify the amount of free time being requested. The property may be specified in an alarm calendar component in order to specify the period between repeating alarms.

The property is defined by the following notation:

```
duration    = "DURATION" ":" duration CRLF
```

The following is an example of this property that specifies an interval of time of 1 hour and zero minutes and zero seconds:

```
DURATION:PT1H0M0S
```

The following is an example of this property that specifies an interval of time of 15 minutes.

```
DURATION:PT15M
```

The data type for this property is DURATION.

[5.6.15](#) Exception Date/Times

This property is identified by the property name EXDATE. This property defines the list of date/time exceptions for a recurring event or to-do component. The times can either be in local time, local time with UTC offset or UTC time.

The property is defined by the following notation:

Dawson/Stenerson	41	Expires January 1998
------------------	----	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

```
exdate      = "EXDATE" ":" date-time *["," date-time]  
CRLF
```

The following is an example of this property:

```
EXDATE:19960402T010000Z,19960403T010000Z,19960404T010000Z
```

The data type for this property is DATE-TIME.

[5.6.16](#) 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 event or to-do. This property may only be specified in the event and to-do calendar components.

This property is defined by the same property values and parameters as specified for the RRULE property. The property is defined by the following notation:

extrule = "EXRULE" [";" paramlist] ":" rvalue CRLF

The following are examples of this property. Except every other week, on Tuesday and Thursday for 4 occurrences:

EXRULE:COUNT=4;INTERVAL=2;BYDAY=TU,TH;FREQ=WEEKLY

Except daily for 10 occurrences:

EXRULE:COUNT=10;FREQ=DAILY

Except yearly in June and July for 8 occurrences:

EXRULE:COUNT=8;BYMONTH=6,7;FREQ=YEARLY

The data type for this property is TEXT.

[5.6.17](#) Free/Busy Time

This property is identified by the property name FREEBUSY. The property defines one or more free or busy time intervals. These time periods may be specified as either a start and end date-time or a start date-time and duration.

The date and time is either local time with UTC offset or a UTC value.

The FREEBUSY property may include the TYPE property parameter to specify the information defines a free or busy time interval. The property may also include the STATUS property parameter to specify the type of busy time. The STATUS parameter may be utilized by the application reading the busy time information in order to provide a richer view of the information.

The property is defined by the following notation:

Dawson/Stenerson 42 Expires January 1998

Internet Draft C&S Core Object Specification July 29, 1997

freebusy = "FREEBUSY" [";" fbparamlist] ":" fbvalue
CRLF

fbparamlist = fbparam / paramlist ";" fbparam
/ fbparam ";" fbparamlist

fbparam = fbtype / fbstatus

fbtype = "TYPE" "=" ("FREE" or "BUSY")

;Default is BUSY

```
fbstatus    = "STATUS" "="  
              "BUSY"          ;Represents busy time interval  
              / "OUT"         ;Represents out-of-office, non-working  
                              ;hours, or other unavailable interval  
              / "PRIVATE"     ;Represents private unavailable time  
              / "CONFIDENTIAL";Represents confidential unavailable  
                              ;time  
;Default is BUSY
```

```
fbvalue     = period *["," period]  
;Value must match default or explicit data type
```

The following are some examples of this property:

```
FREEBUSY;STATUS=OUT:19970308T160000Z/PT8H30M
```

```
FREEBUSY;TYPE=FREE:19970308T160000Z/PT3H,19970308T200000Z/PT1H
```

FREEBUSY properties within the Free/Busy Calendar Component should be sorted in ascending order, based on start time and then end time, with the earliest periods first.

The FREEBUSY property may specify more than one value, separated by the COMMA character (ASCII decimal 44). In such cases, the FREEBUSY property values should all be of the same STATUS (e.g., all values of a particular STATUS listed together in a single property).

The data type for this property is PERIOD.

5.6.18 Geographic Position

This property is identified by the property name GEO. This property specifies information related to the global position for an event or to-do calendar component. The property value specifies latitude and longitude, in that order (i.e., "LAT LON" ordering). 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 longitude and latitude values must be specified as decimal degrees and should be specified to six decimal places. This will allow for granularity within a meter of the geographical position. The simple formula for converting degrees-minutes-seconds into decimal degrees is:

$$\text{decimal} = \text{degrees} + \text{minutes}/60 + \text{seconds}/3600.$$

The property is defined by the following notation:

```
geo          = "GEO" ":" geovalue CRLF
```

```
geovalue     = float ";" float  
;Latitude and Longitude components
```

The following is an example of this property:

```
GEO:37.386013;-122.082932
```

The default data type for this property is FLOAT.

[5.6.19](#) Last Modified

This property is identified by the property name LAST-MODIFIED. The property specifies the date and time that the calendar information was last revised. The property value may include multiple "date-time" values in order to capture the sequence of modifications made to the calendar information. This property may be specified in the event, to-do, journal or free/busy calendar components. The data and time must be a UTC value.

The property is defined by the following notation:

```
last-mod     = "LAST-MODIFIED" ":" date-time ["," date-time]  
CRLF
```

The following are examples of this property:

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

```
LAST-MODIFIED:19970104T083000-0500,19970403T090000-0500,  
19970901T133000-0400
```

The data type for this property is DATE-TIME.

[5.6.20](#) Location

This property is identified by the property name LOCATION. The property defines the intended location for the event or to-do calendar component. The property may only be specified within an event or to-do calendar component.

The property is defined by the following notation:

```
location     = "LOCATION" [";" paramlist] ":" locavalue  
CRLF
```

locavalue = text / url ;The value must be the same type as the
;default or explicit data type.

The following are some examples of this property:

LOCATION:Conference Room - F123, Bldg. 002

Dawson/Stenerson 44 Expires January 1998

Internet Draft C&S Core Object Specification July 29, 1997

LOCATION;VALUE=URL:http://www.xyzcorp.com/~jsmith.vcf

The default data type for this property is TEXT. The data type may be reset to URL. In the case of the data type being URL, the property value may reference a vCard object. This provides a useful mechanism to specify a location in terms of its electronic business card.

5.6.21 Priority

This property is identified by the property name PRIORITY. The property defines the priority for event or to-do. The property may only be specified within an event or to-do calendar component. The value is an integer. A value of zero (ASCII decimal 48) specifies an undefined priority. A value of one (ASCII decimal 49) is the highest priority. A value of two (ASCII decimal 50) is the second highest priority. Subsequent numbers specify a decreasing ordinal priority.

The property is specified by the following notation:

priority = "PRIORITY" ":" integer CRLF
;Default is zero

The following is an example of this property:

PRIORITY:2

The data type for this property is INTEGER.

5.6.22 Recurrence Date/Times

This property is identified by the property name RDATE. This property defines the list of date/times for a recurring event, to-do or time zone calendar component. This property may appear along with the RRULE property to define an aggregate set of repeating occurrences. When they both appear in an iCalendar object, the recurring events are defined by the union of occurrences defined by both the RDATE and

RRULE. The times can either be in local time, local time with UTC offset or UTC based time. If local time is used, the TIMEZONE component must be included in the iCalendar object, otherwise the local time value will be interpreted relative to the time zone of the recipient. The period values for RDATE are specified using a specific start and a specific end basic format (period-explicit) or the period with a specific start and a specific duration basic format (period-start).

The property is defined by the following notation:

```
rdate      = "RDATE" ":" rdvalue *["," rdvalue] CRLF
```

```
rdvalue    = date-time / period
;Value must match the default or explicit data type
```

The following are examples of this property:

```
RDATE:19970714T083000-0400
```

Dawson/Stenerson	45	Expires January 1998
------------------	----	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

```
RDATE;VALUE=PERIOD:19960403T020000Z/19960403T040000Z,
19960404T010000Z/PT3H
```

```
RDATE;VALUE=DATE:19970101,19970120,19970217,19970421
19970526,19970704,19970901,19971014,19971128,19971129,19971225
```

The default data type for this property is DATE-TIME. The value may be reset to DATE or PERIOD.

5.6.23 Recurrence ID

This property is identified by the property name RECURRENCE-ID. This property identifies a specific instance of a recurring event, to-do or journal calendar component. The property value is the effective DTSTART value of the recurrence instance. The time of day component for the value must be either an UTC or a local time with UTC offset time format, unless the original calendar object was expressed as a floating calendar object; that is in local time with no timezone calendar component specified..

The date/time value is set to the time when the original recurrence instance would occur - meaning that if the intent is to change a Friday meeting to Thursday, the date/time is still set to the original Friday meeting.

Recurrence ID is used in conjunction with the UID property to identify a particular instance of a recurring event, to-do or journal.

The property is defined by the following notation:

```
recurid          = "RECURRENCE-ID" [";" rangeparm] ":" date-time
rangeparm        = "RANGE" "=" ("THISANDPRIOR" / "THISANDFUTURE")
```

The default value for the range parameter is the single recurrence instance only.

The following are examples of this property:

```
RECURRENCE-ID:19960401T235959Z
```

```
RECURRENCE-ID;RANGE=THISANDFUTURE:19960120T120000Z
```

5.6.24 Recurrence Rule

This property is identified by the property name RRULE. This property defines a rule or repeating pattern for a recurring events, to-dos, or time zone definitions. The property may be specified in the event, to-do, or time zone calendar components.

The property value is a structured value consisting of a list of one or more recurrence grammar components. Each component is defined by a NAME=VALUE pair. The components are separated from each other by the SEMICOLON character (ASCII decimal 59).

Dawson/Stenerson	46	Expires January 1998
------------------	----	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

The FREQ component identifies the type of recurrence rule. This component must be specified in the recurrence rule. Valid values include HOURLY, to specify repeating events based on an interval of an hour or more; DAILY, to specify repeating events based on an interval of a day or more; WEEKLY, to specify repeating events based on an interval of a week or more; MONTHLY, to specify repeating events based on an interval of a month or more; and YEARLY, to specify repeating events based on an interval of a year or more.

The INTERVAL component contains a positive integer representing how often the RRULE repeats. The default value is "1" or every hour for a HOURLY rule, every day for a DAILY rule, every week for a WEEKLY rule, every month for a MONTHLY rule and every year for a YEARLY

rule. For a HOURLY rule, the value may also be expressed as a duration value, specifying hours and minutes for the repeat interval. For example, PT1H30M, would represent a 1 hour and 30 minute repeat interval.

The UNTIL component defines a date-time value which bounds the RRULE. If not present, and the COUNT component is also not present, the RRULE is considered to repeat forever.

The COUNT component defines the number of occurrences at which to bound the RRULE. This component is ignored if the UNTIL property parameter is also present.

The BYDAY component specifies a COMMA character (ASCII decimal 44) separated list of days of the week; MO, indicates Monday; TU, indicates Tuesday; WE, indicates Wednesday; TH, indicates Thursday; FR, indicates Friday; SA, indicates Saturday; SU, indicates Sunday.

Each of these values may also be preceded by a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of the specific day within the MONTHLY or YEARLY RRULE. For example, within a MONTHLY rule, +1MO (or simply 1MO) represents the first Monday within the month, whereas -1MO represents the last Monday of the month.

The BYMONTHDAY component specifies a COMMA character (ASCII decimal 44) separated list of days of the month. Valid values are 1 to 31 or -31 to -1.

The BYYEARDAY component specifies a COMMA character (ASCII decimal 44) separated list of days of the year. Valid values are 1 to 366 or -366 to -1. For example, -1 represents the last day of the year (December 31st).

The BYSETPOS component specifies a COMMA character (ASCII decimal 44) separated list of values which corresponds to the nth occurrence within the set of events specified by the rule. Valid values are 1 to 366 or -366 to -1. It must only be used in conjunction with another Byxxx component. For example "the last work day of the month" could be represented as:

```
RRULE:BYDAY=MO,TU,WE,TH,FR;BYSETPOS=-1;FREQ=MONTHLY
```

The BYWEEKNO component specifies a comma separated list of weeks of the year. Valid values are 1 to 52. This corresponds to weeks

according to week numbering as defined in [ISO 8601]. That is, a week as "A seven day period within a calendar year, starting on a Monday and identified by its ordinal number within the year; the first calendar week of the year is the one that includes the first Thursday of that year." This property parameter is only valid for YEARLY rules.

The BYMONTH component specifies a comma separated list of months of the year. Valid values are 1 to 12.

The WKST property parameter specifies the day on which the workweek starts. Valid values are MO, TU, WE, TH, FR, SA and SU. This is significant when a WEEKLY RRULE has an interval greater than 1. The default value is MO.

If two different Byxxx components are specified within the RRULE, the recurrence occurrence must meet both criteria.

If Byxxx component values are found which are beyond the available scope (ie, BYMONTHDAY=-30 in February), they are simply ignored. If a positive range limit is beyond the available scope, it will be interpreted as -1. Likewise, if a negative range limits beyond the available scope, it will be interpreted as +1.

The RRULE property requires referencing the DTSTART, DTEND or DURATION properties in the iCalendar object to calculate the Event or To-do instances.

The DTSTART and DTEND pair or DTSTART and DURATION pair, specified within the iCalendar object defines the first instance of the recurrence. When used with a recurrence rule, the DTSTART and DTEND properties must be specified in local time and the appropriate set of TIMEZONE components must be included. For detail on the usage of the TIMEZONE component, see the Time Zone Calendar Component definition.

Any duration associated with the iCalendar object applies to all members of the generated recurrence. Any modified duration for specific recurrences would have to be explicitly specified using the RDATE property.

This property is defined by the following notation:

```
rrule      = "RRULE" [paramlist] ":" rvalue CRLF
```

```
paramlist  = param / paramlist ";" param
```

```
rvalue     = "FREQ" = freq  
            *("UNTIL" "=" enddate  
              / "COUNT" "=" interval  
              / "INTERVAL" "=" rinterval  
              / "BYDAY" "=" bdweekdaylist  
              / "BYMONTHDAY" "=" bmdaylist
```

```
/ "BYYEARDAY" "=" bydaylist
/ "BYSETPOS" "=" bsplist
/ "BYWEEKNO" "=" bwdaylist
```

Dawson/Stenerson

48

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

```
/ "BYMONTH" "=" bmlist
/ "WKST" "=" weekday
/ "X-" word "=" word)
```

freq = "HOURLY" / "DAILY" / "WEEKLY" / "YEARLY"

rinterval = interval ; For any rvalue
/ duration ; Only for rvalue = HOURLY

DIGIT = <any ASCII decimal digit> ;0-9

digits = 1*DIGIT

interval = digits

enddate = date ;A UTC value

plus = "+"

minus = "-"

ordmoday = 1*2digits ;1 to 31

ordwk = 1*2digits ;1 to 52

ordyrday = 1*3digits ;1 to 366

daynumber = (plus / minus) ordmoday

weekday = "SU" / "MO" / "TU" / "WE" / "TH" / "FR" / "SA"
;Corresponding to SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY,
;FRIDAY, SATURDAY and SUNDAY days of the week.

bdweekdaynum = [daynumber] weekday

bdweekdaylist = bdweekdaynum / bdweekdaynum ","
*(bdweekdaynum)

bmposday = [plus] ordmoday

bmnegday = minus ordmoday

bmdaylist = bmposday *("," bmposday / bmnegday)
/ bmnegday *("," bmnegday / bmposday)

```

byposday    = [plus] ordyrday

bynegday    = minus ordyrday

bydaylist   = byposday *(", " byposday / bynegday)
              / bynegday *(", " bynegday / byposday)

bsplist     = byposday *(", " byposday / bynegday)
              / bynegday *(", " bynegday / byposday)

bwposday    = [plus] ordwk

```

Dawson/Stenerson

49

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

```

bwnegday    = minus ordwk

bwdaylist   = bwposday *(", " bwposday / bwnegday)
              / bwnegday *(", " bwnegday / bwposday)

bmposmon     = 1*2digits      ;1 to 12

bmlist      = bmposmon *(", " bmposmon)

```

Examples of this property include the following. Daily for 10 occurrences:

```
RRULE:COUNT=10;FREQ=DAILY
```

Daily until 12/24/94:

```
RRULE:UNTIL=19941224T000000Z;FREQ=DAILY
```

Every other day - forever:

```
RRULE:INTERVAL=2;FREQ=DAILY
```

Every 10 days, 5 occurrences:

```
RRULE:COUNT=5;INTERVAL=10;FREQ=DAILY
```

Weekly for 10 occurrences

```
RRULE:COUNT=10;FREQ=WEEKLY
```

Weekly until 12/24/94

```
RRULE:UNTIL=19941224T000000Z;FREQ=WEEKLY
```

Every other week - forever:

RRULE:INTERVAL=2;WKST=SU;FREQ=WEEKLY

Weekly on Tuesday and Thursday for 5 weeks:

RRULE:INTERVAL=5;WKST=SU;BYDAY=TU,TH;FREQ=WEEKLY

Every other week on Monday, Wednesday and Friday until 12/24/94:

RRULE:INTERVAL=2;WKST=SU;BYDAY=MO,WE,FR;=UNTIL=19941224T000000Z;
FREQ=WEEKLY

Every other week on Tuesday and Thursday, for 8 occurrences:

RRULE:INTERVAL=2;WKST=SU;COUNT=8;BYDAY=TU,TH;FREQ=WEEKLY

Monthly on the 1st Friday for ten occurrences:

RRULE:COUNT=10;BYDAY=1FR;FREQ=MONTHLY

Monthly on the 1st Friday until 12/24/94:

Dawson/Stenerson

50

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

RRULE:UNTIL=19941224T000000Z;BYDAY=1FR;FREQ=MONTHLY

Every other month on the 1st and last Sunday of the month for
10 occurrences:

RRULE:COUNT=10;BYDAY=1SU, -1SU;FREQ=MONTHLY

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

RRULE:COUNT=6;BYDAY=-2MO;FREQ=MONTHLY

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

RRULE:BYMONTHDAY=-3;FREQ=MONTHLY

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

RRULE:COUNT=10;BYMONTHDAY=2,15;FREQ=MONTHLY

Monthly on the first and last day of the month for 10 occurrences:

RRULE:COUNT=10;BYMONTHDAY=1, -1;FREQ=MONTHLY

Every 18 months on the 10th thru 15th of the month for 10

occurrences:

RRULE:COUNT=10;INTERVAL=18;BYMONTHDAY=10,11,12,13,14,15;
FREQ=MONTHLY

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:

RRULE:COUNT=5;BYMONTHDAY=-2;FREQ=MONTHLY

Yearly in June and July for 10 occurrences:

RRULE:COUNT=10;BYMONTH=6,7;FREQ=YEARLY

Every other year on January, February, and March for 10 occurrences:

RRULE:COUNT=10;INTERVAL=2;BYMONTH=1,2,3;FREQ=YEARLY

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

RRULE:COUNT=10;INTERVAL=3;BYYEARDAY=1,100,200;FREQ=YEARLY

Every 20th Monday of the year, forever:

RRULE:BYDAY=20MO;FREQ=YEARLY

Monday of Week No. 20, forever:

RRULE:BYWEEKNO=20;BYDAY=MO;FREQ=YEARLY

Every Thursday in March, forever:

Dawson/Stenerson	51	Expires January 1998
------------------	----	----------------------

Internet Draft	C&S Core Object Specification	July 29, 1997
----------------	-------------------------------	---------------

RRULE:BYDAY=TH;BYMONTH=3;FREQ=YEARLY

Every Thursday, but only in the summer, forever:

RRULE:BYDAY=TH;BYMONTH=6,7,8;FREQ=YEARLY

Every Friday the 13th, forever:

RRULE:BYDAY=FR;BYMONTHDAY=13;FREQ=MONTHLY

The first Saturday that follows the first Sunday of the month, forever:

RRULE:BYDAY=SA;BYMONTHDAY=7,8,9,10,11,12,13;FREQ=MONTHLY

Every four years, the first Tuesday after a Monday in November, forever (U.S. Election day):

```
RRULE:INTERVAL=4;BYDAY=TU;BYMONTHDAY=7,8,9,10,11,12,13;  
FREQ=YEARLY
```

The 3rd instance into the month of any of Tuesday, Wednesday or Thursday, for the next 3 months:

```
RRULE:COUNT=3;BYDAY=TU,WE,TH;BYSETPOS=3;FREQ=MONTHLY
```

The 2nd to last weekday of the month"

```
RRULE:BYDAY=MO,TU,WE,TH,FR;BYSETPOS=-2;FREQ=MONTHLY
```

The data type for this property is TEXT.

5.6.25 Related To

This property is identified by the property name RELATED-TO. The property is used to represent relationships or references between one calendar component and another. The property may only be specified in the event, to-do and journal calendar components. The property value consists of the persistent, globally unique identifier of another MIME calendar component. This value would be represented in a MIME calendar component by the UID property.

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

Changes to a calendar component referenced by this property may impact the related calendar component. For example, if a group event changes its 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 components. It is up to the target calendar system to maintain any property implications of this relationship.

Dawson/Stenerson

52

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

The property is defined by the following notation:


```
related      = "RELATED-T0" [";" paramlist] ":" relvalue
              CRLF

relvalue     = text
```

The following is an example of this property:

```
RELATED-T0:<jsmith.part7.19960817T083000.xyzMail@host3.com>

RELATED-T0:19960401-080045-4000F192713-0052
```

The data type for this property is TEXT.

[5.6.26](#) Repeat Count

This property is identified by the property name REPEAT. This property defines the number of repetitions for an alarm.

The property is defined by the following notation:

```
repeatcnt    = "REPEAT" ":" integer CRLF
              ;Default is "1".
```

The following is an example of this property:

```
REPEAT:4
```

The data type for the property is INTEGER.

[5.6.27](#) Request Status

This property is identified by the property name REQUEST-STATUS. This property defines the status code returned for a scheduling request. This property is used to return status code information related to the processing of an associated iCalendar object. The data type for this property is TEXT.

The value consists of a short return status, a longer return status description, and optionally the offending data. The components of the value are separated by the SEMICOLON character (ASCII decimal 59).

The property is defined by the following notation:

```
Rstatus      = "REQUEST-STATUS" ":" statcode ";"
              statdesc [";" extdata]

Statcode     = 3*DIGIT
              ;Numeric return status code

Statdesc     = *WORD
              ;Textual status description
```

Internet Draft

C&S Core Object Specification

July 29, 1997

Extdata = *WORD
;Textual exception data. For example, the offending property
;name and value or complete property line.

The following are some examples of this property:

REQUEST-STATUS:200;Success

REQUEST-STATUS:301;Invalid property value;DTSTART\;96-Apr-01
;Note escapement of the colon character in property value.

REQUEST-STATUS:208; Success, repeating event ignored. Scheduled
as a single event.;RRULE:INTERVAL=2;FREQ=WEEKLY

REQUEST-STATUS:401;Event conflict. Date/time is busy.

REQUEST-STATUS:307;Invalid calendar user;ATTENDEE:
jsmith@host.com

The following are valid classes for the return status code.
Individual iCalendar profiles will define specific return status
codes for these classes.

Dawson/Stenerson

54

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

Short Return Status Code	Longer Return Status Description
--------------------------	----------------------------------

1xx

Preliminary success. This class of status code indicates that the request has been initially processed, but that completion is pending.

2xx

Successful. This class of status code indicates that the request was completed successfully. However, the exact status code may indicate that a fallback has been taken.

3xx

Client Error. This class of status code indicates that the request was not successful. The error is the result of either a syntax or a semantic error in the client formatted request. Request should not be retried until the condition in the request is corrected.

4xx

Scheduling Error. This class

of status code indicates that the request was not successful. The error is the result of a scheduling conflict with the information in the associated calendar.

5xx

Service Error. This class of status code indicates that the request was not successful. Some sort of error occurred within the calendaring and scheduling service, not directly related to the request itself.

[5.6.28](#) Resources

This property is identified by the property name RESOURCES. This property defines the equipment or resources needed for the event or to-do. The property value is an arbitrary text. The property may only

Dawson/Stenerson

55

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

be specified in the event or to-do calendar component. More than one resource may be specified as a list of resources separated by the COMMA character (ASCII decimal 44).

The property is defined by the following notation:

```
resource    = "RESOURCES" [";" paramlist] ":"  
              resvalist CRLF
```

```
resvalist   = resvalue / resvalue "," resvalist
```

```
resvalue    = "CATERING" / "CHAIRS" / "COMPUTER PROJECTOR"  
              / "EASEL" / "OVERHEAD PROJECTOR" / "SPEAKER PHONE"  
              / "TABLE" / "TV" / "VCR" / "VIDEO PHONE" / "VEHICLE"  
              / word
```

The following is an example of this property:

RESOURCES:EASEL,PROJECTOR,VCR

The data type for this property is TEXT.

5.6.29 Response Sequence Number

This property is identified by the property name RESPONSE-SEQUENCE. This property defines the revision sequence of the calendar component. The property may only be specified in an event, to-do, journal or free/busy calendar component. This property is needed to properly handle the receipt and processing of a sequence of MIME calendar components that have been delivered out of order. Such is the case for store-and-forward based transports. The first response to an a request is created with response sequence number of "0" (ASCII decimal 48). If the value is non-zero, it must be specified. It is incremented each time another reply is sent.

The property is defined by the following notation:

```
respseq    = "RESPONSE-SEQUENCE" ":" integer CRLF
;Default is "0".
```

The following is an example of this property:

```
RESPONSE-SEQUENCE:1
```

The data type for this property is INTEGER.

5.6.30 Sequence Number

This property is identified by the property name SEQUENCE. This property defines the revision sequence of the calendar component used in a request. The property may only be specified in an event, to-do, journal or free/busy calendar component. This property is needed to properly handle the receipt and processing of a sequence of MIME calendar components that have been delivered out of order. Such is the case for store-and-forward based transports. The first request is created with a sequence number of "0" (ASCII decimal 48). It is

Dawson/Stenerson

56

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

incremented each time the ORGANIZER or OWNER issues a revision to the request. A monotonic increment to the sequence number is caused by a change to one of the following properties by the Organizer or Owner:

- DTSTART
- DTEND

- LOCATION
- DUE

The property is defined by the following notation:

```
sequence    = "SEQUENCE" ":" integer CRLF
;Default is "0".
```

The following is an example of this property:

```
SEQUENCE:1
```

The data type for this property is INTEGER.

5.6.31 Status

This property is identified by the property name STATUS. This property defines the originator's view of the overall status for the calendar component. This property may only be specified in the event and to-do calendar components. When specified in an event calendar component, the property is used to specify the originator's view of the general consensus for the meeting. When specified in a group scheduled to-do, the property is used to specify the originator's view of the completion status for the to-do.

The property is defined by the following notation:

```
status      = "STATUS" [";" paramlist] ":" statvalue CRLF

statvalue   = "NEEDS ACTION"           ;Indicates to-do needs action.
              / "COMPLETED"           ;Indicates to-do completed
              / "TENTATIVE"            ;Indicates event is being
                                      ;tentatively scheduled
              / "CONFIRMED"            ;Indicates event is definite
              / "CANCELLED"            ;Indicates event was canceled
```

The following is an example of this property:

```
STATUS:TENTATIVE
```

The data type for this property is TEXT.

5.6.32 Summary

This property is identified by the property name SUMMARY. This property defines a short summary or subject for the calendar component. The property may only be specified in the event, to-do and alarm calendar component.

The property is defined by the following notation:

Internet Draft

C&S Core Object Specification

July 29, 1997

```
summary      = "SUMMARY" [";" paramlist] ":" text CRLF
```

The following is an example of this property:

```
SUMMARY:Department Party
```

The data type for this property is TEXT.

[5.6.33](#) Time Transparency

This property is identified by the property name TRANSP. This property defines whether an event is transparent or not to free/busy time searches. This property may only be specified in an event calendar component.

The property is specified by the following notation:

```
transp       = "TRANSP" [";" paramlist] ":" transvalue  
              CRLF
```

```
transvalue = "BUSY"           ;Opaque/blocks on free/busy searches  
              ;Default value is BUSY  
            / "OUT"           ;Opaque/blocks on free/busy searches  
            / "PRIVATE"       ;Opaque/blocks on free/busy searches  
            / "CONFIDENTIAL" ;Opaque/blocks on free/busy searches  
            / "TRANSPARENT" ;Transparent on free/time searches
```

The following is an example of this property for an event that is transparent or does not block on free/busy time searches:

```
TRANSP:TRANSPARENT
```

The following is an example of this property for an event that is opaque or blocks on free/busy time searches:

```
TRANSP:BUSY
```

The data type for this property is TEXT.

[5.6.34](#) Time Zone Name

This property is identified by the property name TZNAME. This property specifies the customary designation for a time zone descripton. This property may only be specified in the Time Zone Calendar Component.

This property is defined by the following notation:

```
tzname      = "TZNAME" [";" paramlist] ":" text CRLF
```

The following are examples of this property:

```
TZNAME: EST
```

```
TZNAME: PDT
```

Dawson/Stenerson

58

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

The data type for this property is TEXT.

[5.6.35](#) Time Zone Offset

This property is identified by the property name TZOFFSET. This property specifies the offset from UTC for a time zone. This property may only be specified in a Time Zone Calendar Component. A Time Zone Calendar Component must include this property. The property value is a signed numeric indicating the number of hours and possibly minutes from UTC. Positive numbers represents time zones east, or ahead of UTC. Negative numbers represents time zones west of, or behind UTC.

The property is defined by the following notation:

```
tzoffset    = "TZOFFSET" ":" utc-offset CRLF
```

The following are examples of this property:

```
TZOFFSET:-0500
```

```
TZOFFSET:+0530
```

The data type for this property is UTC-OFFSET.

[5.6.36](#) Uniform Resource Locator

This property is identified by the property name URL. This property defines a Uniform Resource Locator (URL) associated with the iCalendar object. This property may be specified in the event, to-do, journal, free/busy, and alarm calendar components.

The property is defined by the following notation:

```
url         = "URL" ":" url CRLF
```


The following is an example of this property:

URL:http://abc.com/pub/calendars/jsmith/mytime.or3

The data type for this property is URL.

5.6.37 Unique Identifier

This property is identified by the property name UID. This property defines the persistent, globally unique identifier for the calendar component. The property must be specified in the event, to-do and journal calendar components.

This identifier is created by the calendar system that generates an iCalendar Object. The identifier is represented as a text value. This is the method for correlating scheduling messages with the referenced event, to-do, or journal.

The property is defined by the following notation:

uid = "UID" [";" paramlist] ":" text CRLF

Dawson/Stenerson 59 Expires January 1998

Internet Draft C&S Core Object Specification July 29, 1997

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 requests with later replies, modifications or deletion requests. Calendaring and scheduling applications must generate this property in event, to-do and journal calendar components to assure interoperability with other group scheduling applications.

The data type for this property is TEXT.

5.6.38 Non-standard Properties

The MIME Calendaring and Scheduling Content Type 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 prefix text of "X-" (the two character sequence: LATIN CAPITAL LETTER X character followed by the HYPHEN-MINUS character). It is recommended that

vendors concatenate onto this sentinel another short prefix text to identify the vendor. This will facilitate readability of the extensions and minimize possible collision of names between different vendors. User agents that support this content type are expected to be able to parse the extension properties and property parameters but may ignore them.

The property is defined by the following notation:

```
extension = "X-" [vendorid] word [";" paramlist] ":"
           value
```

```
vendorid  = 1*char "-" ;Vendor identification prefix text
```

The following might be the ABC vendor s extension for an audio-clip form of subject property:

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

At present, there is no registration authority for names of extension properties and property parameters. The data type for this property is TEXT. Optionally, the data type may be any of the other valid data types.

6. Recommended Practices

These recommended practices should be followed in order to assure consistent handling of the following cases for an iCalendar object.

1. A calendar entry with a DTSTART but no DTEND - The event does not take up any time. It is intended to represent an event that is associated with a given calendar date and time of day, such as an anniversary. Since the event does not take up any time, it must

Dawson/Stenerson

60

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

not be used to record busy time no matter what the value for the TRANSP property.

2. A combination of RRULE and RDATE that produces more than one instance for a given date/time - Only one recurrence can occur on a given date/time interval. Just one instance for the date/time is recorded.
3. A particular calendar profile that specifies ATTENDEE properties with the MEMBER property parameter, for which the recipient has multiple memberships - Recipient should reply to only the first

MEMBER value that it can match.

7. Registration of Content Type Elements

This section provide the process for registration of MIME Calendaring and Scheduling Content Type profiles and new or modified properties.

7.1 Registration of New and ModifiedProfiles

New MIME Calendaring and Scheduling Content Type profile types are registered by the publication of an IETF Request for Comment (RFC). Changes to a profile type are registered by the publication of a revision of the RFC defining the profile type.

7.2 Registration of New Properties

This section defines procedures by which new properties or enumerated property values for the MIME Calendaring and Scheduling Content Type can be registered with the IANA. Note that non-IANA properties may be used by bilateral agreement, provided the associated properties names follow the "X-" convention.

The procedures defined here are designed to allow public comment and review of new properties, while posing only a small impediment to the definition of new properties.

Registration of a new property is accomplished by the following steps.

7.2.1 Define the property

A property is defined by completing the following template.

To: ietf-calendar@imc.org

Subject: Registration of text/calendar MIME property XXX

Property name:

Property purpose:

Property data type(s):

Property encoding:

Property special notes (optional):

Intended usage: (one of COMMON, LIMITED USE or OBSOLETE)

The meaning of each field in the template is as follows.

Property name: The name of the property, as it will appear in the body of an text/calendar MIME Content-Type "property: value" line to the left of the colon ":".

Property purpose: The purpose of the property (e.g., to indicate a delegate for the event or to-do, etc.). Give a short but clear description.

Property data type(s): Any of the valid data types for the property value needs to be specified. The default data type also needs to be specified. If a new data type is specified, it needs to be declared in this section.

Property encoding: The encodings permitted for the property value. This description must be precise and must not violate the general encoding rules defined in this document.

Property special notes: Any special notes about the property, how it is to be used, etc.

7.2.2 Post the Property definition

The property description must be posted to the new property discussion list, ietf-calendar@imc.org.

7.2.3 Allow a comment period

Discussion on the new property must be allowed to take place on the list for a minimum of two weeks. Consensus must be reached on the property before proceeding to the next step.

7.2.4 Submit the property for approval

Once the two-week comment period has elapsed, and the proposer is convinced consensus has been reached on the property, the registration application should be submitted to the Profile Reviewer for approval. The Profile Reviewer is appointed to the Application Area Directors and may either accept or reject the property registration. An accepted registration should be passed on by the Profile Reviewer to the IANA for inclusion in the official IANA profile registry. The registration may be rejected for any of the

following reasons. 1) Insufficient comment period; 2) Consensus not reached; 3) Technical deficiencies raised on the list or elsewhere have not been addressed. The Profile Reviewer's decision to

reject a property may be appealed by the proposer to the IESG, or the objections raised can be addressed by the proposer and the property resubmitted.

Dawson/Stenerson

62

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

7.3 Property Change Control

Existing properties may be changed using the same process by which they were registered.

1. Define the change
2. Post the change
3. Allow a comment period
4. Submit the property for approval

Note that the original author or any other interested party may propose a change to an existing property, but that such changes should only be proposed when there are serious omissions or errors in the published specification. The Profile Reviewer may object to a change if it is not backwards compatible, but is not required to do so.

Property definitions can never be deleted from the IANA registry, but properties which are no longer believed to be useful can be declared OBSOLETE by a change to their "intended use" field.

8. File extension

The file extension of "vcs" is to be used to designate a file containing calendaring and scheduling information consistent with this MIME content type.

9. Macintosh File Type Code

The file type code of "vcal" is to be used in Apple MacIntosh operating system environments to designate a file containing calendaring and scheduling information consistent with this MIME

media type.

10. References

The following document are referred to within this document.

[ICMS] "Internet Calendaring Model Specification", Internet-Draft, July 1997, <ftp://ftp.ietf.org/internet-drafts/draft-ietf-calsch-mod-00.txt>.

[ISO 8601] ISO 8601, "Data elements and interchange formats-

Information interchange-

-Representation of dates and times", International Organization for Standardization, June, 1988. This standard is also addressed by the Internet Draft document <ftp://ds.internic.net/internet-drafts/draft-newman-datetime-00.txt>.

[ISO 9070] ISO/IEC 9070, "Information Technology-

-SGML Support

Facilities-

-Registration Procedures for Public Text Owner Identifiers", Second Edition, International Organization for Standardization, April, 1991.

Dawson/Stenerson

63

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

[ITIP-1] "iCalendar Transport-Independent Interoperability Protocol (iTIP) - Part 1: Scheduling Events and Busytime", Internet-Draft, July 1997, <http://www.imc.org/draft-ietf-calsch-itip-part1-00.txt>.

[ITIP-2] "iCalendar Transport-Independent Interoperability Protocol (iTIP) - Part 2: Scheduling To-dos", Internet-Draft, July 1997, <http://www.imc.org/draft-ietf-calsch-itip-part2-00.txt>.

[ITIP-3] "iCalendar Transport-Independent Interoperability Protocol (iTIP) - Part 3: Scheduling Journal Entries", Internet-Draft, July 1997, <http://www.imc.org/draft-ietf-calsch-itip-part3-00.txt>.

[MIME DIR] Howes, T., Smith, M., "A MIME Content-Type for Directory Information", Internet-draft-ietf-asid-mime-direct-06.txt, July, 1997.

[RFC 822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, [RFC 822](#), August 1982.

[RFC 1738] Berners-Lee, T., Masinter, L., McCahill, M., "Uniform Resource Locators (URL)", [RFC 1738](#), December 1994.

[RFC 1766] Alvestrand, H., "Tags for the Identification of Languages", March 1995.

[RFC 1872] Levinson, E., "The MIME Multipart/Related Content-type", [RFC 1872](#), December 1995.

[RFC 2045] Freed, N., Borenstein, N., " Multipurpose Internet Mail Extensions (MIME) - Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.

[RFC 2046] Freed, N., Borenstein, N., " Multipurpose Internet Mail Extensions (MIME) - Part Two: Media Types", [RFC 2046](#), November 1996.

[RFC 2047] Moore, K., "Multipurpose Internet Mail Extensions (MIME) - Part Three: Message Header Extensions for Non-ASCII Text", [RFC 2047](#), November 1996.

[RFC 2048] Freed, N., J. Klensin, J. Postel, "Multipurpose Internet Mail Extensions (MIME) - Part Four: Registration Procedures", [RFC 2048](#), January 1997.

[UTF-8] "UTF-8, a transformation format of Unicode and ISO 10646", Internet-Draft, July, 1996, [ftp://ftp.ietf.org/internet-drafts/draft-yergeau-utf8-01.txt](http://ftp.ietf.org/internet-drafts/draft-yergeau-utf8-01.txt).

[VCARD] Internet Mail Consortium, "vCard - The Electronic Business Card Version 2.1", <http://www.versit.com/pdi/vcard-21.txt>, September 18, 1996.

[VICAL] Internet Mail Consortium, "vCalendar - The Electronic Calendaring and Scheduling Exchange Format", <http://www.imc.org/pdi/vcal-10.txt>, September 18, 1996.

Dawson/Stenerson

64

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

[XAPIA] "XAPIA CSA, Calendaring and Scheduling Application Programming Interface (CSA) Version 1.0", X.400 API Association, November 15, 1994.

11. Acknowledgments

A hearty thanks to the IETF Calendaring and Scheduling Working Group and also the following individuals who have participated in the drafting, review and discussion of this memo:

Roland Alden, Harald T. Alvestrand, Eric Berman, Denis Bigorgne, John Binici, Bill Bliss, Steve Carter, Andre Courtemanche, Dave Crocker, Alec Dun, John Evans, Ross Finlayson, Randell Flink, Ned Freed, Patrik Falstrom, Chuck Grandgent, Mark Handley, Steve Hanna, Paul B. Hill, Paul Hoffman, Ross Hopson, Mark Horton, Bruce Kahn, C. Harald Koch, Don Lavange, Theodore Lorek, Steve Mansour, Skip Montanaro, Keith Moore, Cecil Murray, Chris Newman, John Noerenberg, Ralph Patterson, Pete Resnick, Keith Rhodes, Robert Ripberger, John Rose, Andras Salamar, Ted Schuh, Vinod Seraphin, Derrick Shadel, Ken Shan, Andrew Shuman, Steve Silverberg, William P. Spencer, Mark Towfiq, Robert Visnov, James L. Weiner, Mike Weston, William Wyatt.

12. Author's Address

The following address information is provided in a MIME-VCARD, Electronic Business Card, format.

The authors of this draft are:

```
BEGIN:VCARD
FN:Frank Dawson
ORG:Lotus Development Corporation
ADR;WORK;POSTAL;PARCEL;;;6544 Battleford Drive;
  Raleigh;NC;27613-3502;USA
TEL;WORK;MSG:+1-919-676-9515
TEL;WORK;FAX:+1-919-676-9564
EMAIL;INTERNET:fdawson@earthlink.net
URL:http://home.earthlink.net/~fdawson
END:VCARD
```

```
BEGIN:VCARD
FN:Derik Stenerson
ORG:Microsoft Corporation
ADR;WORK;POSTAL;PARCEL;;;One Microsoft Way;
  Redmond;WA;98052-6399;USA
TEL;WORK;MSG:+1-206-936-5522
TEL;WORK;FAX:+1-206-936-7329
EMAIL;INTERNET:deriks@Exchange.Microsoft.com
END:VCARD
```

The iCalendar object is a result of the work of the Internet Engineering Task Force Calendaring and Scheduling Working Group. The chairman of that working group is:

Internet Draft

C&S Core Object Specification

July 29, 1997

```
BEGIN:VCARD
FN:Anik Ganguly
ORG:OnTime, Inc.
ADR;WORK;POSTAL;PARCEL:10 Floor;;21700 Northwestern Highway;
  Southfield;MI;48075;USA
TEL;WORK;MSG:+1-810-559-5955
TEL;WORK;FAX:+1-810-559-5034
EMAIL;INTERNET:anik@ontime.com
END:VCARD
```

13. iCalendar Object Examples

The following examples are provided as an informational source of illustrative iCalendar objects consistent with this content type.

The following iCalendar object is specified as the content of a MIME message. The example demonstrates a possible meeting request between the originator and recipient of the message.

```
TO:jsmith@host1.com
FROM:jdoe@host1.com
MIME-VERSION:2.0
MESSAGE-ID:<19960704 08:30:00 EDT xyz@host1.com>
CONTENT-TYPE:text/calendar;PROFILE=request-event

BEGIN:VCALENDAR
PROFILE:event-request
PRODID:-//xyz Corp//NONSGML PDA Calendar Verson 1.0//EN
VERSION:2.0
BEGIN:VEVENT
DTSTART:19960918T143000Z
DTEND:19960920T220000Z
CATEGORIES:CONFERENCE, PROJECT
SUMMARY:Networld+Interop Conference
DESCRIPTION;ENCODING=Q:Networld+Interop_Conference_
  and_Exhibit=0D=0A
  Atlanta_World_Congress_Center=0D=0A
  Atlanta,_Georgia
END:VEVENT
END:VCALENDAR
```

The following example message issues a meeting request that does not require any reply. The message is sent as a singular "text/calendar" content type, body part.

```
From: jsmith@host1.com
```

To: ietf-calendar@imc.org
Subject: First IETF-Calendar Working Group Meeting
MIME-Version: 2.0
Message-ID: <id1@host1.com>
Content-Type: text/calendar;Profile=event,request

BEGIN:VCALENDAR
PROFILE:event-request
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0

Dawson/Stenerson

66

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

BEGIN:VEVENT
ATTENDEE;EXPECT=REQUEST:ietf-calendar@imc.org
DESCRIPTION:First IETF-Calendar and Scheduling Working Group
Meeting
CATEGORIES:MEETING
CLASS:PUBLIC
CREATED:19961022T083000
SUMMARY:IETF Calendar Working Group Meeting
DTSTART:19961210T210000Z
DTEND:19961210T220000Z
LOCATION:San Jose, CA - Fairmont Hotel
UID:guid-1.host1.com
END:VEVENT
END:VCALENDAR

The following is an example of a MIME message with a single body part consisting of a text/calendar content type. The message specifies a meeting request between the originator and recipient of the message.

TO:jsmith@host1.com
FROM:jdoe@host1.com
MIME-VERSION:1.0
MESSAGE-ID:<19970322 08:30:00 EDT xyz@host1.com>
CONTENT-TYPE:text/calendar;PROFILE=event-request

BEGIN:VCALENDAR
PROFILE:event-request
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT
SEQUENCE:0
UID:19970324-080045-4000F192713-0052
ATTENDEE;EXPECT=REQUEST:jsmith@host1.com
DTSTART:19970324T123000Z

DTEND:19970324T210000Z
CATEGORIES:CONFERENCE,PROJECT
CLASS:PUBLIC
SUMMARY:Calendaring Interop Conference
DESCRIPTION;ENCODING=Q:Calendaring_Interop_
Conference_and_Exhibit=0D=0A
Atlanta,_Georgia
LOCATION:Atlanta World Congress Center
ATTACH;VALUE=URL:file://xyzCorp.com/conf/bkgrnd.ps
END:VEVENT
END:VCALENDAR

Example of a reply to the above request, accepting the meeting.

TO:jdoe@host1.com
FROM:jsmith@host1.com
MIME-VERSION:1.0
MESSAGE-ID:<19970322 08:30:00 EDT xyz@host1.com>
CONTENT-TYPE:text/calendar;PROFILE=event-reply

BEGIN:VCALENDAR
PROFILE:event-reply

Dawson/Stenerson

67

Expires January 1998

Internet Draft

C&S Core Object Specification

July 29, 1997

VERSION:2.0
PROID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT
SEQUENCE:0
RESPONSE-SEQUENCE:0
UID:19970324-080045-4000F192713-0052
ATTENDEE;STATUS=CONFIRMED;EXPECT=REQUEST:jsmith@host1.com
END:VEVENT
END:VCALENDAR

An example of a meeting cancelation:

TO:jsmith@host1.com
FROM:jdoe@host1.com
MIME-VERSION:1.0
MESSAGE-ID:<19970322 08:30:00 EDT xyz@host1.com>
CONTENT-TYPE:text/calendar;PROFILE=event-cancel

BEGIN:VCALENDAR
PROFILE:event-cancel
VERSION:2.0
PROID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT

UID:19970324-080045-4000F192713-0052
END:VEVENT
END:VCALENDAR

Dawson/Stenerson

68

Expires January 1998