

Network Working Group
Internet Draft
<ietf-calsch-ical-06.txt>
Expires September 1998

Frank Dawson, Lotus
Derik Stenerson, Microsoft
March 13, 1998

Internet Calendaring and Scheduling Core Object Specification (iCalendar)

Status of this Memo

This memo 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 memo is unlimited.

Copyright (C) The Internet Society 1997. All Rights Reserved.

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 memo has been defined to provide the 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, to-do and journal entry 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, using HTTP or some other Internet transport. In

Dawson/Stenerson

1

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

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.

This memo 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 memo is to be known as the iCalendar specification.

Readers may also wish to refer to the calendaring and scheduling model defined in [] for a description of this Internet application.

This memo defines the format for specifying iCalendar object methods. An iCalendar object method is a set of usage constraints for the iCalendar object. For example, these methods 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. The iCalendar Transport-independent Interoperability Protocol (iTIP) defined in [[ITIP](#)] is one such scheduling protocol.

Table of Contents

| | | |
|-----------------------|-------------------------------------------------------------|--------------------|
| 1 | Introduction..... | 8 |
| 2 | Basic Grammar and Conventions..... | 8 |
| 2.1 | Formatting Conventions | 9 |
| 2.2 | Related Memos | 10 |
| 3 | TEXT/CALENDAR Registration Information..... | 10 |
| 4 | iCalendar Object Specification..... | 13 |
| 4.1 | Content Lines | 13 |
| 4.1.1 | List and Field Separators | 16 |
| 4.1.2 | Multiple Values | 16 |
| 4.1.3 | Binary Content | 17 |
| 4.1.4 | Character Set | 17 |
| 4.2 | Property Parameters | 17 |
| 4.2.1 | Alternate Text Representation | 18 |
| 4.2.2 | Inline Encoding | 19 |
| 4.2.3 | Language | 19 |
| 4.2.4 | Time Zone Identifier | 20 |
| 4.2.5 | Value Data Types | 21 |
| 4.3 | Property Value Data Types | 21 |
| 4.3.1 | Binary | 22 |
| 4.3.2 | Boolean | 22 |
| 4.3.3 | Calendar User Address | 23 |
| 4.3.4 | Date | 23 |
| 4.3.5 | Date-Time | 24 |
| 4.3.6 | Duration | 25 |
| 4.3.7 | Float | 26 |
| 4.3.8 | Integer | 27 |
| 4.3.9 | Period of Time | 27 |

| | | |
|-------------------------|----------------------------------------|--------------------|
| 4.3.10 | Recurrence Rule | 28 |
| 4.3.11 | Text | 33 |
| 4.3.12 | Time | 34 |
| 4.3.13 | URI | 36 |
| 4.3.14 | UTC Offset | 36 |
| 4.4 | iCalendar Object | 37 |
| 4.5 | Property | 38 |
| 4.6 | Calendar Components | 38 |
| 4.6.1 | Event Component | 38 |
| 4.6.2 | To-do Component | 40 |
| 4.6.3 | Journal Component | 41 |
| 4.6.4 | Free/Busy Component | 43 |
| 4.6.5 | Timezone Component | 45 |
| 4.6.6 | Alarm Component | 50 |
| 4.7 | Calendar Properties | 53 |
| 4.7.1 | Calendar Scale | 54 |
| 4.7.2 | Method | 54 |
| 4.7.3 | Product Identifier | 55 |
| 4.7.4 | Version | 56 |
| 4.8 | Component Properties | 56 |
| 4.8.1 | Descriptive Component Properties | 56 |
| 4.8.1.1 | Attachment | 56 |
| 4.8.1.2 | Categories | 57 |

Dawson/Stenerson

3

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

| | | |
|--------------------------|------------------------------------------|--------------------|
| 4.8.1.3 | Classification | 58 |
| 4.8.1.4 | Comment | 59 |
| 4.8.1.5 | Description | 59 |
| 4.8.1.6 | Geographic Position | 60 |
| 4.8.1.7 | Location | 61 |
| 4.8.1.8 | Percent Complete | 61 |
| 4.8.1.9 | Priority | 62 |
| 4.8.1.10 | Resources | 63 |
| 4.8.1.11 | Status | 64 |
| 4.8.1.12 | Summary | 65 |
| 4.8.2 | Date and Time Component Properties | 66 |
| 4.8.2.1 | Date/Time Completed | 66 |
| 4.8.2.2 | Date/Time End | 66 |
| 4.8.2.3 | Date/Time Due | 67 |
| 4.8.2.4 | Date/Time Start | 68 |
| 4.8.2.5 | Duration | 69 |
| 4.8.2.6 | Free/Busy Time | 69 |
| 4.8.2.7 | Time Transparency | 71 |
| 4.8.3 | Time Zone Component Properties | 72 |
| 4.8.3.1 | Time Zone Identifier | 72 |

| | | |
|---------|-----------------------------------------------------------------|-----|
| 4.8.3.2 | Time Zone Name | 72 |
| 4.8.3.3 | Time Zone Offset From | 73 |
| 4.8.3.4 | Time Zone Offset To | 74 |
| 4.8.3.5 | Time Zone URL | 74 |
| 4.8.4 | Relationship Component Properties | 75 |
| 4.8.4.1 | Attendee | 75 |
| 4.8.4.2 | Contact | 78 |
| 4.8.4.3 | Organizer | 79 |
| 4.8.4.4 | Recurrence ID | 80 |
| 4.8.4.5 | Related To | 81 |
| 4.8.4.6 | Uniform Resource Locator | 82 |
| 4.8.4.7 | Unique Identifier | 83 |
| 4.8.5 | Recurrence Component Properties | 84 |
| 4.8.5.1 | Exception Date/Times | 84 |
| 4.8.5.2 | Exception Rule | 85 |
| 4.8.5.3 | Recurrence Date/Times | 86 |
| 4.8.5.4 | Recurrence Rule | 88 |
| 4.8.6 | Alarm Component Properties | 95 |
| 4.8.6.1 | Alarm Type | 96 |
| 4.8.6.2 | Repeat Count | 96 |
| 4.8.6.3 | Trigger | 97 |
| 4.8.7 | Change Management Component Properties | 98 |
| 4.8.7.1 | Date/Time Created | 98 |
| 4.8.7.2 | Date/Time Stamp | 99 |
| 4.8.7.3 | Last Modified | 99 |
| 4.8.7.4 | Sequence Number | 100 |
| 4.8.8 | Miscellaneous Component Properties | 101 |
| 4.8.8.1 | Non-standard Properties | 101 |
| 4.8.8.2 | Request Status | 102 |
| 5 | iCalendar Object Examples..... | 103 |
| 6 | Recommended Practices..... | 107 |
| 7 | Registration of Content Type Elements..... | 107 |
| 7.1 | Registration of New and Modified iCalendar Object Methods | 107 |
| 7.2 | Registration of New Properties | 108 |

Dawson/Stenerson

4

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

| | | |
|-------|----------------------------------------|-----|
| 7.2.1 | Define the property | 108 |
| 7.2.2 | Post the Property definition | 109 |
| 7.2.3 | Allow a comment period | 109 |
| 7.2.4 | Submit the property for approval | 109 |
| 7.3 | Property Change Control | 109 |
| 8 | References..... | 109 |
| 9 | Acknowledgments..... | 111 |
| 10 | Author's Address..... | 111 |
| 11 | Full Copyright Statement..... | 112 |

Dawson/Stenerson

5

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

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 knowledgeware applications. This memo is intended to progress the level of interoperability possible between dissimilar calendaring and scheduling applications. This memo 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 (PIM) or a Group Scheduling product.

The calendaring and scheduling model, defined in the [[ICMS](#)], is a useful reference to terms and the general framework of this Internet application.

The iCalendar 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 memo also provides for the definition of iCalendar object methods 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 iCalendar object methods can be used to define other calendaring and scheduling operations such as requesting for and replying with free/busy time data. Such a scheduling protocol is defined in the iCalendar Transport-independent Interoperability Protocol (iTIP) defined in [[ITIP](#)].

The memo also includes a formal grammar for the content type based on the Internet ABNF defined in [[RFC 2234](#)]. This ABNF is required for the implementation of parsers and to serve as the definitive reference when ambiguities or questions arise in interpreting the descriptive prose definition of the memo.

2 Basic Grammar and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this

document are to be interoperated as described in [[RFC 2119](#)].

Dawson/Stenerson

6

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

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

The notation used in this memo is the ABNF notation of [[RFC 2234](#)]. Readers intending on implementing this format defined in this memo should be familiar with this notation in order to properly interpret the specifications of this memo.

All numeric and hexadecimal values used in this memo 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 memo. 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.

[2.1](#) Formatting Conventions

The mechanisms defined in this memo are defined in propose. Many of the terms used to describe these have common usage that is different than the standards usage of this memo. In order to reference within this memo elements of the calendaring and scheduling model [[ICMS](#)], core object (this memo) or interoperability protocol [[ITIP](#)] some formatting conventions have been used. Calendaring and scheduling roles defined by [[ICMS](#)] are referred to in quoted-strings of text with the first character of each word in upper case. For example, "Organizer" refers to a role of a "Calendar User" within the scheduling protocol defined by [[ITIP](#)] Calendar components defined by this memo are referred to with capitalized, quoted-strings of text. All calendar components start with the letter "V". For example, "VEVENT" refers to the event calendar component, "VTODO" refers to the to-do calendar component and "VJOURNAL" refers to the daily journal calendar component. Scheduling methods defined by [[ITIP](#)] are

referred to with capitalized, quoted-strings of text. For example, "REQUEST" refers to the method for requesting a scheduling calendar component be created or modified, "REPLY" refers to the method a recipient of a request uses to update their status with the "Organizer" of the calendar component.

The properties defined by this memo are referred to with capitalized, quoted-strings of text, followed by the word "property". For example, "ATTENDEE" property refers to the iCalendar property used to convey the calendar address of a calendar user. Property parameters defined by this memo are referred to with lower case, quoted-strings of text,

Dawson/Stenerson

7

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

followed by the word "parameter". For example, "value" parameter refers to the iCalendar property parameter used to override the default data type for a property value. Enumerated values defined by this memo are referred to with capitalized text, either alone or followed by the word "value". For example, the "MINUTELY" value can be used with the "FREQ" component of the "RECUR" data type to specify repeating components based on an interval of one minute or more.

2.2 Related Memos

Implementers will need to be familiar with several other memos that, along with this memo, form a framework for Internet calendaring and scheduling standards. This memo, [ICAL], specifies a core specification of objects, data types, properties and property parameters.

[ICMS] - specifies a common terminology and abstract model;

[ITIP] - specifies an interoperability protocol for scheduling between different implementations;

[IMIP] specifies an Internet email binding for [[ITIP](#)];

[IRIP] - specifies an Internet real time protocol binding for [[ITIP](#)].

This memo does not attempt to repeat the specification of concepts or definitions from these other memos. Where possible, references are made to the memo that provides for the specification of these concepts or definitions.

3 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 memo is in no way limited solely as a MIME content type.

The following text is intended to register this memo 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

Required parameters: none

Optional parameters: charset, method and component

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.

Dawson/Stenerson

8

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

The "method" parameter is used to convey the iCalendar object method or transaction semantics for the calendaring and scheduling information. It also is an identifier for the restricted set of properties and values that the iCalendar object consists of. The parameter is to be used as a guide for 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 method definition specifically calls for this behavior. Unless specifically forbidden by a particular method definition, a text/calendar content type MAY contain any set of properties permitted by the Calendaring and Scheduling Core Object Specification. The "method" parameter MUST be the same value as that specified in the "METHOD" component property in the iCalendar object. If one is present, the other must also be present.

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

```
method = 1*(ALPHA / DIGIT / "-")
; IANA registered iCalendar object method
```

The "component" parameter conveys the type of iCalendar calendar component within the body part. If the iCalendar object contains more than one calendar component, then different components are

each specified once in a comma-separated list of values.

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

```
component      = ("VEVENT" / "VTODO" / "VJOURNAL" / "VFREEBUSY"  
                  / "VTIMEZONE" / x-name / iana-token) ["," [WSP]  
                  component]
```

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 a text valued property in the content entity MAY also have content encoding of special characters using a BACKSLASH character (ASCII decimal 92) escapement technique. This means that content values MAY end up encoded twice.

Security considerations: SPOOFING - - In this memo, the "Organizer" is the only person authorized to make changes to an existing "VEVENT", "VTODO", "VJOURNAL" calendar component and redistribute the updates to the "Attendees". An iCalendar object that maliciously changes or cancels an existing "VEVENT", "VTODO" or "VJOURNAL" or "VFREEBUSY" calendar component MAY be constructed by someone other than the "Organizer" and sent to the "Attendees". In addition in this memo, other than the "Organizer", an "Attendee" of a "VEVENT", "VTODO", "VJOURNAL" calendar component is the only other person authorized to update any parameter associated with their "ATTENDEE" property and send it to the "Organizer". An

Dawson/Stenerson

9

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

iCalendar object that maliciously changes the "ATTENDEE" parameters MAY be constructed by someone other than the real "Attendee" and sent to the "Organizer".

PROCEDURAL ALARMS - - An iCalendar object can be created that contains a "VEVENT" and "VTODO" calendar component with an "VALARM" calendar components. The "VALARM" calendar component MAY be of type PROCEDURE and MAY have an attachment containing some sort of executable program. Implementations that incorporate these types of alarms are subject to any virus or malicious attack that MAY occur as a result of executing the attachment.

ATTACHMENTS - - An iCalendar object MAY include references to

Uniform Resource Locators that MAY be programmed resources.

Implementers and users of this memo 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 memo.

Interoperability considerations: This MIME content type is intended to define a common format for conveying calendaring and scheduling information between different systems. It is heavily based on the earlier [[VCAL](#)] industry specification.

Applications which use this media type: This content-type is designed for widespread use by Internet calendaring and scheduling applications. In addition, applications in the workflow and document management area may find this content-type applicable. The [[ITIP](#)], [[IMIP](#)] and [[IRIP](#)] Internet protocols directly use this content-type also. Future work on an Internet calendar access protocol will utilize this content-type too.

Additional information: This memo defines this content-type.

Magic numbers: None.

File extensions: The file extension of "ics" is to be used to designate a file containing an arbitrary set of calendaring and scheduling information consistent with this MIME content type.

The file extension of "ifb" is to be used to designate a file containing free or busy time information consistent with this MIME content type.

Macintosh file type codes: The file type code of "ical" 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.

The file type code of "ifb" is to be used in Apple MacIntosh operating system environments to designate a file containing free or busy time information consistent with this MIME media type.

Dawson/Stenerson

10

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Person & email address to contact for further information:

Frank Dawson

6544 Battleford Drive
Raleigh, NC 27613-3502
919-676-9515 (Telephone)
919-676-9564 (Data/Facsimile)
Frank_Dawson@Lotus.com (Internet Mail)

Derik Stenerson
One Microsoft Way
Redmond, WA 98052-6399
425-936-5522 (Telephone)
425-936-7329 (Facsimile)
deriks@microsoft.com (Internet Mail)

Intended usage: COMMON

Author/change controller:

Frank Dawson
6544 Battleford Drive
Raleigh, NC 27613-3502
919-676-9515 (Telephone)
919-676-9564 (Data/Facsimile)
Frank_Dawson@Lotus.com (Internet Mail)

Derik Stenerson
One Microsoft Way
Redmond, WA 98052-6399
425-936-5522 (Telephone)
425-936-7329 (Facsimile)
deriks@microsoft.com (Internet Mail)

4 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 memo has direct applicability for use as a calendaring and scheduling exchange format in file-, memory- or network-based transport mechanisms.

4.1 Content Lines

The iCalendar object is organized into individual lines of text, called content lines. Content lines are delimited by a line break, which is a CRLF sequence (ASCII decimal 13, followed by ASCII decimal 10).

Internet Draft

C&S Core Object Specification

March 6, 1998

Lines of text SHOULD NOT be longer than 75 characters, excluding the line break. Long content lines SHOULD 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 linear white space character (i.e., SPACE, ASCII decimal 32 or HTAB, ASCII decimal 9). Any sequence of CRLF followed immediately by a single linear white space 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 lo
ng 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 linear white space character that immediately follows.

When parsing a content line, folded lines must first be unfolded according to the unfolding procedure described above. When generating a content line, lines longer than 75 characters SHOULD be folded according to the folding procedure described above.

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 content lines.

The following notation defines the defines the lines of content in an iCalendar object:

```
contentline      = name *(";" [WSP] param ) ":" value CRLF
; This ABNF is just a general definition for an initial parsing
; of the content line into its property name, parameter list,
; and value string

; When parsing a content line, folded lines must first
; be unfolded according to the unfolding procedure
; described above. When generating a content line, lines
```

; longer than 75 characters SHOULD be folded according to
; the folding procedure described above.

name = x-name / iana-token

iana-token = 1*(ALPHA / DIGIT / "-")
; iCalendar identifier registered with IANA

Dawson/Stenerson

12

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

x-name = "X-" 1*(ALPHA / DIGIT / "-")
; Reserved for experimental use. Not intended for use in
; released products.

param = param-name "=" param-value
*(", " param-value)
; Each property defines the specific ABNF for the parameters
; allowed on the property. Refer to specific properties for
; precise parameter ABNF.

param-name = iana-token

param-value = paramtext / quoted-string

paramtext = *SAFE-CHAR

value = *VALUE-CHAR

quoted-string = DQUOTE *qtext DQUOTE

qtext = QSAFE-CHAR / QUOTED-CHAR

NON-ASCII = %x80-FF
; Use restricted by charset parameter
; on outer MIME object (UTF-8 preferred)

QSAFE-CHAR = WSP / %x21 / %x23-5B / %x5D-7E / NON-ASCII
; Any character except CTLs, DQUOTE, or "\"

QUOTED-CHAR = "\" ("\" / "\", " / DQUOTE)
; \\ encodes \, \" encodes "

SAFE-CHAR = WSP / %x21 / %x23-2B / %x2D-39 / %x3C-7E
/ NON-ASCII
; Any character except CTLs, DQUOTE, ";", ":", ",", "

VALUE-CHAR = WSP / VCHAR / NON-ASCII
; Any textual character

VCHAR = %x21-7E
; visible (printing) characters

CR = %x0D
; carriage return

LF = %x0A

CRLF = CR LF
; Internet standard newline

CTL = %x00-1F / %x7F
; Controls

ALPHA = %x41-5A / %x61-7A ; A-Z / a-z

Dawson/Stenerson

13

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

DIGIT = %x30-39
; 0-9

DQUOTE = %x22
; Quotation Mark

WSP = SPACE / HTAB

SPACE = %x20

HTAB = %x09

The property value component of a content line has a format that is property specific. Refer to the section describing each property for a definition of this format.

[4.1.1](#) List and Field Separators

List of values MAY be specified for property values or property parameter values. Each value in a list of values MUST be separated by a COMMA character (ASCII decimal 44).

Some property values are defined in terms of multiple components. These structured property values MUST have their components separated by a SEMICOLON character (ASCII decimal 59).

Lists of property parameters MAY be specified for a property. Each

property parameter in a list of property parameters MUST be separated by a SEMICOLON character (ASCII decimal 59).

Property parameters with values containing a COLON, a SEMICOLON or a COMMA character must be placed in quoted text string.

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=REQ-PARTICIPANT:MAILTO:jsmith@host.com
```

```
RDATE;VALUE=DATE:19970304,19970504,19970704,19970904
```

[4.1.2](#) Multiple Values

Some properties defined in the iCalendar object MAY have multiple values. 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 character (ASCII decimal 44).

Dawson/Stenerson 14 Expires September 1998

Internet Draft C&S Core Object Specification March 6, 1998

[4.1.3](#) Binary Content

Binary content information in an iCalendar object SHOULD be referenced using a URI within a property value. That is the binary content information SHOULD be placed in an external MIME entity that can be referenced by a URI from within the iCalendar object. In applications where this is not feasible, binary content information MAY be included within an iCalendar object, but only after first encoding it into text using the "B" encoding method defined in [RFC 2047]. Support for inline binary content SHOULD be restricted to those applications requirements that necessitate conveying the complete calendaring and scheduling information within a single iCalendar object. A property containing inline binary content information MUST specify the "ENCODING" property parameter. Binary content information placed external to the iCalendar object MUST be referenced by a uniform resource identifier (URI).

The following example specifies an "ATTACH" property that references an attachment external to the iCalendar object with a URI reference:

```
ATTACH:http://xyz.com/public/quarterly-report.doc
```

The following example specifies an "ATTACH" property with inline binary encoded content information:

```
ATTACH;ENCODING=b;VALUE=binary:MIICajCCAdOgAwIBAgICBEUwDQYJKoZI
hvcNAQEEBQAwdzELMAkGA1UEBhMCVVMxLDAqBgNVBAoTI05ldHNjYXB1IENv
bW11bm1jYXRpb25zIENvcnBvcnF0aW9uMRwwGgYDVQQLExNJbmZvcmlhdGlv
biBTeXN0 <...remainder of "B" encoded binary data...>
```

4.1.4 Character Set

There is not a property parameter to declare the character set used in a property value. The default character set for an iCalendar object is [[UTF-8](#)].

The "charset" Content-Type parameter MAY be used in MIME transports to specify any other IANA registered character set.

4.2 Property Parameters

A property MAY have additional attributes associated with it. These "property parameters" contain meta information about the property or the property value. Property parameters MAY be used to specify the location of an alternate text representation for a property value, the language of a text property value or the data type of the property value. In addition, individual properties MAY define property-specific parameters that apply only to that property.

Property parameter values that contain the COLON, SEMICOLON, COMMA or BACKSLASH character separators MUST be specified as quoted-string text values. For example:

Dawson/Stenerson

15

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
DESCRIPTION;ALTREP="http://www.wiz.org":The Fall'98 Wild Wizards
Conference - - Las Vegas, NV, USA
```

Property parameter values that are not in quoted strings are case insensitive.

The general property parameters defined by this memo are specified

the following notation:

```
parameter = altrepparm           ;Alternate text representation
           / encodingparm        ;Inline encoding
           / languageparm        ;National language for text
           / tzidparm            ;Reference to time zone object
           / valuetyeparm        ;Property value data type
           / iana-token
           ;Some other IANA registered iCalendar parameter, such as those
           ;defined by an individual property.
           / x-name
           ;A non-standard, experimental parameter
```

4.2.1 Alternate Text Representation

Parameter Name: ALTREP

Purpose: To specify an alternate text representation for the property value.

Conformance: Optional

Format Definition: The property parameter is defined by the following notation:

```
altrepparm = "altrep" "=" DQUOTE uri DQUOTE
```

Description: The parameter specifies a URI that points to an alternate representation for a textual property value. A property specifying this parameter MUST also include a value that reflects the default representation.

This property parameter MAY include multiple values, separated by the COMMA character (ASCII decimal 44). The property parameter MAY only be specified in the "COMMENT", "CONTACT", "DESCRIPTION", "LOCATION" and "SUMMARY" properties.

Example:

```
DESCRIPTION;ALTREP="CID:<part3.msg.970415T083000@host.com>":Project
XYZ Review Meeting will include the following agenda items: (a)
Market Overview, (b) Finances, (c) Project Management
```

The "ALTREP" property parameter value might point to a "text/html" content portion.

Content-Type:text/html

Content-Id:<part3.msg.970415T083000@host.com>

<p>Project XYZ Review Meeting will include the following
agenda items:Market
OverviewFinancesProject Management</p>

[4.2.2](#) **Inline Encoding**

Parameter Name: ENCODING

Purpose: To specify an alternate inline encoding for the property value.

Conformance: Required, if the inline encoding is not "8bit".

Format Definition: The property parameter is defined by the following notation:

```
encodingparm      = "encoding" "=" "8bit"  
                    ;"8bit" text encoding is defined in [RFC 2045]  
                    / "b"  
                    ;"b" binary encoding format is defined in [RFC 2047]  
                    / iana-token  
                    ;Some other IANA registered iCalendar encoding type  
                    / x-name  
                    ;A non-standard, experimental encoding type
```

Description: The "ENCODING" property parameter is an OPTIONAL property parameter. It identifies the inline encoding used in a property value. The default encoding type is "8bit", corresponding to a property value consisting of text. The "b" encoding type corresponds to a property value encoded using the "B" encoding defined in [RFC 2047].

If the value type parameter is ";VALUE=BINARY", then the inline encoding parameter MUST be specified with the value ";ENCODING=B".

[4.2.3](#) **Language**

Parameter Name: LANGUAGE

Purpose: To specify the national language for text values in a property or property parameter.

Conformance: Optional. There is no default national language for a text value.

Format Definition: The property parameter is defined by the following

notation:

languageparm = "language" "=" language

language = <Text identifying a language, as defined in [[RFC 1766](#)]>

Dawson/Stenerson

17

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Description: The parameter identifies the national language of the text in the property or property parameter value. 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.

Example:

SUMMARY;LANGUAGE=us-EN:Company Holiday Party

ATTENDEE;CN=Henri Weisz;LANGUAGE=ca-FR:hweisz@host.com

[4.2.4](#) Time Zone Identifier

Parameter Name: TZID

Purpose: To specify the identifier for the time zone definition for a time value.

Conformance: Optional

Format Definition: This property parameter is defined by the following notation:

tzidparm = "TZID" "=" [tzidprefix] paramtext CRLF

tzidprefix = "/"

Description: The parameter MAY only be used on the "DTSTART", "DTEND", "DUE", "EXDATE" and "RDATE" properties when either a DATE-TIME or TIME value type is specified. This property parameter specifies a text value which uniquely identifies the "VTIMEZONE" calendar component to be used when evaluating the time portion of the property. The value of the TZID property parameter will be equal to the value of the TZID property for the matching "TIMEZONE" object. The presence of the SOLIDUS character (ASCII decimal 47) as a prefix, indicates that this TZID represents a unique ID in a globally defined

time zone registry (when such registry is defined).

The following are examples of this property parameter:

```
DTSTART;TZID=America-New_York:19980119T020000
```

```
DTEND;TZID=America-New_York:19980119T030000
```

The TZID property parameter MUST NOT be applied to DATE-TIME nor TIME properties whose time values are specified in UTC.

The use of local time in a DATE-TIME or TIME value without the TZID property parameter is to be interpreted as a local time value, regardless of the existence of "VTIMEZONE" calendar components in the iCalendar object.

| | | |
|------------------|----|------------------------|
| Dawson/Stenerson | 18 | Expires September 1998 |
|------------------|----|------------------------|

| | | |
|----------------|-------------------------------|---------------|
| Internet Draft | C&S Core Object Specification | March 6, 1998 |
|----------------|-------------------------------|---------------|

For more information see the sections on the data types DATE-TIME and TIME.

[4.2.5](#) Value Data Types

Parameter Name: VALUE

Purpose: To explicitly specify the data type format for a property value.

Conformance: Optional if using the default value type in a property value. Otherwise, it is required.

Format Definition: The "VALUE" property parameter is defined by the following notation:

```
valuetypeparm = "value" "=" valuetype
```

```
valuetype    = "binary"  
              / "boolean"  
              / "cal-address"  
              / "date"  
              / "date-time"  
              / "duration"  
              / "float"  
              / "integer"  
              / "period"  
              / "recur"  
              / "text"
```

```

/ "time"
/ "uri"
/ "utc-offset"
/ x-name
    ;Some experimental iCalendar data type.
/ iana-token
    ;Some other IANA registered iCalendar data type.

```

Description: The parameter is used to identify the data type and format of the property value. The property values MUST be of a single value type. For example, a "RDATE" property cannot have a combination of DATE-TIME and TIME value types.

4.3 Property Value Data Types

The properties in an iCalendar object are strongly typed. The definition of each property restricts the value to be one of the value data types, or simply value types, defined in this section. The value type for a property will either be specified implicitly as the default value type or will be explicitly specified with the "VALUE" parameter. If the value type of a property is one of the alternate valid types, then it MUST be explicitly specified with the "VALUE" parameter.

| | | |
|------------------|-------------------------------|------------------------|
| Dawson/Stenerson | 19 | Expires September 1998 |
| Internet Draft | C&S Core Object Specification | March 6, 1998 |

4.3.1 Binary

Value Name: BINARY

Purpose: This value type is used to identify properties that contain a character encoding of inline binary data. For example, an inline attachment of an object code MAY BE included in an iCalendar object.

Formal Definition: The value type is defined by the following notation:

```

binary      = *(4b-char) [b-end]
; A "B" encoded character string, as defined by [RFC 2047].

b-end       = (2b-char "==") / (3b-char "=")

b-char = ALPHA / DIGIT / "+" / "/"

```

Description: Property values with this value type MUST also include

the inline encoding parameter sequence of ";ENCODING=B". That is, all inline binary data MUST first be character encoded using the "B" encoding method defined in [RFC 2047]. No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

Example: The following is an abridged example of a "B" encoded binary value data.

```
ATTACH;VALUE=BINARY:MIICajCCAdOgAwIBAgICBEUwDQYJKoZIhvcNAQEEBQA
wdzELMAkGA1UEBhMCVVMxLDAqBgNVBAoTI05ldHNjYXB1IENvbW11bm1jYXRpb
25zIENvcnBvcnF0aW9uMRwwGgYDVQQLExNjbmZvcn1hdGlvbiBTeXN0
<...remainder of "B" encoded binary data...>
```

4.3.2 Boolean

Value Name: BOOLEAN

Purpose: This value type is used to identify properties that contain either a "true" or "false" boolean value.

Formal Definition: The value type is defined by the following notation:

```
boolean    = "TRUE" / "FALSE"
```

Description: These values are case insensitive text. No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

Example: For example, any of the following are equivalent:

```
TRANSP;VALUE=BOOLEAN:TRUE
TRANSP;VALUE=BOOLEAN:true
TRANSP;VALUE=BOOLEAN:TrUe
```

Dawson/Stenerson

20

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

4.3.3 Calendar User Address

Value Name: CAL-ADDRESS

Purpose: This value type is used to identify properties that contain a calendar user address.

Formal Definition: The value type is as defined by the following notation:

cal-address = uri

Description: The value is a URI as defined by [[RFC 1738](#)] or any other IANA registered form for a URI. When used to address an Internet email transport address for a calendar user, the value MUST be a MAILTO URI, as defined by [[RFC 1738](#)]. No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

Example:

ATTENDEE;VALUE=CAL-ADDRESS:MAILTO:jane_doe@host.com

[4.3.4](#) Date

Value Name: DATE

Purpose: This value type is used to identify values that contain a calendar date.

Formal Definition: The value type is defined by the following notation:

```
date           = date-value *("," [WSP] date-value)

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

Description: If the property permits, multiple "date" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. The format for the value type is expressed as the [ISO 8601] complete representation, basic format for a calendar date. The textual 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.

No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

Example: The following represents July 14, 1997:

19970714

4.3.5 Date-Time

Value Name: DATE-TIME

Purpose: This value type is used to identify values that specify a precise calendar data and time of day.

Formal Definition: The value type is defined by the following notation:

date-time = dt-value *("," [WSP] dt-value)

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

Description: If the property permits, multiple "date-time" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

The "DATE-TIME" data type is used to identify values that contain a precise calendar date and time of day. The format is based on 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.

The "DATE-TIME" data type expresses time values in three forms:

The form of date and time with UTC offset MUST NOT be used. For example, the following is not valid for a date-time value:

DTSTART:19980119T230000-0800

FORM #1: DATE WITH LOCAL TIME

The date with local time form is simply a date-time value that does not contain the UTC designator nor does it reference a time zone. For example, the following represents January 18, 1998, at 11 PM:

DTSTART:19980118T230000

Date-time values of this type are said to be "floating" and are not bound to any time zone in particular. They are used to represent the same hour, minute, and second value regardless of which time zone is currently being observed. 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, no matter which timezone the person is in. 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 whatever timezone the ATTENDEE is in at any given moment. This means that two ATTENDEEs may participate in the same event at

Dawson/Stenerson

22

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

different UTC times; floating time should only be used where that is reasonable behavior.

In most cases, a fixed time is desired. To properly communicate a fixed time in a property value, either UTC time or local time with time zone reference MUST be specified.

The use of local time in a DATE-TIME value without the TZID property parameter is to be interpreted as a local time value, regardless of the existence of "VTIMEZONE" calendar components in the iCalendar object.

FORM #2: DATE WITH UTC TIME

The date with UTC time, or absolute time, is identified by a LATIN CAPITAL LETTER Z suffix character (ASCII decimal 90), the UTC designator, appended to the time value. For example, the following represents January 19, 1998, at 0700 UTC:

```
DTSTART:19980119T070000Z
```

The TZID property parameter MUST NOT be applied to DATE-TIME properties whose time values are specified in UTC.

FORM #3: DATE WITH LOCAL TIME AND TIME ZONE REFERENCE

The date and local time with reference to time zone information is identified by the use the TZID property parameter to reference the appropriate time zone definition. TZID is discussed in detail in the section on Time Zone. For example, the following represents 2 AM in New York on January 19, 1998:

```
DTSTART;TZID=America-New_York:19980119T020000
```

Example: The following represents July 14, 1997, at 1:30 PM in New York City in each of the three time formats, using the "DTSTART" property.

```
DTSTART:19970714T133000          ;Local time
```

```
DTSTART:19970714T153000Z        ;UTC time
```

```
DTSTART;TZID=America-NYC:19970714T133000 ;Local time and time
```

; zone reference

4.3.6 Duration

Value Name: DURATION

Purpose: This value type is used to identify properties that contain a duration of time.

Formal Definition: The value type is defined by the following notation:

duration = dur-value *(", " [WSP] dur-value)

Dawson/Stenerson

23

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

dur-value = (["+"] / "-") "P" (dur-date / dur-time / dur-week)

dur-date = (dur-day) [dur-time]

dur-time = "T" (dur-hour / dur-minute / dur-second)

dur-week = 1*DIGIT "W"

dur-hour = 1*DIGIT "H" [dur-minute]

dur-minute = 1*DIGIT "M" [dur-second]

dur-second = 1*DIGIT "S"

dur-day = 1*DIGIT "D"

Description: If the property permits, multiple "duration" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. 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.

No additional content value encoding (i.e., BACKSLASH character encoding) are defined for this value type.

Example: A duration of 15 days, 5 hours, 30 minutes and 20 seconds would be:

P15DT5H30M20S

A duration of 7 weeks would be:

P15DT5H30M20S

4.3.7 Float

Value Name: FLOAT

Purpose: This value type is used to identify properties that contain a real value number value.

Formal Definition: The value type is defined by the following notation:

float = float-value *("," [WSP] float-value)

float-value = (["+"] / "-") 1*DIGIT ["." 1*DIGIT]

Description: If the property permits, multiple "float" values MAY be specified using a COMMA character (ASCII decimal 44) separator character.

No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

Example:

Dawson/Stenerson

24

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

1000000.0000001

1.333

-3.14

4.3.8 Integer

Value Name:INTEGER

Purpose: This value type is used to identify properties that contain a signed integer value.

Formal Definition: The value type is defined by the following notation:

integer = integer-value *("," [WSP] integer-value)

integer-value = (["+"] / "-") *DIGIT

Description: If the property permits, multiple "integer" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. 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.

No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

Example:

```
1234567890
-1234567890
+1234567890
432109876
```

[4.3.9](#) Period of Time

Value Name: PERIOD

Purpose: This value type is used to identify values that contain a precise period of time.

Formal Definition: The data type is defined by the following notation:

```
period      = period-value *("," [WSP] period-value)

period-value = period-explicit / period-start

period-explicit = date-time "/" date-time
;ISO 8601 complete representation basic format for a period of time
;consisting of a start and end. The start MUST be before the end.
```

| | | |
|------------------|----|------------------------|
| Dawson/Stenerson | 25 | Expires September 1998 |
|------------------|----|------------------------|

| | | |
|----------------|-------------------------------|---------------|
| Internet Draft | C&S Core Object Specification | March 6, 1998 |
|----------------|-------------------------------|---------------|

```
period-start = date-time "/" duration
;ISO 8601 complete representation basic format for a period of time
;consisting of a start and duration of time.
```

Description: If the property permits, multiple "period" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. There are two forms of a period of time. A period of time MAY be identified by it's start and it's end. This format is expressed as the [ISO 8601] complete representation, basic format for "DATE-TIME" start of the period, followed by a SOLIDUS character (ASCII decimal 47), followed by the "DATE-TIME" of the end of the period. A period of time MAY also be defined by a start and a duration of time. The format is expressed as the [ISO 8601] complete

representation, basic format for the "DATE-TIME" start of the period, followed by a SOLIDUS character (ASCII decimal 47), followed by the [ISO 8601] basic format for "DURATION" of the period.

Example: The period starting at 18:00:00 UTC, on January 1, 1997 and ending at 07:00:00 UTC on January 2, 1997 would be:

```
19970101T180000Z/19970102T070000Z
```

The period start at 18:00:00 on January 1, 1997 and lasting 5 hours and 30 minutes would be:

```
19970101T180000Z/PT5H30M
```

No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

4.3.10 Recurrence Rule

Value Name: RECURR

Purpose: This value type is used to identify properties that contain a recurrence rule specification.

Formal Definition: The value type is defined by the following notation:

```
recur      = recur-value *("," [WSP] recur-value)

recur-value = "FREQ"=freq ";"
              [ ("UNTIL" "=" enddate ";") / ("COUNT" "=" 1*DIGIT ";")]
              ["INTERVAL" "=" 1*DIGIT ";"]
              ["BYSECOND" "=" byseclist ";"]
              ["BYMINUTE" "=" byminlist ";"]
              ["BYHOUR" "=" byhrlist ";"]
              ["BYDAY" "=" byweekdaylist ";"]
              ["BYMONTHDAY" "=" bymonthdaylist ";"]
              ["BYYEARDAY" "=" byyeardaylist ";"]
              ["BYWEEKNO" "=" byweekno ";"]
              ["BYMONTH" "=" bymonthlist ";"]
              ["BYSETPOS" "=" bysetposlist ";"]
```

Dawson/Stenerson

26

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
["WKST" "=" weekday ";"]
*(x-name "=" text) ";"
;Individual components MAY only be specified once.
```

;Rule components need not be specified in particular any order.

```
freq          = "SECONDLY" / "MINUTELY" / "HOURLY" / "DAILY"  
              / "WEEKLY" / "MONTHLY" / "YEARLY"  
  
enddate       = date  
enddate       =/ date-time           ;A UTC value  
  
byseclist     = seconds / ( seconds *(", " seconds) )  
  
seconds       = 1DIGIT / 2DIGIT      ;0 to 59  
  
byminlist     = minutes / ( minutes *(", " minutes) )  
  
minutes       = 1DIGIT / 2DIGIT      ;0 to 59  
  
byhrlist      = hour / ( hour *(", " hour) )  
  
hour          = 1DIGIT / 2DIGIT      ;0 to 23  
  
byweekdaylist = weekdaynum / ( weekdaynum *(", " weekdaynum) )  
  
weekdaynum    = ([[plus] ordwk / minus ordwk]) weekday  
  
plus          = "+"  
  
minus         = "-"  
  
ordwk         = 1DIGIT / 2DIGIT      ;1 to 53  
  
weekday       = "SU" / "MO" / "TU" / "WE" / "TH" / "FR" / "SA"  
;Corresponding to SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY,  
;FRIDAY, SATURDAY and SUNDAY days of the week.  
  
bymodaylist   = monthdaynum / ( monthdaynum *(", " monthdaynum) )  
  
monthdaynum    = ([[plus] ordmoday) / (minus ordmoday)  
  
ordmoday      = 1DIGIT / 2DIGIT      ;1 to 31  
  
byyrdaylist   = yeardaynum / ( yeardaynum *(", " yeardaynum) )  
  
yeardaynum     = ([[plus] ordyrday) / (minus ordyrday)  
  
ordyrday      = 1DIGIT / 2DIGIT / 3DIGIT ;1 to 366  
  
bywkno1list   = weeknum / ( weeknum *(", " weeknum) )  
  
weeknum        = ([[plus] ordwk) / (minus ordwk)  
  
bymolist      = monthnum / ( monthnum *(", " monthnum) )
```


Internet Draft

C&S Core Object Specification

March 6, 1998

```
monthnum    = 1DIGIT / 2DIGIT          ;1 to 12

bysplist    = setposday / ( setposday *("," setposday) )

setposday   = yeardaynum
```

Description: If the property permits, multiple "recur" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. The value type 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). The components are not ordered in any particular sequence. Individual components MAY only be specified once.

The FREQ component identifies the type of recurrence rule. This component MUST be specified in the recurrence rule. Valid values include SECONDLY, to specify repeating events based on an interval of a second or more; MINUTELY, to specify repeating events based on an interval of a minute or more; 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 recurrence rule repeats. The default value is "1" or every minute for a MINUTELY rule, 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.

The UNTIL component defines a date-time value which bounds the recurrence rule in an inclusive manner. If the value specified by UNTIL is synchronized with the specified recurrence, this date-time becomes the last instance of the recurrence. 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 range-bound the recurrence. This component is ignored if the "UNTIL" component is also present.

The BYSECOND component specifies a COMMA character (ASCII decimal 44) separated list of seconds within a minute. Valid values are 0 to 59.

The BYMINUTE component specifies a COMMA character (ASCII decimal 44) separated list of minutes within an hour. Valid values are 0 to 59. The BYHOUR component specifies a COMMA character (ASCII decimal 44) separated list of hours of the day. Valid values are 0 to 23.

The BYDAY component specifies a COMMA character (ASCII decimal 44) separated list of days of the week; MO, indicates Monday; TU,

Dawson/Stenerson

28

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

indicates Tuesday; WE, indicates Wednesday; TH, indicates Thursday; FR, indicates Friday; SA, indicates Saturday; SU, indicates Sunday.

Each BYDAY value 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. If an integer modifier is not present, it means all days of this type within the specified frequency. For example, within a MONTHLY rule, MO represents all Mondays within 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.

Each BYMONTHDAY value MAY include a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of the specific day of the month within the MONTHLY rule. If an integer modifier is not present, it means all days of this type within the specified frequency. For example, within a MONTHLY rule, -10 represents the tenth to the last day of the month.

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 BYWEEKNO component specifies a comma-separated list of weeks of the year. Valid values are 1 to 53. 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 component is only valid for YEARLY rules.

Each BYWEEKNO value MAY include a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of the specific week of the year within the YEARLY rule. If an integer modifier is not present, it means all days of this type within the specified frequency. For example, within a YEARLY rule, 3 represents the third week of the year.

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

The WKST component 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, and a BYDAY component is specified. The default value is MO.

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

Dawson/Stenerson

29

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

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

Each BYSETPOS value MAY include a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of the specific occurrence within the set of events specified by the rule.

If BYxxx component values are found which are beyond the available scope (ie, BYMONTHDAY=30 in February), they are simply ignored.

Information, not contained in the rule, necessary to determine the various recurrence instance start time and dates are derived from the Start Time (DTSTART) entry attribute. For example, "FREQ=YEARLY;BYMONTH=1" doesn't specify a specific day within the month or a time. This information would be the same as what is specified for DTSTART.

BYxxx components modify the recurrence in some manner. BYxxx components for a period of time which is the same or greater than the frequency generally reduce or limit the number of occurrences of the recurrence generated. For example, "FREQ=DAILY;BYMONTH=1" reduces the number of recurrence instances from all days (if BYMONTH tag is not present) to all days in January. BYxxx components for a period of

time less than the frequency generally increase or expand the number of occurrences of the recurrence. For example, "FREQ=YEARLY;BYMONTH=1,2" increases the number of days within the yearly recurrence set from 1 (if BYMONTH tag is not present) to 2.

If only one BYxxx component is specified in the recurrence rule, the list of "n" unique values would cause "n" occurrences of the recurrence within each specified frequency interval, where each unique list value is substituted in the appropriate date position within DTSTART for each such occurrence.

If multiple BYxxx components are specified, then after evaluating the specified FREQ and INTERVAL components, the BYxxx components are applied to the current set of evaluated occurrences in the following order: BYMONTH, BYWEEKNO, BYYEARDAY, BYMONTHDAY, BYDAY, BYHOUR, BYMINUTE, BYSECOND and BYSETPOS.

Here is an example of evaluating multiple BYxxx components.

```
DTSTART;TZID=EST=19970105T083000
RRULE:FREQ=YEARLY;INTERVAL=2;BYMONTH=1;BYDAY=SU;BYHOUR=8,9;
BYMINUTE=30
```

First, the "INTERVAL=2" would be applied to "FREQ=YEARLY" to arrive at "every other year". Then, "BYMONTH=1" would be applied to arrive at "every January, every other year". Then, "BYDAY=SU" would be applied to arrive at "every Sunday in January, every other year". Then, "BYHOUR=8,9" would be applied to arrive at "every Sunday in

Dawson/Stenerson

30

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

January at 8 AM and 9 AM, every other year". Then, "BYMINUTE=30" would be applied to arrive at "every Sunday in January at 8:30 AM and 9:30 AM, every other year". Similarly, if the BYMINUTE, BYHOUR, BYDAY, BYMONTHDAY or BYMONTH component were missing, the appropriate minute, hour, day or month would have been retrieved from the "DTSTART" property.

No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

Example: The following is a rule which specifies 10 meetings which occur every other day:

```
FREQ=DAILY;COUNT=10;INTERVAL=2
```

There are other examples specified in the "RRULE" specification.

4.3.11 Text

Value Name: TEXT

Purpose This value type is used to identify values that contain human readable text.

Formal Definition: The value type is defined by the following notation:

```
text      = text-value *(", " text-value)
; Folded according to folded-value syntax above or
; encoded according to param-value syntax above.

text-value = *(SAFE-CHAR / ":" / DQUOTE / ESCAPED-CHAR)
```

```
ESCAPED-CHAR = "\\\" / "\\;" / "\\," / "\\N\" / "\\n")
; \\ encodes \, \\N encodes newline
; \; encodes ;, \, encodes ,
```

```
NON-ASCII  = %x80-FF
; use restricted by charset parameter
; on outer MIME object
```

```
SAFE-CHAR  = %x20-21 / %x23-2B / %x2D-39 / %x3C-5B
             %x5D-7E / NON-ASCII
; Any character except CTLs, DQUOTE, ";", ":", "\"", ",",
```

Description: If the property permits, multiple "text" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. The character set and language in which the text is represented is controlled by the "LANGUAGE" property parameters.

An intentional formatted text line break MAY only be included in a "TEXT" property value by representing the line break with the character sequence of BACKSLASH (ASCII decimal 92), followed by a

Dawson/Stenerson

31

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

LATIN SMALL LETTER N (ASCII decimal 110) or a LATIN CAPITAL LETTER N (ASCII decimal 78), that is "\n" or "\N".

The "TEXT" property values may also contain special characters that are used to signify delimiters, such as a COMMA character for lists of values or a SEMICOLON character for structured values. In order to

support the inclusion of these special characters in "TEXT" property values, they need to be escaped with a BACKSLASH character. A BACKSLASH character (ASCII decimal 92) in a "TEXT" property value MUST be escaped with another BACKSLASH character. A COMMA character in a "TEXT" property value MUST be escaped with a BACKSLASH character (ASCII decimal 92). A SEMICOLON character in a "TEXT" property value MUST be escaped with a BACKSLASH character (ASCII decimal 92). However, a COLON character in a "TEXT" property value SHALL NOT be escaped with a BACKSLASH character. Example: A multiple line value of:

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

would be represented as:

```
Project XYZ Final Review\nConference Room - 3B\nCome Prepared.
```

[4.3.12](#) Time

Value Name: TIME

Purpose: This value type is used to identify values that contain a time of day.

Formal Definition: The data type is defined by the following notation:

```
time                = time-value *("," [WSP] time-value)

time-value           = time-hour time-minute time-second [time-utc]

time-hour            = 2DIGIT           ;00-23
time-minute          = 2DIGIT           ;00-59
time-second          = 2DIGIT           ;00-60
;The "60" value is used to account for years with "leap" seconds.

time-utc             = "Z"
```

Description: If the property permits, multiple "time" values MAY be specified using a COMMA character (ASCII decimal 44) separator character. No additional content value encoding (i.e., BACKSLASH character encoding) is defined for this value type.

The "TIME" data type is used to identify values that contain a time of day. The format is based on 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). Fractions of a second are not supported by this format.

In parallel to the "DATE-TIME" definition above, the "TIME" data type expresses time values in three forms:

Note: The form of time with UTC offset MUST NOT be used. For example, the following is NOT VALID for a time value:

230000-0800

FORM #1 LOCAL TIME

The local time form is simply a time value that does not contain the UTC designator nor does it reference a time zone. For example, 11 PM:

230000

Time values of this type are said to be "floating" and are not bound to any time zone in particular. They are used to represent the same hour, minute, and second value regardless of which time zone is currently being observed. 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, no matter which timezone the person is in. 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 whatever timezone the ATTENDEE is in at any given moment. This means that two ATTENDEEs may participate in the same event at different UTC times; floating time should only be used where that is reasonable behavior.

In most cases, a fixed time is desired. To properly communicate a fixed time in a property value, either UTC time or local time with time zone reference MUST be specified.

The use of local time in a TIME value without the TZID property parameter is to be interpreted as a local time value, regardless of the existence of "VTIMEZONE" calendar components in the iCalendar object.

FORM #2: DATE WITH UTC TIME

UTC time, or absolute time, is identified by a LATIN CAPITAL LETTER Z suffix character (ASCII decimal 90), the UTC designator, appended to

the time value. For example, the following represents 0700 UTC:

070000Z

The TZID property parameter MUST NOT be applied to TIME properties whose time values are specified in UTC.

Dawson/Stenerson

33

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

FORM #3: DATE WITH LOCAL TIME AND TIME ZONE REFERENCE

The local time with reference to time zone information form is identified by the use the TZID property parameter to reference the appropriate time zone definition. TZID is discussed in detail in the section on Time Zone.

Example: The following represents 8:30 AM in New York, five hours behind UTC, in each of the three formats using the "DTSTART" property:

DTSTART:19980312T083000

DTSTART:19980312T133000Z

DTSTART;TZID=America-New York:19980312T083000

[4.3.13](#) URI

Value Name: URI

Purpose: This value type is used to identify values that contain a uniform resource identifier (URI) type of reference to the property value.

Formal Definition: The data type is defined by the following notation:

uri = <As defined by any IETF RFC>

Description: 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 URI value formats in [RFC 1738](#), [RFC 2111](#) and any other IETF registered value format MAY be specified.

Any IANA registered URI format MAY be used. These include, but are not limited to, those defined in [RFC 1738](#) and [RFC 2111](#).

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" ":" [WSP] "VCALENDAR" CRLF
            icalbody
            "END" ":" [WSP] "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:19970714T170000Z
DTEND:19970715T045959Z
SUMMARY:Bastille Day Party
END:VEVENT
END:VCALENDAR
```

| | | |
|------------------|-------------------------------|------------------------|
| Dawson/Stenerson | 35 | Expires September 1998 |
| Internet Draft | C&S Core Object Specification | March 6, 1998 |

4.5 Property

A property is the definition of an individual attribute describing a calendar property or a calendar component. A property takes the form defined by the "contentline" notation defined in [section 4.1.1](#).

The following is an example of a property:

```
DTSTART:19960415T133000Z
```

This memo 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".

4.6 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 express a particular calendar semantic. For example, the calendar component MAY specify an event, a to-do, a journal entry, time zone information, or free/busy time information, or an alarm.

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

```
icalbody    = calprops 1*component

calprops    = [calscale] [method] prodid version

component   = 1*(eventc / todoc / journalc / freebusyc /
               / timezonec / x-name / iana-token)
```

An iCalendar object must include the "PRODID" and "VERSION" calendar properties. In addition, it must include at least one calendar component. Special forms of iCalendar objects are possible to publish just busy time (i.e., only a "VFREEBUSY" calendar component) or time zone (i.e., only a "VTIMEZONE" calendar component) information. In addition, a complex iCalendar object is possible that is used to capture a complete snapshot of the contents of a calendar (e.g., composite of many different calendar components). More commonly, an iCalendar object will consist of just a single "VEVENT", "VTODO" or "VJOURNAL" calendar component.

[4.6.1](#) Event Component

Component Name: "VEVENT"

Purpose: Provide a grouping of component properties that describe an event.

Dawson/Stenerson 36 Expires September 1998

Internet Draft C&S Core Object Specification March 6, 1998

Format Definition: A "VEVENT" calendar component is defined by the following notation:

```
eventc      = "BEGIN" ":" [WSP] "VEVENT" CRLF
              eventprop *alarmc
              "END" ":" [WSP] "VEVENT" CRLF

eventprop   = *attach *attendee *categories [class] *comment
              *contact [created] [description] [dtend / duration]
              dtstart *exdate *extrule [geo] [last-mod] [location]
              [organizer] [priority] [rstatus] *related *resources
```

*rdate *rrule dtstamp [seq] [status] summary
[transp] uid [url] [recurid]

Description: A "VEVENT" calendar component is a grouping of component properties and an OPTIONAL "VALARM" 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. Generally, these events will take up time on an individual calendar. Hence, the event will appear as an opaque interval in a search for busy time. Alternately, the event MAY have its Time Transparency set to "TRANSPARENT" in order to prevent blocking of the event in searches for busy time.

The "VEVENT" is also the calendar component used to specify an anniversary or daily reminder within a calendar. These events have a DATE value type for the "DTSTART" and "DTEND" properties instead of the default data type of DATE-TIME. If such a "VEVENT" has an end time, it MUST be specified as a DATE value also. The anniversary type of "VEVENT" MAY span more than one date (i.e, "DTEND" property value is set to a calendar date after the "DTSTART" property value).

The "DTSTART" property for a "VEVENT" specifies the inclusive start of the event. For recurring events, it also specifies the very first instance in the recurrence set. The "DTEND" property for a "VEVENT" calendar component specifies the non-inclusive end of the event. For cases where a "VEVENT" calendar component specifies a "DTSTART" property with a DATE data type but no "DTEND" property, the events non-inclusive end is the end of the calendar date specified by the "DTSTART" property. For cases where a "VEVENT" calendar component specifies a "DTSTART" property with a DATE-TIME data type but no "DTEND" property, the event ends on the same calendar date and time of day specified by the "DTSTART" property.

The "VEVENT" calendar component MUST include the "DTSTAMP", "DTSTART", "SUMMARY" and "UID" properties. In addition, it MUST include the "SEQUENCE" property, if it's value is greater than zero.

The "VEVENT" calendar component cannot be nested within another calendar component. The "VEVENT" calendar components MAY be related to each other or to a "VTODO" or "VJOURNAL" calendar component with the "RELATED-TO" property.

Example: The following is an example of the "VEVENT" calendar

component used to represent a meeting that will also be opaque to searches for busy time:

```
BEGIN:VEVENT
UID:19970901T130000Z-123401@host.com
DTSTAMP:19970901T1300Z
DTSTART:19970903T163000Z
DTEND:19970903T190000Z
SUMMARY:Annual Employee Review
CLASS:PRIVATE
CATEGORIES:BUSINESS,HUMAN RESOURCES
END:VEVENT
```

The following is an example of the "VEVENT" calendar component used to represent a reminder that will not be opaque, but rather transparent, to searches for busy time:

```
BEGIN:VEVENT
UID:19970901T130000Z-123402@host.com
DTSTAMP:19970901T1300Z
DTSTART:19970401T163000Z
DTEND:19970402T010000Z
SUMMARY:Laurel is in sensitivity awareness class.
CLASS:PUBLIC
CATEGORIES:BUSINESS,HUMAN RESOURCES
TRANSP:TRANSPARENT
END:VEVENT
```

The following is an example of the "VEVENT" calendar component used to represent an anniversary that will occur annually. Since it takes up no time, it will not appear as opaque in a search for busy time; no matter what the value of the "TRANSP" property indicates:

```
BEGIN:VEVENT
UID:19970901T130000Z-123403@host.com
DTSTAMP:19970901T1300Z
DTSTART:19971102
SUMMARY:Our Blissful Anniversary
CLASS:CONFIDENTIAL
CATEGORIES:ANNIVERSARY,PERSONAL,SPECIAL OCCASION
RRULE:FREQ=YEARLY
END:VEVENT
```

4.6.2 To-do Component

Component Name: VTODO

Purpose: Provide a grouping of calendar properties that describe a to-do.

Formal Definition: A "VTODO" calendar component is defined by the

following notation:

Dawson/Stenerson

38

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
todoc      = "BEGIN" ":" [WSP] "VTODO" CRLF
            todoprop *alarmc
            "END" ":" [WSP] "VTODO" CRLF

todoprop    = *attach *attendee *categories [class] *comment
            [completed] *contact [created] [description] dtstamp
            [dtstart] [due / duration] *exdate *exrule [geo]
            [last-mod] [location] [organizer] [percent] priority
            [rstatus] *related *resources *rdate *rrule
            [recurid] [seq] [status] summary uid [url]
```

Description: A "VTODO" calendar component is a grouping of component properties and an OPTIONAL "VALARM" 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".

The "VTODO" calendar component MUST include the "DTSTAMP", "PRIORITY", "SUMMARY" and "UID" properties. In addition, it MUST include the "SEQUENCE" property, if it's value is greater than zero.

The "VTODO" calendar component cannot be nested within another calendar component. If "VTODO" calendar components need to be related to each other or to a "VTODO" or "VJOURNAL" calendar component, they can specify a relationship with the "RELATED-TO" property.

A "VTODO" calendar component without the "DTSTART" and "DUE" (or "DURATION") properties specifies a to-do that is associated with each successive calendar dates, until it is completed.

Example: The following is an example of a "VTODO" calendar component:

```
BEGIN:VTODO
UID:19970901T130000Z-123404@host.com
DTSTAMP:19970901T1300Z
DTSTART:19970415T133000Z
DUE:19970416T045959Z
SUMMARY:1996 Income Tax Preparation
CLASS:CONFIDENTIAL
CATEGORIES:FAMILY,FINANCE
PRIORITY:1
STATUS:NEEDS-ACTION
END:VEVENT
```

4.6.3 Journal Component

Component Name: VJOURNAL

Purpose: Provide a grouping of component properties that describe a journal entry.

Formal Definition: A "VJOURNAL" calendar component is defined by the following notation:

Dawson/Stenerson

39

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
journalc  = "BEGIN" ":" [WSP] "VJOURNAL" CRLF
           jourprop
           "END" ":" [WSP] "VJOURNAL" CRLF

jourprop  = *attach *attendee *categories [class] *comment
           *contact [created] description dtstart dtstamp
           *exdate *exrule [last-mod] [organizer] [recurid]
           *related *rdate *rrule [rstatus] [seq] summary
           uid [url]
```

Description: A "VJOURNAL" calendar component is a grouping of component properties that represent one or more descriptive text notes on a particular calendar date. The "DTSTART" property is used to specify the calendar date that the journal entry is associated with. Generally, it will have a DATE value data type, but it MAY also be used to specify a DATE-TIME value data type. Examples of a journal entry include a daily record of a legislative body or a journal entry of individual telephone contacts for the day or an ordered list of accomplishments for the day. The "VJOURNAL" calendar component can also be used to associate a document with a calendar date.

The "VJOURNAL" calendar component does not take up time on a calendar. Hence, it does not play a role in free or busy time searches - - it is as though it has a time transparency value of TRANSPARENT. It is transparent to any such searches.

The "VJOURNAL" calendar component MUST include the "DTSTAMP", "DTSTART", "DESCRIPTION", "SUMMARY" and "UID" properties. In addition, it MUST include the "SEQUENCE" property, if its value is greater than zero.

The "VJOURNAL" calendar component cannot be nested within another


```
1*freebusy [last-mod] [rstatus] [seq] uid
[url]
;This set of properties is used for free/busy time reply.
```

```
busytime = attendee dtstamp dtstart dtend *comment 1*freebusy
[url]
```

Description: A "VFREEBUSY" calendar component is a grouping of component properties that represents either a request for, a reply to a request for free or busy time information or a published set of busy time information.

When used to request free/busy time information, the "VFREEBUSY" calendar component MUST include the "ATTENDEE", "DTSTAMP", "DTSTART", "DTEND", and "UID" properties. In addition, it MUST include the "SEQUENCE" property, if it's value is greater than zero.

When used to reply to a request for free/busy time, the "VFREEBUSY" calendar component MUST include the "ATTENDEE", "DTSTAMP", "FREEBUSY", and "UID" properties. In addition, it MUST include the "SEQUENCE" property, if it's value is greater than zero.

When used to publish busy time, the "VFREEBUSY" calendar component MUST include the "ATTENDEE", "DTSTAMP", "DTSTART", "DTEND", "FREEBUSY" properties.

The "VFREEBUSY" calendar component cannot be nested within another calendar component. Multiple "VFREEBUSY" calendar components MAY be specified within an iCalendar object. This permits the grouping of Free/Busy information into logical collections, such as monthly groups of busy time information.

The "VFREEBUSY" calendar component is intended for use in iCalendar object methods 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 "VFREEBUSY" calendar component to describe the Free/Busy information.

When present in a "VFREEBUSY" calendar component, the "DTSTART" and "DTEND" properties 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 "VFREEBUSY" calendar component. Any recurring events are resolved into their individual busy time periods using the "VFREEBUSY" property.

Example: The following is an example of a "VFREEBUSY" calendar component used to request free or busy time information:

```
BEGIN:VFREEBUSY
ORGANIZER:MAILTO:jane_doe@host1.com
ATTENDEE:MAILTO:john_public@host2.com
DTSTART:19971015T050000Z
DTEND:19971016T050000Z
DTSTAMP:19970901T083000Z
SEQUENCE:0
UID:19970901T0830000-uid1@host1.com
END:VFREEBUSY
```

The following is an example of a "VFREEBUSY" calendar component used to reply to the request with busy time information:

```
BEGIN:VFREEBUSY
ATTENDEE:MAILTO:john_public@host2.com
DTSTAMP:19970901T100000Z
SEQUENCE:0
UID:19970901T0830000-uid1@host1.com
FREEBUSY;VALUE=PERIOD:19971015T050000Z/PT8H30M,
  19971015T160000Z/PT5H30M,19971015T223000Z/PT6H30M
URL:http://host2.com/pub/busy/jpublic-01.vcs
COMMENT:This iCalendar file contains busy time information for
  the next three months.
END:VFREEBUSY
```

The following is an example of a "VFREEBUSY" calendar component used to published busy time information.

```
BEGIN:VFREEBUSY
ATTENDEE:jsmith@host.com
DTSTART:19980313T141711Z
```

```
DTEND:19980410T141711Z
FREEBUSY:19980314T233000Z/19980315T003000Z
```

```
FREEBUSY:19980316T153000Z/19980316T163000Z
FREEBUSY:19980318T030000Z/19980318T040000Z
URL:http://www.host.com/calendar/busytime/jsmith.ifb
END:VFREEBUSY
```

4.6.5 Timezone Component

Component Name: VTIMEZONE

Purpose: Provide a grouping of component properties that defines a time zone.

Formal Definition: A "VTIMEZONE" calendar component is defined by the following notation:

```
timezonec = "BEGIN" ":" [WSP] "VTIMEZONE" CRLF
           tzid [last-mod] [tzurl] 1*(standardc / daylightc)
           "END" ":" [WSP] "VTIMEZONE" CRLF

standardc = "BEGIN" ":" [WSP] "STANDARD" CRLF
           tzprop
           "END" ":" [WSP] "STANDARD" CRLF

daylightc = "BEGIN" ":" [WSP] "DAYLIGHT" CRLF
           tzprop
           "END" ":" [WSP] "DAYLIGHT" CRLF

tzprop     = *comment dtstart [rdate / rrule]
           *tzname tzoffsetto tzoffsetfrom
```

Description: 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 starting from 1967. Each line represents a description or rule for a particular observance.

Effective Observance Rule

Internet Draft

C&S Core Object Specification

March 6, 1998

| Date | (Date/Time) | Offset | Abbreviation |
|-----------|-------------------------|--------|--------------|
| 1967-* | lastSun 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, to-dos or journal entries, 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.

If present, the "VTIMEZONE" calendar component defines the set of Standard Time and Daylight Saving Time observances (or rules) for a particular time zone for a given interval of time. The "VTIMEZONE" calendar component cannot be nested within other calendar components. Multiple "VTIMEZONE" calendar components MAY exist in an iCalendar object. In this situation, each "VTIMEZONE" MUST represent a unique time zone definition. This is necessary for some classes of events, such as airline flights, that start in one time zone and end in another.

The "VTIMEZONE" calendar component MUST be present if the iCalendar object contains an RRULE that generates dates on both sides of a time zone shift (e.g. both in Standard Time and Daylight Saving Time) UNLESS the iCalendar object intends to convey a floating time (See the section "4.1.10.11 Time" for proper interpretation of floating time). It MAY be present if the iCalendar object does not contain such a RRULE. In addition, if a RRULE is present, there must be valid time zone information for all recurrence instances.

The "VTIMEZONE" calendar component MUST include the "TZID" property and at least one definition of a standard or daylight component. The standard or daylight component MUST include the "DTSTART", "TZOFFSETFROM" and "TZOFFSETTO" properties.

Each "VTIMEZONE" calendar component consists of a collection of one or more sub-components that describe the rule for a particular observance (either a Standard Time or a Daylight Saving Time observance). The "STANDARD" sub-component consists of a collection of properties that describe Standard Time. The "DAYLIGHT" sub-component consists of a collection of properties that describe Daylight Saving Time. In general this collection of properties consists of:

- the first onset date-time for the observance

Dawson/Stenerson

44

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

- the last onset date-time for the observance, if a last onset is known.
- the offset to be applied for the observance
- a rule that describes the day and time when the observance takes effect
- an optional name for the observance

For a given time zone, there may be multiple unique definitions of the observances over a period of time. Each observance is described using either a "STANDARD" or "DAYLIGHT" sub-component. The collection of these sub-components is used to describe the time zone for a given period of time. The offset to apply at any given time is found by locating the observance that has the last onset date and time before the time in question, and using the offset value from that observance.

The top-level properties in a "VTIMEZONE" calendar component are:

The mandatory "TZID" property is a text value that uniquely identifies the VTIMZONE calendar component within the scope of an iCalendar object.

The optional "LAST-MODIFIED" property is a UTC value that specifies the date and time that this time zone definition was last updated.

The optional "TZURL" property is url value that points to a published VTIMEZONE definition.

The collection of properties that are used to define the STANDARD and DAYLIGHT sub-components include:

The mandatory "DTSTART" property gives the effective onset date and local time for the time zone sub-component definition. "DTSTART" in this usage MUST be specified as a local DATE-TIME value.

The mandatory "TZOFFSETFROM" property gives the UTC offset which is in use when the onset of this time zone observance begins.

"TZOFFSETFROM" is combined with "DTSTART" to define the effective onset for the time zone sub-component definition. For example, the following represents the time at which the observance of Standard Time took effect in 1967 for the eastern United States:

```
DTSTART:19671029T020000
```

```
TZOFFSETFROM:-0400
```

The mandatory "TZOFFSETTO" property gives the UTC offset for the time zone sub-component (Standard Time or Daylight Saving Time) when this observance is in use.

Dawson/Stenerson

45

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

The optional "TZNAME" property is the customary name for the time zone. It may be specified multiple times, to allow for specifying multiple language variants of the time zone names. This could be used for displaying dates.

If specified, the onset for the observance defined by the time zone sub-component is defined by either the "RRULE" or "RDATE" property. If neither are specified, only one sub-component can be specified in the "VTIMEZONE" calendar component and it is assumed that the single observance specified is always in effect.

The "RRULE" property defines the recurrence rule for the onset of the observance defined by this time zone sub-component. Some specific requirements for the usage of RRULE for this purpose:

- If observance is known to have an effective end date, the "UNTIL" recurrence rule parameter MUST be used to specify the last valid onset of this observance (i.e., the UNTIL date-time will be equal to the last instance generated by the recurrence pattern). It MUST be specified as a full DATE-TIME value, using local time plus the UTC offset.
- The local time and UTC offset values provided in the "DTSTART" property for the given observance MUST be used when generating

the onset date-time values (instances) from the RRULE. This interpretation of the "DTSTART" UTC offset value is specific to "RRULE" usage in the "VTIMEZONE" specification.

Alternatively, the "RDATE" property MAY be used to define the onset of the observance by giving the individual onset date and times. "RDATE" in this usage MUST be specified as a local DATE-TIME value with UTC offset.

The optional "COMMENT" property is also allowed for descriptive explanatory text.

Example: The following are examples of the "VTIMEZONE" calendar component:

This is a simple example showing time zone information for the Eastern United States using "RDATE" property. Note that this is only suitable for a recurring event that starts on or later than April 6, 1997 at 02:00:00 EST (i.e., the earliest effective transition date and time) and ends no later than April 7, 1998 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 December 31, 1997.

```
BEGIN:VTIMEZONE
TZID:America-New_York
LAST-MODIFIED:19870101T000000Z
BEGIN:STANDARD
DTSTART:19971026T020000
RDATE:19971026T020000
```

Dawson/Stenerson

46

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
TZOFFSETFROM:-0400
TZOFFSETO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19971026T020000
RDATE:19970406T020000
TZOFFSETFROM:-0500
TZOFFSETO:-0400
TZNAME:EDT
END:DAYLIGHT
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
TZID:America-New_York
LAST-MODIFIED:19870101T000000Z
TZURL:http://zones.stds_r_us.net/tz/America-New_York
BEGIN:STANDARD
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
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
TZID:America-New_York
LAST-MODIFIED:19870101T000000Z
BEGIN:STANDARD
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
```

Dawson/Stenerson

47

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4;UNTIL=19980404T070000
TZOFFSETFROM:-0500
```



```
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
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
TZID:America-New_York
LAST-MODIFIED:19870101T000000Z
BEGIN:STANDARD
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4;UNTIL=19980404T070000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
BEGIN:DAYLIGHT
DTSTART:19990424T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=4
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
END:VTIMEZONE
```

4.6.6 Alarm Component

Component Name: VALARM

Purpose: Provide a grouping of component properties that define an alarm.

Formal Definition: A "VALARM" calendar component is defined by the following notation:

```
alarmc      = "BEGIN" ":" [WSP] "VALARM" CRLF
              (audioprop / dispprop / emailprop
              / procprop)
              "END" ":" [WSP] "VALARM" CRLF
```

Internet Draft

C&S Core Object Specification

March 6, 1998

```
audioprop = alarmtype trigger [duration repeat] attach

dispprop  = alarmtype description trigger [duration
        repeat]

emailprop = alarmtype 1*attendee *attach description
        trigger [duration repeat] summary

procprop  = alarmtype attach [description] trigger [duration
        repeat]
```

Description: A "VALARM" calendar component is a grouping of component properties that is a reminder or alarm for an event or a to-do. For example, it may be used to define a reminder for a pending event or an overdue to-do.

The "VALARM" calendar component MUST include the "ALARM-TYPE" and "TRIGGER" properties. In addition, an AUDIO type of alarm MUST include the "ATTACH" property to point to a digital sound resource to be played. The DISPLAY type of alarm MUST include the "DESCRIPTION" property. The EMAIL type of alarm MUST include the "ATTENDEE", "DESCRIPTION" and "SUMMARY" properties. The PROCEDURE type of alarm MUST include the "ATTACH" property to point to a procedure resource to be invoked.

The "VALARM" calendar component MUST only appear within either a "VEVENT" or "VTODO" calendar component. The "VALARM" calendar components cannot be nested. Multiple "VALARM" calendar components MAY be specified.

The "TRIGGER" property specifies when the alarm will be triggered. The "TRIGGER" property specifies a duration prior to the start of an event or a to-do. The "TRIGGER" edge may be explicitly set to be relative to the "START" or "END" of the event or to-do with the "RELATED" parameter of the "TRIGGER" property. The "TRIGGER" property value type MAY alternately be set to an absolute calendar date and time of day value.

In an alarm set to trigger on the "START" of an event or to-do, the "DTSTART" property MUST be present in the associated event or to-do. In an alarm in a "VEVENT" calendar component set to trigger on the "END" of the event, the "DTEND" or "DTSTART" and "DURATION" properties MUST be present. In an alarm in a "VTODO" calendar component set to trigger on the "END" of the to-do, the "DUE" or

"DTSTART" and "DURATION" properties MUST be present.

The alarm MAY be defined such that it triggers repetitively. A definition of an alarm with a repeating trigger MUST include both the "DURATION" and "REPEAT" properties. The "DURATION" property specifies the delay period, after which the alarm will repeat. The "REPEAT" property specifies the number of additional repetitions that the alarm will be triggered. This count is in addition to the initial triggering of the alarm. Both of these properties must be present in order to specify a repeating alarm. If one of these two properties

Dawson/Stenerson

49

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

are absent, then the alarm will not repeat beyond the initial trigger.

The "ALARM-TYPE" property is used within the "VALARM" calendar component to specify the type of the alarm. The "ALARM-TYPE" property value of AUDIO specifies an alarm that triggers with an audio sound; a value of DISPLAY specifies an alarm that triggers with the "Calendar User Agent" displaying text; the value of EMAIL specifies an alarm that triggers the posting of an electronic email message to one or more email addresses; and the value of PROCEDURE specifies an alarm that triggers the execution of a procedure. The "ALARM-TYPE" property MUST specify just one of these valid values. Multiple categories of alarms MAY BE specified by defining more than one, different categories of "VALARM" calendar components within a "VEVENT" or "VTODO" calendar component.

In an AUDIO type of alarm, the "ATTACH" property in the "VALARM" calendar component MUST specify an audio sound file that is intended to be rendered as the alarm effect. If an "ATTACH" property is specified that does not refer to a digital sound resource, then no audio alarm will be sounded. If the digital sound resource is in a format that cannot be rendered, then no audio alarm will be sounded.

In a DISPLAY type of alarm, the "DESCRIPTION" property in the "VALARM" calendar component is intended to be displayed as the alarm effect. In an EMAIL category of alarm, the intended alarm effect is to use the "DESCRIPTION" property in the "VALARM" calendar component as the body text of an email message that is to be sent to the addresses specified by any "ATTENDEE" properties present in the "VALARM" calendar component. The "SUMMARY" property in the "VALARM" calendar component is intended to be used as the subject text for the email. Any "ATTACH" properties are sent as email attachments.

In a PROCEDURE type of alarm, the "ATTACH" property in the "VALARM"

calendar component MUST specify a procedure or program that is intended to be invoked as the alarm effect. If the procedure or program is in a format that cannot be rendered, then no procedure alarm will be invoked. If the "DESCRIPTION" property is present, its value specifies the argument string to be passed to the procedure or program. "Calendar User Agents" that receive an iCalendar object with this category of alarm, should allow the "Calendar User" to disable or otherwise ignore this type of alarm. While a very useful alarm capability, the PROCEDURE type of alarm should be treated by the "Calendar User Agent" as a potential security risk.

Example: The following example is for a "VALARM" calendar component that specifies an audio alarm that will sound at a precise time and repeat 4 more times at 15 minute intervals:

```
BEGIN:VALARM
TRIGGER;VALUE=DATE-TIME:19970317T133000Z
REPEAT:4
DURATION:PT15M
ALARM-TYPE: AUDIO
```

Dawson/Stenerson

50

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
ATTACH:ftp://host.com/pub/sounds/bell-01.wav
END:VALARM
```

The following example is for a "VALARM" calendar component that specifies a display alarm that will trigger 15 minutes before the scheduled start of the event or the due date/time of the to-do it is associated with and will repeat 2 more times at 15 minute intervals:

```
BEGIN:VALARM
TRIGGER:PT30M
REPEAT:2
DURATION:PT15M
ALARM-TYPE:DISPLAY
DESCRIPTION:Breakfast meeting with executive\n
team at 8:30 AM EST.
END:VALARM
```

The following example is for a "VALARM" calendar component that specifies an email alarm that will trigger 2 days before the scheduled due date/time of a to-do it is associated with. It does not repeat. The email has a subject, body and attachment link.

```
BEGIN:VALARM
TRIGGER:P2D
```

```
ALARM-TYPE:EMAIL
ATTENDEE:MAILTO:john_doe@host.com
SUMMARY:*** REMINDER: SEND AGENDA FOR WEEKLY STAFF MEETING ***
DESCRIPTION:A draft agenda needs to be sent out to the attendees
  to the weekly managers meeting (MGR-LIST). Attached is a
  pointer the document template for the agenda file.
ATTACH:http://host.com/templates/agenda.doc
END:VALARM
```

The following example is for a "VALARM" calendar component that specifies a procedural alarm that will trigger at a precise date/time and will repeat 23 more times at one hour intervals. The alarm will invoke a procedure file.

```
BEGIN:VALARM
TRIGGER;VALUE=DATE-TIME:19980101T050000Z
REPEAT:23
DURATION:PT1H
ALARM-TYPE:PROCEDURE
ATTACH:ftp://host.com/novo-procs/felizano.exe
END:VALARM
```

4.7 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" property and prior to any calendar component.

| | | |
|------------------|-------------------------------|------------------------|
| Dawson/Stenerson | 51 | Expires September 1998 |
| Internet Draft | C&S Core Object Specification | March 6, 1998 |

4.7.1 Calendar Scale

Property Name: CALSCALE

Purpose: This property defines the calendar scale used for the calendar information specified in the iCalendar object.

Value Type: TEXT

Conformance: Property MAY be specified in an iCalendar object. The default value is "GREGORIAN".

Description: This memo 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 memo.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
calscale    = "CALSCALE" ":" [WSP] calvalue CRLF
```

```
calvalue    = "GREGORIAN" / iana-token
```

Example: The following is an example of this property:

```
CALSCALE:GREGORIAN
```

4.7.2 Method

Property Name: METHOD

Purpose: This property defines the iCalendar object method associated with the calendar object.

Value Type: TEXT

Conformance: The property MAY be specified in an iCalendar object.

Description: When used in a MIME message entity, the value of this property MUST be the same as the Content-Type "method" parameter value. This property can only appear once within the iCalendar object. If either the "METHOD" property or the Content-Type "method" parameter is specified, then the other must also be specified.

No methods are defined by this specification. This is the subject of other specifications, such as the iCalendar Transport-independent Interoperability Protocol (iTIP) defined by [[ITIP](#)].

Dawson/Stenerson

52

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

If this property is not present in the iCalendar object, then no scheduling transaction MAY be assumed. In such cases, the iCalendar object is merely being used to transport a snapshot of some calendar information; without the intention of conveying a scheduling semantic.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

method = "METHOD" ":" [WSP] method-value CRLF

method-value = iana-token

Example: The following is a hypothetical example of this property to convey that the iCalendar object is a request for a meeting:

METHOD:REQUEST

4.7.3 Product Identifier

Property Name: PROPID

Purpose: This property specifies the identifier for the product that created the iCalendar object.

Value Type: TEXT

Conformance: The property MUST be specified in an iCalendar object but can only appear once.

Description: 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 property should not be used to alter the interpretation of an iCalendar object beyond the semantics specified in this memo. For example, it is not to be used to further the understanding of non-standard properties.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

prodid = "PROPID" ":" [WSP] pidvalue CRLF

pidvalue = text
;Any text that describes the product and version
;and that is generally assured of being unique.

Example: The following is an example of this property:

PROPID:-//ABC Corporation//NONSGML My Product//EN

4.7.4 Version

Property Name: VERSION

Purpose: This property specifies the identifier corresponding to the highest version number or the minimum and maximum range of the iCalendar specification that is required in order to interpret the iCalendar object.

Value Type: TEXT

Conformance: This property MUST be specified by in an iCalendar object, but MAY only be specified once.

Description: A value of "2.0" corresponds to this memo.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
version      = "VERSION" ":" [WSP] vvalue CRLF
```

```
vvalue       = "2.0"           ;This memo  
              / maxver  
              / (minver ";" [WSP] maxver)
```

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

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

Example: The following is an example of this property:

```
VERSION:2.0
```

4.8 Component Properties

The following properties MAY appear within calendar components, as specified by each component property definition.

4.8.1 Descriptive Component Properties

The following properties specify descriptive information about calendar components.

4.8.1.1 Attachment

Property Name: ATTACH

Purpose: The property provides the capability to associate a document object with a calendar component.

Dawson/Stenerson

54

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Value Type: The default value type for this property is URI. The value type MAY also be reset to BINARY in order to include inline binary encoded content information.

Conformance: The property MAY be specified in an iCalendar object.

Description: The property MAY only be specified within "VEVENT", "VTOD0", "VJOURNAL", or "VALARM" calendar components. This property MAY be specified multiple times within an iCalendar object.

Only the "ENCODING" and "VALUE", property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
attach      = ("ATTACH" ":" [WSP] uri  CRLF)
```

```
attach      =/ ("ATTACH" ";" [WSP] "ENCODING" "=" "b" ";" [WSP]
               "VALUE" "=" "text" ":" [WSP] binary
```

Example: The following are examples of this property:

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

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

4.8.1.2 Categories

Property Name: CATEGORIES

Purpose: This property defines the categories for a calendar component.

Value Type: TEXT

Conformance: The property MAY only be specified within "VEVENT", "VTOD0" or "VJOURNAL" calendar components.

Description: This property is used to specify categories or subtypes of the calendar component. The categories are useful in searching for

a calendar component of a particular type and category. Within the "VEVENT", "VTOD0" or "VJOURNAL" calendar components, more than one category MAY be specified as a list of categories separated by the COMMA character (ASCII decimal 44).

Only the "LANGUAGE" property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
categories = "CATEGORIES" [catparam] ":" [WSP] catvalue CRLF
```

```
catparam   = ";" [WSP] languageparm
```

Dawson/Stenerson

55

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
catvalue   = "ANNIVERSARY" / "APPOINTMENT" / "BUSINESS"
             / "EDUCATION" / "HOLIDAY" / "MEETING" / "MISCELLANEOUS"
             / "NON-WORKING HOURS" / "NOT IN OFFICE" / "PERSONAL"
             / "PHONE CALL" / "SICK DAY" / "SPECIAL OCCASION"
             / "TRAVEL" / "VACATION" / text
;These are just examples
;Used only in "VEVENT", "VTOD0" and "VJOURNAL" calendar components.
```

Example: The following are examples of this property:

```
CATEGORIES:APPOINTMENT,EDUCATION
```

```
CATEGORIES:MEETING
```

4.8.1.3 Classification

Property Name: CLASS

Purpose: This property defines the access classification for a calendar component.

Value Type: TEXT

Conformance: The property MAY only be specified in a "VEVENT", "VTOD0" or "VJOURNAL" calendar components. The property MAY only be specified once.

Description: 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 classification of an individual iCalendar component is useful when measured along with the other security components of a calendar system (e.g., calendar user authentication, authorization, access rights, access role, etc.). Hence, the semantics of the individual access classifications cannot be completely defined by this memo alone. Additionally, due to the "blind" nature of most exchange processes using this memo, these access classifications cannot 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 [ICMS] provides a broader description of the security system within a calendar application.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
class      = "CLASS" ":" [WSP] classvalue CRLF

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

Dawson/Stenerson

56

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Example: The following is an example of this property:

```
CLASS:PUBLIC
```

[4.8.1.4](#) Comment

Property Name: COMMENT

Purpose: This property specifies non-processing information intended to provide a comment to the calendar user.

Value Type: TEXT

Conformance: This property may be specified in an iCalendar object.

Description: The property MAY be specified multiple times.

Only the "ALTREP" or "LANGUAGE" property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

The property MAY be specified in any of the calendar components.

comment = "COMMENT" [commparam] ":" [WSP] text CRLF

commparam = [";" [WSP] altrepparm] [";" [WSP] languageparm]

Example: 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.

4.8.1.5 Description

Property Name: DESCRIPTION

Purpose: This property provides a more complete description of the calendar component, than that provided by the "SUMMARY" property.

Value Type: TEXT

Conformance: The property MAY be specified in the "VEVENT", "VTOD0" and "VJOURNAL" calendar components. The property MAY be specified multiple times only within a "VJOURNAL" calendar component. The property MAY be specified in "VALARM" calendar component.

Description: This property is used in the "VEVENT" and "VTOD0" to capture lengthy textual descriptions associated with the activity.

Dawson/Stenerson

57

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

This property is used in the "VJOURNAL" calendar component to capture one more textual journal entries.

This property is used in the "VALARM" calendar component to capture the display text for a DISPLAY category of alarm, to capture the body text for an EMAIL category of alarm and to capture the argument string for a PROCEDURE category of alarm.

Only the "ALTREP" or "LANGUAGE" property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

description = "DESCRIPTION" [descparam] ":" [WSP] text CRLF

descparam = [":" [WSP] altrepparm] [":" [WSP] languageparm]

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

DESCRIPTION:Meeting to provide technical review for "Phoenix"
design.\n Happy Face Conference Room. Phoenix design team
MUST attend this meeting.\n 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.

4.8.1.6 Geographic Position

Property Name: GEO

Purpose: This property specifies information related to the global position for the activity specified by a calendar component.

Value Type: Two FLOAT values.

Conformance: This property MAY be specified in an iCalendar object.

Description: 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:

decimal = degrees + minutes/60 + seconds/3600.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

geo = "GEO" ":" [WSP] geovalue CRLF

geovalue = float ";" [WSP] float
;Latitude and Longitude components

Example: The following is an example of this property:

GEO:37.386013;-122.082932

4.8.1.7 Location

Property Name: LOCATION

Purpose: The property defines the intended venue for the activity defined by a calendar component.

Value Type: TEXT

Conformance: This property MAY be specified in the "VEVENT" or "VTODO" calendar component.

Description: Specific venues such as conference or meeting rooms may be explicitly specified using this property. An alternate representation may be specified that is a URI that points to directory information with more structured specification of the location. For example, the alternate representation may specify either an LDAP URI pointing to an LDAP server entry or a CID URI pointing to a MIME body part containing a vCard for the location.

Only the "ALTREP" or "LANGUAGE" property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

location = "LOCATION [locparam] ":" [WSP] text CRLF

locparam = [";" [WSP] altrepparm] [";" [WSP] languageparm]

Example: The following are some examples of this property:

LOCATION:Conference Room - F123, Bldg. 002

LOCATION;ALTREP="http://xyzcorp.com/conf-rooms/f123.vcf":
Conference Room - F123, Bldg. 002

4.8.1.8 Percent Complete

Property Name: PERCENT-COMPLETE

Internet Draft

C&S Core Object Specification

March 6, 1998

Purpose: This property is used by an assignee or delegatee of a to-do to convey the percent completion of a to-do to the Organizer.

Value Type: INTEGER

Conformance: This property MAY only be specified in a "VTODO" calendar component.

Description: The property value is a positive integer between zero and one hundred. A value of "0" indicates the to-do has not yet been started. A value of "100" indicates that the to-do has been completed. Integer values in between indicate the percent partially complete.

When a to-do is assigned to multiple individuals, the property value indicates the percent complete for that portion of the to-do assigned to the assignee or delegatee. For example, if a to-do is assigned to both individuals "A" and "B". A reply from "A" with a percent complete of "70" indicates that "A" has completed 70% of the to-do assigned to them. A reply from "B" with a percent complete of "50" indicates "B" has completed 50% of the to-do assigned to them.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

percent = "PERCENT-COMPLETE" ":" [WSP] integer CRLF

Example: The following is an example of this property to show 39% completion:

PERCENT-COMPLETE:39

4.8.1.9 Priority

Property Name: PRIORITY

Purpose: The property defines the relative priority for a calendar component.

Value Type: INTEGER

Conformance: The property MAY only be specified in a "VEVENT" or "VTODO" calendar component.

Description: The priority is specified as an integer in the range zero to nine. 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. A value of nine (ASCII decimal 58) is the lowest priority.

A CUA with a three-level priority scheme of "HIGH", "MEDIUM" and "LOW" is mapped into this property such that a property value in the

Dawson/Stenerson

60

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

range of one (ASCII decimal 49) to four (ASCII decimal 52) specifies "HIGH" priority. A value of five (ASCII decimal 53) is the normal or "MEDIUM" priority. A value in the range of six (ASCII decimal 54) to nine (ASCII decimal 58) is "LOW" priority.

A CUA with a priority schema of "A1", "A2", "A3", "B1", "B2", ..., "C3" is mapped into this property such that a property value of one (ASCII decimal 49) specifies "A1", a property value of two (ASCII decimal 50) specifies "A2", a property value of three (ASCII decimal 51) specifies "A3", and so forth up to a property value of 9 (ASCII decimal 58) specifies "C3".

Other integer values are reserved for future use.

Within a "VEVENT" calendar component, this property specifies a priority for the event. This property may be useful when more than one event is scheduled for a given time period.

Within a "VTODO" calendar component, this property specified a priority for the to-do. This property is useful in prioritizing multiple action items for a given time period.

No property parameters MAY be specified on this property.

Format Definition: The property is specified by the following notation:

```
priority    = "PRIORITY" ":" [WSP] privalue CRLF
;Default is zero
```

```
privalue    = integer           ;Must be in the range [0..9]
; All other values are reserved for future use
```

The following is an example of a property with the highest priority:

The following is an example of a property with a next highest priority:

Example: The following is an example of a property with no priority. This is equivalent to not specifying the "PRIORITY" property.:

4.8.1.10 Resources

Purpose: This property defines the equipment or resources anticipated for an activity specified by a calendar entity..

| | | |
|------------------|----|------------------------|
| Dawson/Stenerson | 61 | Expires September 1998 |
|------------------|----|------------------------|

Internet Draft C&S Core Object Specification March 6, 1998

Value Type: TEXT

Conformance: This property MAY be specified in a "VEVENT" or "VTODO" calendar component.

Description: The property value is an arbitrary text. More than one resource MAY be specified as a list of resources separated by the COMMA character (ASCII decimal 44).

Only the "LANGUAGE" property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
resources = "RESOURCES" [resrcparam] ":" [WSP] resvalist CRLF
```

```
resrcparam = ";" [WSP] languageparm
```

```
resvalist = resvalue / resvalue "," [WSP] resvalist
```

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

;These are just examples.

Example: The following is an example of this property:

RESOURCES:EASEL,PROJECTOR,VCR

4.8.1.11 **Status**

Property Name: STATUS

Purpose: This property defines the overall status or confirmation for the calendar component.

Value Type: TEXT

Conformance: This property MAY only be specified in the "VEVENT", "VTODO", "VJOURNAL" calendar components.

Description: In a group scheduled calendar component, the property is used by the "Organizer" to provide a confirmation of the event to the "Attendees". For example in a "VEVENT" calendar component, the "Organizer" can indicate that a meeting is tentative, confirmed or cancelled. In a "VTODO" calendar component, the "Organizer" can indicate that an action item needs action, is completed, is in process or being worked on, or has been cancelled. In a "VJOURNAL" calendar component, the "Organizer" can indicate that a journal entry is draft, final or has been cancelled or removed.

No property parameter MAY be specified on this property.

Dawson/Stenerson

62

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Format Definition: The property is defined by the following notation:

status = "STATUS" ":" [WSP] statvalue CRLF

statvalue = "TENTATIVE" ;Indicates event is
 ;tentative.
 / "CONFIRMED" ;Indicates event is
 ;definite.
 / "CANCELLED" ;Indicates event was
 ;cancelled.
 ;Status values for a "VEVENT"

statvalue =/ "NEEDS-ACTION" ;Indicates to-do needs action.
 / "COMPLETED" ;Indicates to-do completed.
 / "IN-PROCESS" ;Indicates to-do in process of
 / "CANCELLED" ;Indicates to-do was cancelled.

;Status values for "VTOD0".

```
statvalue  =/ "DRAFT"           ;Indicates journal is draft.
            / "FINAL"           ;Indicates journal is final.
            / "CANCELLED"       ;Indicates journal is removed.
;Status values for "VJOURNAL".
```

Example: The following is an example of this property for a "VEVENT" calendar component:

STATUS:TENTATIVE

The following is an example of this property for a "VTOD0" calendar component:

STATUS:NEEDS-ACTION

The following is an example of this property for a "VJOURNAL" calendar component:

STATUS:DRAFT

[4.8.1.12](#) **Summary**

Property Name: SUMMARY

Purpose: This property defines a short summary or subject for the calendar component.

Value Type: TEXT

Conformance: The property MUST be specified in the "VEVENT", "VTOD0" and "VJOURNAL" calendar components. In addition, it MAY appear in the "VALARM" calendar component.

Description: This property is used in the "VEVENT", "VTOD0" and "VJOURNAL" calendar components to capture a short, one line summary about the activity or journal entry.

Dawson/Stenerson 63 Expires September 1998

Internet Draft C&S Core Object Specification March 6, 1998

This property is used in the "VALARM" calendar component to capture the subject of an EMAIL category of alarm.

Only the "ALTREP" and "LANGUAGE" property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

summary = "SUMMARY" [summparm] ":" [WSP] text CRLF

summparm = [";" [WSP] altrepparm] [";" [WSP] languageparm]

Example: The following is an example of this property:

SUMMARY:Department Party

4.8.2 Date and Time Component Properties

The following properties specify date and time related information in calendar components.

4.8.2.1 Date/Time Completed

Property Name: COMPLETED

Purpose: This property defines the date and time that a to-do was actually completed.

Value Type: DATE-TIME

Conformance: The property MAY only be specified in a "VTOD0" component in an iCalendar object.

Description: The date and time must be in a UTC format.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

completed = "COMPLETED" ":" [WSP] date-time CRLF

Example: The following is an example of this property:

COMPLETED:19960401T235959Z

4.8.2.2 Date/Time End

Property Name: DTEND

Purpose: This property specifies the date and time that a calendar component ends.

Value Type: The default value type is DATE-TIME. The value type MAY BE reset to a DATE value type.

Conformance: This property MAY only be specified within the "VEVENT", "VFREEBUSY", and "VTIMEZONE" calendar components.

Description: Within the "VEVENT" calendar component, this property defines the date and time by which the event ends. The value MUST be later in time than the value of the "DTSTART" property.

Within the "VFREEBUSY" calendar component, this property defines the end date and time for the free or busy time information. The time MUST be specified as in the UTC time format. The value MUST be later in time than the value of the "DTSTART" property.

Only the "VALUE" and "TZID" property parameters MAY be specified on this property. The "TZID" property parameter MAY only be specified when the value type is DATE-TIME.

Format Definition: The property is defined by the following notation:

dtend = "DTEND" [dtendparam] ":" [WSP] dtendval CRLF

dtendparam = [";" [WSP] "VALUE" "=" "DATE-TIME"]
[";" [WSP] "TZID" "=" paramtext]

dtendparam = / ";" [WSP] "VALUE" "=" "DATE"

dtendval = date-time / date
;Value MUST match value type

Example: The following is an example of this property:

DTEND:19960401T235959Z

DTEND;VALUE=DATE:19980704

4.8.2.3 Date/Time Due

Property Name: DUE

Purpose: This property defines the date and time that a to-do is expected to be completed.

Value Type: The default value type is DATE-TIME. The value type MAY BE reset to a DATE value type.

Conformance: The property MAY only be specified in a "VTOD0" calendar component and only once.

Description: The value MUST be a date/time equal to or after the DTSTART value, if specified.

Only the "VALUE" and "TZID" property parameters MAY be specified on this property. The "TZID" property parameter MAY only be specified when the value type is DATE-TIME.

Dawson/Stenerson

65

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Format Definition: The property is defined by the following notation:

due = "DUE" [dueparam]":" [WSP] dueval CRLF

dueparam = [";" [WSP] "VALUE" "=" "DATE-TIME"]
[";" [WSP] "TZID" "=" paramtext

dueparam =/ ";" [WSP] "VALUE" "=" "DATE"

dueval = date-time / date
;Value MUST match value type

Example: The following is an example of this property:

DUE:19960401T235959Z

[4.8.2.4](#) **Date/Time Start**

Property Name: DTSTART

Purpose: This property specifies when the calendar component begins.

Value Type: The default value type is DATE-TIME. The DATE-TIME value will be one of the forms defined for the DATE-TIME value type. The value type MAY BE reset to a DATE value type.

Conformance: This property MAY be specified within the "VEVENT", "VTOD", "VFREEBUSY", and "VTIMEZONE" calendar components.

Description: Within the "VEVENT" calendar component, this property defines the start date and time for the event. The property is REQUIRED in "VEVENT" calendar components. 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 "VFREEBUSY" calendar component, this property defines the start date and time for the free or busy time information. The time MUST be specified in UTC time.

Within the "VTIMEZONE" calendar component, this property defines the effective start date and time for a time zone specification. This

property is REQUIRED within "VTIMEZONE" calendar components. The time MUST be specified in the UTC time format.

Only the "VALUE" and "TZID" property parameters MAY be specified on this property. The "TZID" property parameter MAY only be specified when the value type is DATE-TIME.

Format Definition: The property is defined by the following notation:

```
dtstart      = "DTSTART" [dtstparam] ":" [WSP] dtstval CRLF
```

```
dtstparam    = [";" [WSP] "VALUE" "=" "DATE-TIME"]  
              [";" [WSP] "TZID" "=" paramtext]
```

Dawson/Stenerson

66

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
dtstparam    =/ ";" [WSP] "VALUE" "=" "DATE"
```

```
dtstval      = date-time / date  
;Value MUST match value type
```

Example: The following is an example of this property:

```
DTSTART:19980118T073000Z
```

[4.8.2.5](#) Duration

Property Name: DURATION

Purpose: The property specifies a duration of time .

Value Type: DURATION

Conformance: The property MAY be specified in a "VEVENT", "VTODO", "VFREEBUSY" or "VALARM" calendar components.

Description: In a "VEVENT" calendar component the property may be used to specify a duration of the event, instead of an explicit end date/time. In a "VTODO" calendar component the property may be used to specify a duration for the to-do, instead of an explicit due date/time. In a "VFREEBUSY" calendar component the property may be used to specify the interval of free time being requested. In a "VALARM" calendar component the property may be used to specify the delay period prior to repeating an alarm.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

duration = "DURATION" ":" [WSP] duration CRLF

Example: 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

[4.8.2.6](#) Free/Busy Time

Property Name: FREEBUSY

Purpose: The property defines one or more free or busy time intervals.

Value Type: PERIOD. The data and time value MUST be in a UTC time format.

Dawson/Stenerson

67

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Conformance: The property MAY only be specified in a "VFREEBUSY" calendar component.

Description: 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 MUST be a UTC time format.

The "FREEBUSY" property MAY include the "TYPE" property parameter to specify whether the information defines a FREE or BUSY time interval. The default "TYPE" property parameter value is BUSY.

If the "TYPE" property parameter is set to "BUSY", then the property MAY also include the "BSTAT" property parameter to provide added information about the busy time. For example, if the busy time is associated with a time interval that would be unavailable for scheduling in any case the parameter MAY BE set to UNAVAILABLE or if the busy time that has been tentatively scheduled the parameter MAY BE set to TENTATIVE. The default "BSTAT" property parameter value is BUSY, providing no additional busy status information.

"FREEBUSY" properties within the "VFREEBUSY" 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 "TYPE" and "BSTAT" property parameter type (e.g., all values of a particular "TYPE" and "BSTAT" listed together in a single property).

No property parameters other than the property specific "TYPE" and "BSTAT" parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
freebusy    = "FREEBUSY" [";" [WSP] fbparmlist] ":" [WSP] fbvalue
              CRLF

fbparmlist  = fb-free / fb-busy

fb-free     = "TYPE" "=" "FREE"

fb-busy     = ["TYPE" "=" "BUSY"] [";" [WSP] fbstatus]
;Default is BUSY

fbstatus    = "BSTAT" "="
              "BUSY"           ;Represents busy time interval.
              / "UNAVAILABLE" ;Represents busy, but unavailable
                          ;interval for cheduling; such as
                          ;out-of-office or non-working hours.
              / "TENTATIVE"   ;Represents busy, but only tentatively
                          ;scheduled interval.
              / iana-token    ;Represents other registered busy status
```

Dawson/Stenerson

68

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
              / x-name           ;Represents experimental busy status
;Default is BUSY

fbvalue      = period *["," [WSP] period]
;Time value MUST be in the UTC time format.
```

Example: The following are some examples of this property:

```
FREEBUSY;BSTAT=UNAVAILABLE:19970308T160000Z/PT8H30M
```

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

[4.8.2.7](#) Time Transparency

Property Name: TRANSP

Purpose: This property defines whether an event is transparent or not to busy time searches.

Value Type: TEXT

Conformance: This property MAY only be specified in a "VEVENT" calendar component.

Description: Transparency is the state that determines whether an event takes up time on a calendar, and with be reflected as "busy time", or is just "penciled in" to the calendar, and will not be reflected as "busy time". If the event is transparent, then other events can be scheduled in the same time period without there being a time period conflict. If not specified, the default state is OPAQUE.

No property parameters MAY be specified on this property.

Format Definition: The property is specified by the following notation:

```
transp      = "TRANSP" ":" [WSP] transvalue CRLF
transvalue = "OPAQUE"      ;Blocks or opaque on busy time searches.
              / "TRANSPARENT" ;Transparent on busy time searches.
              ;Default value is OPAQUE
```

Example: 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:OPAQUE
```

[4.8.3](#) Time Zone Component Properties

The following properties specify time zone information in calendar

components.

4.8.3.1 Time Zone Identifier

Property Name: TZID

Purpose: This property specifies the text value that uniquely identifies the VTIMZONE calendar component.

Value Type: TEXT

Conformance: This property MUST be specified in a "VTIMEZONE" calendar component.

Description: This is the key by which a time zone object is referenced by any iCalendar properties whose data type is either DATE-TIME or TIME. The presence of the SOLIDUS character (ASCII decimal 47) as a prefix, indicates that this TZID represents an unique ID in a globally defined time zone registry (when such registry is defined).

No property parameters MAY be specified on this property.

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

```
tzid          = "TZID" ":" [WSP] [tzidprefix] text CRLF
```

```
tzidprefix = "/"
```

Example: The following are examples of non-globally unique time zone identifiers:

```
TZID:America-New_York
```

```
TZID:/US-California-Los_Angeles
```

The following is an example of a globally unique time zone identifier:

```
TZID:/US-New_York-New_York
```

4.8.3.2 Time Zone Name

Property Name: TZNAME

Purpose: This property specifies the customary designation for a time zone description.

Value Type: TEXT

Internet Draft

C&S Core Object Specification

March 6, 1998

Conformance: This property MAY only be specified in the "VTIMEZONE" calendar component.

Description: This property may be specified in multiple languages; in order to provide for different national language requirements.

Only the "LANGUAGE" property parameter MAY be specified on this property.

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

tzname = "TZNAME" [tznparam] ":" [WSP] text CRLF

tznparam = ";" [WSP] languageparm

Example: The following are examples of this property:

TZNAME: EST

TZNAME;LANGUAGE=fr-CA:HNE

[4.8.3.3](#) Time Zone Offset From

Property Name: TZOFFSETFROM

Purpose: This property specifies the offset which is in use prior to this time zone observance.

Value Type: UTC-OFFSET

Conformance: This property MUST only be specified in a "VTIMEZONE" calendar component. A "VTIMEZONE" calendar component MUST include this property.

Description: This property specifies the offset which is in use prior to this time observance. It is used to calculate the absolute time at which the transition to a given observance takes place. This property MUST only be specified in a "VTIMEZONE" calendar component. A "VTIMEZONE" 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.

No property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

`tzoffsetfrom = "TZOFFSETFROM" ":" [WSP] utc-offset CRLF`

Example: The following are examples of this property:

`TZOFFSETFROM:-0500`

Dawson/Stenerson

71

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

`TZOFFSETFROM:+0530`

[4.8.3.4](#) Time Zone Offset To

Property Name: TZOFFSETTO

Purpose: This property specifies the offset which is in use in this time zone observance.

Value Type: UTC-OFFSET

Conformance: This property MAY only be specified in a "VTIMEZONE" calendar component. A "VTIMEZONE" calendar component MUST include this property.

Description: This property specifies the offset which is in use in this time zone observance. It is used to calculate the absolute time for the new observance. 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.

No property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

`tzoffsetto = "TZOFFSETTO" ":" [WSP] utc-offset CRLF`

Example: The following are examples of this property:

`TZOFFSETTO:-0500`

`TZOFFSETTO:+0530`

[4.8.3.5](#) Time Zone URL

Property Name: TZURL

Purpose: The TZURL provides a means for a VTIMEZONE component to point to a network location that can be used to retrieve an up-to-date version of itself.

Value Type: URI

Conformance: This property MAY only be specified in a "VTIMEZONE" calendar component.

Description: The TZURL provides a means for a VTIMEZONE component to point to a network location that can be used to retrieve an up-to-date version of itself. This provides a hook to handle changes government bodies impose upon timezone definitions. Retrieval of this resource results in an iCalendar object containing a single VTIMEZONE component and a METHOD property set to PUBLISH.

| | | |
|------------------|----|------------------------|
| Dawson/Stenerson | 72 | Expires September 1998 |
|------------------|----|------------------------|

| | | |
|----------------|-------------------------------|---------------|
| Internet Draft | C&S Core Object Specification | March 6, 1998 |
|----------------|-------------------------------|---------------|

No property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

tzurl = "TZURL" ":" [WSP] uri CRLF

Example: The following is an example of this property:

TZURL:http://timezones.r.us.net/tz/US-California-Los_Angeles

4.8.4 Relationship Component Properties

The following properties specify relationship information in calendar components.

4.8.4.1 Attendee

Property Name: ATTENDEE

Purpose: The property defines an attendee within a calendar component.

Value Type: CAL-ADDRESS

Conformance: This property MUST be specified in an iCalendar object that specifies a group scheduled calendar entity. This property MUST be specified in an iCalendar object that specifies the publication of a calendar user's busy time. This property is not specified in an

iCalendar object that specifies only a time zone definition or that defines calendar entities that are not group scheduled entities, but are entities only on a single user's calendar.

Description: The property is specified within the "VEVENT", "VTODO", "VJOURNAL" calendar components to specify participants, non-participants and the chair of a group scheduled calendar entity. The property is specified within the "VFREEBUSY" calendar component to specify the calendar user associated with the free or busy time. The property is specified within an "EMAIL" category of the "VALARM" calendar component to specify an email address that is to receive the email type of iCalendar alarm.

The property has the property parameter CN, for the common or displayable name associated with the calendar address, ROLE, for the intended role that the attendee will have in the calendar component; ATTSTAT, 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 groups that the attendee belongs to; DELEGATED-TO, to indicate the calendar users that the original request was delegated to; and DELEGATED-FROM, to indicate whom the request was delegated from; SENT-BY, to indicate whom is acting on behalf of the ATTENDEE and DIR, to indicate the URI that points to the directory information corresponding to the attendee. These property parameters MAY only be specified on an "ATTENDEE" property

Dawson/Stenerson

73

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

in either a "VEVENT", "VTODO" or "VJOURNAL" calendar component. They MUST not be specified in an "ATTENDEE" property in a "VFREEBUSY" or "VALARM" calendar component.

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.

Only the "TYPE", "CN", "ROLE", "ATTSTAT", "RSVP", "DELEGATED-TO", "DELEGATED-FROM", "SENT-BY" or "DIR" property specific parameters MAY be specified on this property. In addition, the "LANGUAGE" property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
attendee    = "ATTENDEE" *("(" [WSP] attparam) ":" [WSP]
```

cal-address CRLF

attparam = typeparam / cnparam / roleparam / statusparam / rsvpparm
/ memberparam / deletoparm / delefromparam
/ sentbyparam / dirparam

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

cnparam = "CN" "=" param-value

roleparam = "ROLE" "="
("CHAIR" ; Indicates CHAIR of the calendar entity
/ "REQ-PARTICIPANT" ; Indicates a participant whose
;participation is required
/ "OPT-PARTICIPANT" ;Indicates a participant whose
;participation is optional
/ "NON-PARTICIPANT" ;Indicates a participant who
;is copied for information purposes only
;Default is REQ-PARTICIPANT

statusparam = "ATTSTAT" "="
("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.
/ "IN-PROCESS" ;Indicates to-do is in the process of
; being completed.

Dawson/Stenerson

74

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

/ "DELEGATED" ; Indicates event or to-do delegated
; to another ATTENDEE
;Default is NEEDS-ACTION

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


```

memberparm = "MEMBER" "=" DQUOTE uri DQUOTE
              *(", " [WSP] DQUOTE uri DQUOTE)
; Indicates a group or mailing list
; MUST be a MAILTO URI as defined in [RFC 1738]
; The parameter value must be put in a quoted string.

deletoparm = "DELEGATED-TO" "=" DQUOTE uri DQUOTE
              *(", " [WSP] DQUOTE uri DQUOTE)
; Indicates who the request is delegated to
; MUST be a MAILTO URI as defined in [RFC 1738]
; The parameter value must be put in a quoted string.
delefromparm =
"DELEGATED-FROM" "=" DQUOTE uri DQUOTE
              *(", " [WSP] DQUOTE uri DQUOTE)
; Indicates who the request is delegated from
; MUST be a MAILTO URI as defined in [RFC 1738]
; The parameter value must be put in a quoted string.

sentbyparm = "SENT-BY" "=" DQUOTE uri DQUOTE
; Indicates a calendar user acting on behalf of the attendee.
; MUST be a MAILTO URI as defined in [RFC 1738]
; The parameter value must be put in a quoted string.
dirparm
    = "DIR" "=" DQUOTE uri DQUOTE

```

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

```

ORGANIZER:MAILTO:jsmith@host1.com
ATTENDEE;MEMBER="MAILTO:DEV-GROUP@host2.com":
    MAILTO:joecool@host2.com
ATTENDEE;DELEGATED-FROM="MAILTO:immud@host3.com":
    MAILTO:ildoit@host1.com

```

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

```

ORGANIZER:MAILTO:jsmith@host1.com
ATTENDEE;ROLE=REQ-PARTICIPANT;ATTSTAT=TENTATIVE;CN=Henry Cabot
    :MAILTO:hcabot@host2.com
ATTENDEE;ROLE=REQ-PARTICIPANT;DELEGATED-FROM="MAILTO:bob@host.com"
    ;ATTSTAT=ACCEPTED;CN=Jane Doe: MAILTO:jdoe@host1.com

```

The following is an example of this property with a URI to the directory information associated with the attendee:

```
ATTENDEE;CN=John Smith;DIR="ldap://host.com:6666/o=eDABC%
20Industries,c=3DUS??(cn=3DBJim%20Dolittle)":MAILTO:jimdo@
host1.com
```

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

```
ORGANIZER;CN=John Smith:MAILTO:jsmith@host.com
ATTENDEE;ROLE=REQ-PARTICIPANT;ATTSTAT=TENTATIVE;DELEGATED-FROM=
"MAILTO:iamboss@host2.com";CN=Henry Cabot:MAILTO:hcabot@
host2.com
ATTENDEE;ROLE=NON-PARTICIPANT;ATTSTAT=DELEGATED;DELEGATED-TO=
"MAILTO:hcabot@host2.com";CN=The Big Cheese:MAILTO:iamboss
@host2.com
ATTENDEE;ROLE=REQ-PARTICIPANT;ATTSTAT=ACCEPTED;CN=Jane Doe
:MAILTO:jdoe@host1.com
```

Example: The following is an example of this property's use when another calendar user is acting on behalf of the "Attendee":

```
ATTENDEE;SENT-BY=MAILTO:jan_doe@host1.com;CN=John Smith:MAILTO:
jsmith@host1.com
```

4.8.4.2 Contact

Property Name: CONTACT

Purpose: The property is used to represent contact information or alternately a reference to contact information associated with the calendar component.

Value Type: TEXT

Conformance: The property MAY only be specified in a "VEVENT", "VTOD0" and "VJOURNAL" calendar component in an iCalendar object.

Description: The property value consists of textual contact information. An alternative representation for the property value MAY also be specified that refers to a URI pointing to an alternate form, such as a vCard, for the contact information.

Only the "ALTREP" or "LANGUAGE" property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
contact      = "CONTACT" [contparam] ":" [WSP] text CRLF
contparam    = [";" [WSP] altrepparm] [";" [WSP] languageparm]
```

Example: The following is an example of this property referencing textual contact information:

Dawson/Stenerson

76

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
CONTACT:Jim Dolittle\, ABC Industries\, +1-919-555-1234
```

The following is an example of this property with an alternate representation of a LDAP URI to a directory entry containing the contact information:

```
CONTACT;ALTREP="ldap://host.com:6666/o=3DABC%20Industries\,
c=3DUS??(cn=3DBJim%20Dolittle)":Jim Dolittle\, ABC Industries\,
+1-919-555-1234
```

The following is an example of this property with an alternate representation of a MIME body part containing the contact information, such as a vCard embedded in a [MIME-DIR] content-type:

```
CONTACT;ALTREP="CID=<part3.msg970930T083000SILVER@host.com>":Jim
Dolittle\, ABC Industries\, +1-919-555-1234
```

The following is an example of this property referencing a network resource, such as a vCard object containing the contact information:

```
CONTACT;ALTREP="http://host.com/pdi/jdoe.vcf":Jim
Dolittle\, ABC Industries\, +1-919-555-1234
```

4.8.4.3 Organizer

Property Name: ORGANIZER

Purpose: The property defines the organizer for a calendar component.

Value Type: CAL-ADDRESS

Conformance: This property MUST be specified in an iCalendar object that specifies a group scheduled calendar entity. This property MUST be specified in an iCalendar object that specifies the publication of a calendar user's busy time. This property MUST not be specified in an iCalendar object that specifies only a time zone definition or that defines calendar entities that are not group scheduled entities, but are entities only on a single user's calendar.

Description: The property is specified within the "VEVENT", "VTODO",

"VJOURNAL calendar components to specify the organizer of a group scheduled calendar entity. The property is specified within the "VFREEBUSY" calendar component to specify the calendar user requesting the free or busy time.

The property has the property parameters CN, for specifying the common or display name associated with the "Organizer", DIR, for specifying a pointer to the directory information associated with the "Organizer", SENT-BY, for specifying another calendar user that is acting on behalf of the "Organizer". No other parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

Dawson/Stenerson 77 Expires September 1998

Internet Draft C&S Core Object Specification March 6, 1998

```
organizer = "ORGANIZER" [orgparam] ":" [WSP]
           cal-address CRLF

orgparam  = [";" [WSP] "CN" "=" param-value]
           [";" [WSP] "DIR" "=" DQUOTE uri DQUOTE]
           proxyparam

proxyparam = (";" [WSP] "SENT-BY" "=" DQUOTE cal-address DQUOTE)
```

Example: The following is an example of this property:

```
ORGANIZER;CN=John Smith:MAILTO:jsmith@host1.com
```

The following is an example of this property with a pointer to the directory information associated with the organizer:

```
ORGANIZER;CN=JohnSmith;DIR="ldap://host.com:6666/o=3DDC%20Associ
ates,c=3DUS??(cn=3DJohn%20Smith)":MAILTO:jsmith@host1.com
```

The following is an example of this property used by another calendar user who is acting on behalf of the organizer, with responses intended to be sent back to the organizer, not the other calendar user:

```
ORGANIZER;SENT-BY="MAILTO:jane_doe@host.com":
MAILTO:jsmith@host1.com
```

[4.8.4.4](#) Recurrence ID

Property Name: RECURRENCE-ID

Purpose: This property is used in conjunction with the "UID" and "SEQUENCE" property to identify a specific instance of a recurring "VEVENT", "VTODO" or "VJOURNAL" calendar component. The property value is the effective value of the "DTSTART" property of the recurrence instance.

Value Type: The default value type for this property is DATE-TIME. The time format MAY BE any of the valid forms defined for a DATE-TIME value type. See DATE-TIME value type definition for specific interpretations of the various forms. The value type MAY be reset to DATE.

Conformance: This property MAY be specified in an iCalendar object containing a recurring calendar component.

Description: The full range of calendar components specified by a recurrence set is referenced by referring to just the "UID" property value corresponding to the calendar component. The "RECURRENCE-ID" property allows the reference to an individual instance within the recurrence set.

If the value of the "DTSTART" property is a DATE type value, then the value MUST be the calendar date for the recurrence instance.

Dawson/Stenerson

78

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

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.

The "RECURRENCE-ID" property is used in conjunction with the "UID" and "SEQUENCE" property to identify a particular instance of a recurring event, to-do or journal. For a given pair of "UID" and "SEQUENCE" property values, the "RECURRENCE-ID" value for a recurrence instance is fixed. When the definition of the recurrence set for a calendar component changes, and hence the "SEQUENCE" property value changes, the "RECURRENCE-ID" for a given recurrence instance might also change. The "RANGE" parameter is used to specify the effective range of recurrence instances from the instance specified by the "RECURRENCE-ID" property value. The default value for the range parameter is the single recurrence instance only. The value can also be "THISANDPRIOR" to indicate a range defined by the given recurrence instance and all prior instances or the value can be "THISANDFUTURE" to indicate a range defined by the given recurrence instance and all subsequent instances.

Only the "VALUE" property parameter and the property-specific "RANGE" parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
recurid    = "RECURRENCE-ID" *(";" [WSP] parameter)
              [;" [WSP] rangeparm] ":" [WSP] [date-time / date]
              ;Value must match value type.
```

```
recurid    = "RECURRENCE-ID" [ridparam] [rangeparm] ":" [WSP]
              ridval CRLF
```

```
ridparam   = ";" [WSP] "VALUE" "=" ("DATE-TIME" / "DATE" /
              / "PERIOD")
```

```
rangeparm  = ";" [WSP] "RANGE" "=" ("THISANDPRIOR"
              / "THISANDFUTURE")
```

```
ridval     = date-time / date / period
              ;Value MUST match value type
```

Example: The following are examples of this property:

```
RECURRENCE-ID;VALUE=DATE:19960401
```

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

[4.8.4.5](#) Related To

Property Name: RELATED-TO

Purpose: The property is used to represent a relationship or reference between one calendar component and another.

Dawson/Stenerson 79 Expires September 1998

Internet Draft C&S Core Object Specification March 6, 1998

Value Type: TEXT

Conformance: The property MAY only be specified once in the "VEVENT", "VTOD" and "VJOURNAL" calendar components.

Description: The property value consists of the persistent, globally unique identifier of another calendar component. This value would be represented in a calendar component by the "UID" property.

By default, the property value points to another calendar component that has a PARENT relationship to the referencing object. The

Changes to a calendar component referenced by this property MAY have an implied impact to 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. Similarly, if a PARENT calendar component is canceled or deleted, then there is an implied impact to the related CHILD calendar components. 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.

Format Definition: The property is defined by the following notation:

The following is an example of this property:

RELATED-TO:<19960401-080045-4000F192713-0052@host1.com>

Property Name: URL

Expires September 1998

March 6, 1998

Purpose: This property defines a Uniform Resource Locator (URL)

associated with the iCalendar object.

Value Type: URI

Conformance: This property MAY be specified once in the "VEVENT", "VTOD0", "VJOURNAL" and "VFREEBUSY" calendar components.

Description: This property may be used in a calendar component to convey a location where a more dynamic rendition of the calendar information associated with the calendar component can be found. This memo does not attempt to standardize the form of the URI, nor the format of the resource pointed to by the property value. If the

No property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
url          = "URL" ":" [WSP] uri CRLF
```

Example: The following is an example of this property:

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

4.8.4.7 Unique Identifier

Property Name: UID

Purpose: This property defines the persistent, globally unique identifier for the calendar component.

Value Type: TEXT

Conformance: The property MUST be specified in the "VEVENT", "VTOD0", "VJOURNAL" and "VFREEBUSY" calendar components.

Description: The UID itself MUST be a globally unique identifier. The generator of the identifier MUST guarantee that the identifier is unique. There are several algorithms that can be used to accomplish this. The identifier is RECOMMENDED to be the identical syntax to the [\[RFC 822\]](#) addr-spec. A good method to assure uniqueness is to put the domain name or a domain literal IP address of the host on which the identifier was created on the right hand side of the "@", and on the left hand side, put a combination of the current calendar date and time of day (i.e., formatted in as a DATE-TIME value) along with some other currently unique (perhaps sequential) identifier available on the system (for example, a process id number). Using a date/time value on the left hand side and a domain name or domain literal on the right hand side makes it possible to guarantee uniqueness since no two hosts should be using the same domain name or IP address at the same time. Though other algorithms will work, it is RECOMMENDED that the right hand side contain some domain identifier (either of the host itself or otherwise) such that the generator of the message

Internet Draft

C&S Core Object Specification

March 6, 1998

identifier can guarantee the uniqueness of the left hand side within the scope of that domain.

This is the method for correlating scheduling messages with the referenced "VEVENT", "VTOD0", or "VJOURNAL" calendar component.

The full range of calendar components specified by a recurrence set is referenced by referring to just the "UID" property value corresponding to the calendar component. The "RECURRENCE-ID" property allows the reference to an individual instance within the recurrence set.

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 "VEVENT", "VTOD0" and "VJOURNAL" calendar components to assure interoperability with other group scheduling applications. This identifier is created by the calendar system that generates an iCalendar object.

Implementations MUST be able to receive and persist values of at least 255 characters for this property.

No property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

uid = "UID" ":" [WSP] text CRLF

Example: The following is an example of this property:

UID:19960401T080045Z-4000F192713-0052@host1.com

4.8.5 Recurrence Component Properties

The following properties specify recurrence information in calendar components.

4.8.5.1 Exception Date/Times

Property Name: EXDATE

Purpose: This property defines the list of date/time exceptions for a recurring calendar component.

Value Type: The default value type for this property is DATE-TIME. The value type MAY be reset to DATE.

Conformance: This property MAY be specified in an iCalendar object that includes a recurring calendar component.

Description: The exception dates, if specified, is used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is

Dawson/Stenerson

82

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

generated by considering the initial "DTSTART" property along with the "RRULE", "RDATE", "EXDATE" and "EXRULE" properties contained within the iCalendar object. The "DTSTART" property 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" properties, only one recurrence is considered. Duplicate instances are ignored.

The "EXDATE" property MAY be used to exclude the value specified in "DTSTART". However, in such cases the original "DTSTART" date MUST still be maintained by the calendaring and scheduling system because the original "DTSTART" value has inherent usage dependencies by other properties such as the "RECURRENCE-ID".

Only the "VALUE" and "TZID" property parameters MAY be specified on this property. The "TZID" property parameter MAY only be specified when the value type is DATE-TIME.

Format Definition: The property is defined by the following notation:

exdate = "EXDATE" [exdtparam] ":" [WSP] exdtval CRLF

exdtparam = [";" [WSP] "VALUE" "=" "DATE-TIME"]
[";" [WSP] "TZID" "=" paramtext]

exdtparam =/ ";" [WSP] "VALUE" "=" "DATE"

exdtval = date-time / date
;Value MUST match value type

Example: The following is an example of this property:

EXDATE:19960402T010000Z,19960403T010000Z,19960404T010000Z

4.8.5.2 Exception Rule

Property Name: EXRULE

Purpose: This property defines a rule or repeating pattern for an exception to a recurrence set.

Value Type: RECUR

Conformance: This property MAY only be specified in the "VEVENT", "VTODO" or "VJOURNAL" calendar components.

Dawson/Stenerson

83

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Description: The exception rule, if specified, is used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is generated by considering the initial "DTSTART" property 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" properties, only one recurrence is considered. Duplicate instances are ignored.

The "EXRULE" property MAY be used to exclude the value specified in "DTSTART". However, in such cases the original "DTSTART" date MUST still be maintained by the calendaring and scheduling system because the original "DTSTART" value has inherent usage dependencies by other properties such as the "RECURRENCE-ID".

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
exrule      = "EXRULE" ":" [WSP] recur CRLF
```

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

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

Except daily for 10 occurrences:

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

Except yearly in June and July for 8 occurrences:

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

The data type for this property is RECUR.

4.8.5.3 Recurrence Date/Times

Property Name: RDATE

Purpose: This property defines the list of date/times for a recurrence set.

Dawson/Stenerson

84

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Value Type: The default value type for this property is DATE-TIME. The value type MAY be reset to DATE or PERIOD.

Conformance: The property MAY only appear within the "VEVENT", "VTOD", "VJOURNAL" or "VTIMEZONE" calendar components.

Description: 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 recurrence dates, if specified, is used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is generated by considering the initial "DTSTART" property along with the "RRULE",

"RDATE", "EXDATE" and "EXRULE" properties contained within the iCalendar object. The "DTSTART" property 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" properties, only one recurrence is considered. Duplicate instances are ignored.

Only the "VALUE" and "TZID" property parameters MAY be specified on this property. The "TZID" property parameter MAY only be specified when the value type is DATE-TIME.

Format Definition: The property is defined by the following notation:

```

rdate      = "RDATE" [rdtparam] ":" [WSP] rdtval
              *("," [WSP] rdtval) CRLF

rdtparam   = [";" [WSP] "VALUE" "=" "DATE-TIME"]
              [";" [WSP] "TZID" "=" paramtext]

rdtparam   =/ ";" [WSP] "VALUE" "=" "DATE"

rdtparam   =/ ";" [WSP] "VALUE" "=" "PERIOD"

rdtval     = date-time / date / period
;Value MUST match value type

```

Example: The following are examples of this property:

```
RDATE:19970714T123000Z
```

```
RDATE;TZID=EST:19970714T083000
```

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

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

4.8.5.4 Recurrence Rule

Property Name: RRULE

Purpose: This property defines a rule or repeating pattern for a recurring events, to-dos, or time zone definitions.

Value Type: RECUR

Conformance: This property MAY be specified one or more times in recurring "VEVENT", "VTODO" and "VJOURNAL" calendar components. It MAY also be specified once in the "VTIMEZONE" calendar component.

Description: The recurring rule, if specified, is used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is generated by considering the initial "DTSTART" property along with the "RRULE", "RDATE", "EXDATE" and "EXRULE" properties contained within the iCalendar object. The "DTSTART" property 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" properties, only one recurrence is considered. Duplicate instances are ignored.

The "DTSTART" and "DTEND" property pair or "DTSTART" and "DURATION" property 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 "VTIMEZONE" calendar components MUST be included. For detail on the usage of the "VTIMEZONE" calendar component, see the "VTIMEZONE" 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.

No property parameters MAY be specified on this property.

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

```
rrule      = "RRULE" ":" [WSP] recur CRLF
```

Example: All examples assume the Eastern United States time zone.

Daily for 10 occurrences:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=DAILY;COUNT=10
```

```
==> (1997 9AM EDT)September 2-11
```

Daily until 12/24/97:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=DAILY;UNTIL=19971224T000000Z
```

```
==> (1997 9AM EDT)September 2-30;October 1-25
    (1997 9AM EST)October 26-31;November 1-30;December 1-23
```

Every other day - forever:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=DAILY;INTERVAL=2
==> (1997 9AM EDT)September 2, 4, 6, 8... 24, 26, 28, 30;
    October 2, 4, 6... 20, 22, 24
    (1997 9AM EST)October 26, 28, 30; November 1, 3, 5, 7... 25, 27, 29;
    Dec 1, 3, ...
```

Every 10 days, 5 occurrences:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=DAILY;INTERVAL=10;COUNT=5
```

```
==> (1997 9AM EDT)September 2, 12, 22; October 2, 12
```

Everyday in January, for 3 years:

```
DTSTART;TZID=America-New York:19980101T090000
RRULE;TZID=America-New York:FREQ=YEARLY;UNTIL:20000131T090000;
    BYMONTH=1;BYDAY=SU, MO, TU, WE, TH, FR, SA
or
RRULE;TZID=America-New York:FREQ=DAILY;UNTIL:20000131T090000;
    BYMONTH=1
```

==> (1998 9AM EDT)January 1-31
(1999 9AM EDT)January 1-31
(2000 9AM EDT)January 1-31

Weekly for 10 occurrences

DTSTART;TZID=America-New York:19970902T090000

Dawson/Stenerson

87

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

RRULE:FREQ=WEEKLY;COUNT=10

==> (1997 9AM EDT)September 2,9,16,23,30;October 7,14,21
(1997 9AM EST)October 28;November 4

Weekly until 12/24/94

DTSTART;TZID=America-New York:19970902T090000

RRULE:FREQ=WEEKLY;UNTIL=19971224T000000Z

==> (1997 9AM EDT)September 2,9,16,23,30;October 7,14,21
(1997 9AM EST)October 28;November 4,11,18,25;
December 2,9,16,23

Every other week - forever:

DTSTART;TZID=America-New York:19970902T090000

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

==> (1997 9AM EDT)September 2,16,30;October 14
(1997 9AM EST)October 28;November 11,25;December 9,23
(1998 9AM EST)January 6,20;February

...

Weekly on Tuesday and Thursday for 5 weeks:

DTSTART;TZID=America-New York:19970902T090000

RRULE;TZID=America-New York:FREQ=WEEKLY;UNTIL=19971007T000000;

WKST=SU;BYDAY=TU,TH

or

RRULE:FREQ=WEEKLY;COUNT=10;WKST=SU;BYDAY=TU,TH

==> (1997 9AM EDT)September 2,4,9,11,16,18,23,25,30;October 2

Every other week on Monday, Wednesday and Friday until 12/24/97, but
starting on Tuesday, 9/2/97:

DTSTART;TZID=America-New York:19970902T090000
RRULE;TZID=America-New York:FREQ=WEEKLY;INTERVAL=2;UNTIL=
19971224T000000;WKST=SU;BYDAY=MO,WE,FR
==> (1997 9AM EDT)September 2,3,5,15,17,19,29;October 1,3,13,15,17
(1997 9AM EST)October 27,29,31;November 10,12,14,24,26,28;
December 8,10,12,22

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

DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=WEEKLY;INTERVAL=2;COUNT=8;WKST=SU;BYDAY=TU,TH

==> (1997 9AM EDT)September 2,4,16,18,30;October 2,14,16

Monthly on the 1st Friday for ten occurrences:

| | | |
|------------------|----|------------------------|
| Dawson/Stenerson | 88 | Expires September 1998 |
|------------------|----|------------------------|

| | | |
|----------------|-------------------------------|---------------|
| Internet Draft | C&S Core Object Specification | March 6, 1998 |
|----------------|-------------------------------|---------------|

DTSTART;TZID=America-New York:19970905T090000
RRULE:FREQ=MONTHLY;COUNT=10;BYDAY=1FR

==> (1997 9AM EDT)September 5;October 3
(1997 9AM EST)November 7;Dec 5
(1998 9AM EST)January 2;February 6;March 6;April 3
(1998 9AM EDT)MAY 1;June 5

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

DTSTART;TZID=America-New York:19970905T090000
RRULE:FREQ=MONTHLY;UNTIL=19971224T000000Z;BYDAY=1FR

==> (1997 9AM EDT)September 5;October 3
(1997 9AM EST)November 7;December 5

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

DTSTART;TZID=America-New York:19970907T090000
RRULE:FREQ=MONTHLY;INTERVAL=2;COUNT=10;BYDAY=1SU, -1SU

==> (1997 9AM EDT)September 7,28
(1997 9AM EST)November 2,30
(1998 9AM EST)January 4,25;March 1,29
(1998 9AM EDT)MAY 3,31

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

DTSTART;TZID=America-New York:19970922T090000
RRULE:FREQ=MONTHLY;COUNT=6;BYDAY=-2MO

==> (1997 9AM EDT)September 22;October 20
(1997 9AM EST)November 17;December 22
(1998 9AM EST)January 19;February 16

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

DTSTART;TZID=America-New York:19970928T090000
RRULE:FREQ=MONTHLY;BYMONTHDAY=-3

==> (1997 9AM EDT)September 28
(1997 9AM EST)October 29;November 28;December 29
(1998 9AM EST)January 29;February 26
...

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

DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=MONTHLY;COUNT=10;BYMONTHDAY=2,15

==> (1997 9AM EDT)September 2,15;October 2,15
(1997 9AM EST)November 2,15;December 2,15
(1998 9AM EST)January 2,15

Dawson/Stenerson

89

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

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

DTSTART;TZID=America-New York:19970930T090000
RRULE:FREQ=MONTHLY;COUNT=10;BYMONTHDAY=1, -1

==> (1997 9AM EDT)September 30;October 1
(1997 9AM EST)October 31;November 1,30;December 1,31
(1998 9AM EST)January 1,31;February 1

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

DTSTART;TZID=America-New York:19970910T090000
RRULE:FREQ=MONTHLY;INTERVAL=18;COUNT=10;BYMONTHDAY=10,11,12,13,14,15

==> (1997 9AM EDT)September 10,11,12,13,14,15
(1999 9AM EST)March 10,11,12,13

Every Tuesday, every other month:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=MONTHLY;INTERVAL=2;BYDAY=TU
```

```
==> (1997 9AM EDT)September 2,9,16,23,30
      (1997 9AM EST)November 4,11,18,25
      (1998 9AM EST)January 6,13,20,27;March 3,10,17,24,31
      ...
```

Yearly in June and July for 10 occurrences:

```
DTSTART;TZID=America-New York:19970610T090000
RRULE:FREQ=YEARLY;COUNT=10;BYMONTH=6,7
```

```
==> (1997 9AM EDT)June 10;July 10
      (1998 9AM EDT)June 10;July 10
      (1999 9AM EDT)June 10;July 10
      (2000 9AM EDT)June 10;July 10
      (2001 9AM EDT)June 10;July 10
```

Note: Since none of the BYDAY, BYMONTHDAY or BYYEARDAY components are specified, the day is gotten from DTSTART

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

```
DTSTART;TZID=America-New York:19970310T090000
RRULE:FREQ=YEARLY;INTERVAL=2;COUNT=10;BYMONTH=1,2,3
```

```
==> (1997 9AM EST)March 10
      (1999 9AM EST)January 10;February 10;March 10
      (2001 9AM EST)January 10;February 10;March 10
      (2003 9AM EST)January 10;February 10;March 10
```

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

Dawson/Stenerson

90

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

```
DTSTART;TZID=America-New York:19970101T090000
RRULE:FREQ=YEARLY;INTERVAL=3;COUNT=10;BYYEARDAY=1,100,200
```

```
==> (1997 9AM EST)January 1
      (1997 9AM EDT)April 10;July 19
      (2000 9AM EST)January 1
      (2000 9AM EDT)April 9;July 18
      (2003 9AM EST)January 1
```

(2003 9AM EDT)April 10;July 19
(2006 9AM EST)January 1

Every 20th Monday of the year, forever:

DTSTART;TZID=America-New York:19970519T090000
RRULE:FREQ=YEARLY;BYDAY=20MO

==> (1997 9AM EDT)MAY 19
(1998 9AM EDT)MAY 18
(1999 9AM EDT)MAY 17

...

Monday of Week No. 20, forever:

DTSTART;TZID=America-New York:19970512T090000
RRULE:FREQ=YEARLY;BYWEEKNO=20;BYDAY=MO

==> (1997 9AM EDT)MAY 12
(1998 9AM EDT)MAY 11
(1999 9AM EDT)MAY 17

...

Every Thursday in March, forever:

DTSTART;TZID=America-New York:19970313T090000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=TH

==> (1997 9AM EST)March 13,20,27
(1998 9AM EST)March 5,12,19,26
(1999 9AM EST)March 4,11,18,25

...

Every Thursday, but only during summer months, forever:

DTSTART;TZID=America-New York:19970605T090000
RRULE:FREQ=YEARLY;BYDAY=TH;BYMONTH=6,7,8

==> (1997 9AM EDT)June 5,12,19,26;July 3,10,17,24,31;
August 7,14,21,28
(1998 9AM EDT)June 4,11,18,25;July 2,9,16,23,30;
August 6,13,20,27
(1999 9AM EDT)June 3,10,17,24;July 1,8,15,22,29;
August 5,12,19,26

...

Every Friday the 13th, forever:

```
DTSTART;TZID=America-New York:19970902T090000
EXDATE;TZID=America-New York:19970902T090000
RRULE:FREQ=MONTHLY;BYDAY=FR;BYMONTHDAY=13
```

```
==> (1998 9AM EST)February 13;March 13;November 13
      (1999 9AM EDT)August 13
      (2000 9AM EDT)October 13
...

```

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

```
DTSTART;TZID=America-New York:19970913T090000
RRULE:FREQ=MONTHLY;BYDAY=SA;BYMONTHDAY=7,8,9,10,11,12,13
```

```
==> (1997 9AM EDT)September 13;October 11
      (1997 9AM EST)November 8;December 13
      (1998 9AM EST)January 10;February 7;March 7
      (1998 9AM EDT)April 11;MAY 9;June 13...
...

```

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

```
DTSTART;TZID=America-New York:19961105T090000
RRULE:FREQ=YEARLY;INTERVAL=4;BYMONTH=11;BYDAY=TU;BYMONTHDAY=2,3,4,
5,6,7,8
```

```
==> (1996 9AM EST)November 5
      (2000 9AM EST)November 7
      (2004 9AM EST)November 2
...

```

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

```
DTSTART;TZID=America-New York:19970904T090000
RRULE:FREQ=MONTHLY;COUNT=3;BYDAY=TU,WE,TH;BYSETPOS=3
```

```
==> (1997 9AM EDT)September 4;October 7
      (1997 9AM EST)November 6

```

The 2nd to last weekday of the month:

```
DTSTART;TZID=America-New York:19970929T090000
RRULE:FREQ=MONTHLY;BYDAY=M0,TU,WE,TH,FR;BYSETPOS=-2
```

```
==> (1997 9AM EDT)September 29

```

(1997 9AM EST)October 30;November 27;December 30
(1998 9AM EST)January 29;February 26;March 30

...

Dawson/Stenerson

92

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Every 3 hours from 9AM to 5PM on a specific day:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE;TZID=America-New York:FREQ=HOURLY;INTERVAL=3;
UNTIL=19970902T170000
```

==> (September 2, 1997 EDT)09:00,12:00,15:00

Every 15 minutes for 6 occurrences:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=MINUTELY;INTERVAL=15;COUNT=6
```

==> (September 2, 1997 EDT)09:00,09:15,09:30,09:45,10:00,10:15

Every hour and a half for 4 occurrences:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=MINUTELY;INTERVAL=90;COUNT=4
```

==> (September 2, 1997 EDT)09:00,10:30;12:00;13:30

Every 20 minutes from 9AM to 4:40PM every day:

```
DTSTART;TZID=America-New York:19970902T090000
RRULE:FREQ=DAILY;BYHOUR=9,10,11,12,13,14,15,16;BYMINUTE=0,20,40
or
RRULE:FREQ=MINUTELY;INTERVAL=20;BYHOUR=9,10,11,12,13,14,15,16
```

==> (September 2, 1997 EDT)9:00,9:20,9:40,10:00,10:20,
... 16:00,16:20,16:40

(September 3 1997 EDT)9:00,9:20,9:40,10:00,10:20,
...16:00,16:20,16:40

...

An example where the days generated makes a difference because of WKST:

```
DTSTART;TZID=America-New York:19970805T090000
RRULE:FREQ=WEEKLY;INTERVAL=2;COUNT=4;BYDAY=TU,SU;WKST=MO
```

4.8.6.2 Repeat Count

Property Name: REPEAT

Purpose: This property defines the number of time the alarm should be repeated, after the initial trigger.

Value Type: INTEGER

Conformance: This property MAY only be specified in a "VALARM" calendar component.

Description: If the alarm triggers more than once, then this property MUST be specified; along with the "DURATION" property.

No property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
repeatcnt = "REPEAT" ":" [WSP] integer CRLF
;Default is "0", zero.
```

Dawson/Stenerson

94

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Example: The following is an example of this property for an alarm that repeats 4 additional times with a 5 minute delay after the initial triggering of the alarm.:

```
REPEAT:4
DURATION:PT5M
```

4.8.6.3 Trigger

Property Name: TRIGGER

Purpose: This property specifies when an alarm will trigger.

Value Type: The default value type is DURATION. The value type MAY BE reset to a DATE-TIME value type.

Conformance: This property MUST be specified within the "VALARM" calendar component.

Description: Within the "VALARM" calendar component, this property defines when the alarm will trigger. The default value type is DURATION, specifying a relative time for the trigger of the alarm. The default duration is relative to the start of an event or to-do

that the alarm is associated with. The duration can be explicitly set to trigger from either the end or the start of the associated event or to-do with the "RELATED" parameter. A value of START will set the alarm to trigger off the start of the associated event or to-do. A value of END will set the alarm to trigger off the end of the associated event or to-do.

Either a positive or negative duration may be specified for the "TRIGGER" property. An alarm with a positive duration is triggered before the associated start or end of the event or to-do. An alarm with a negative duration is triggered after the associated start or end of the event or to-do.

This parameter is not valid if the value type of the property is reset to DATE-TIME, in order to specify an absolute trigger for the alarm. If a value type of DATE-TIME is specified, then the value MUST be specified in the UTC time format. If an absolute trigger is specified on an alarm for a recurring event or to-do, then the alarm will only trigger for the specified absolute date/time, along with any specified repeating instances.

If the trigger is set relative to START, then the "DTSTART" property must be present in the associated "VEVENT" or "VTODO" calendar component. If an alarm is specified for an event with the trigger set relative to the END, then the "DTEND" property or the "DSTART" and "DURATION" properties must be present in the associated "VEVENT" calendar component. If the alarm is specified for a to-do with a trigger set relative to the END, then either the "DUE" property or the "DSTART" and "DURATION" properties must be present in the associated "VTODO" calendar component.

Dawson/Stenerson

95

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Alarms specified in an event or to-do that is defined solely in terms of a DATE value type will be triggered relative to midnight on the day prior to the specified date. For example, if "DTSTART:19980205", then the a duration trigger will be relative to 19980204T240000Z.

Only the "VALUE" and "TZID" property parameters and property-specific "RELATED" parameter MAY be specified on this property. The "TZID" property parameter MAY only be specified when the value type is DATE-TIME.

Format Definition: The property is defined by the following notation:

trigger = "TRIGGER" (trigrel / trigabs)

```

trigrel    = ([";" [WSP] "VALUE" "=" "DURATION"]
              ";" [WSP] "RELATED" "=" ("START" / "END"))
              ":" [WSP] (["+"] / "-") duration

trigabs     = ([";" [WSP] "VALUE" "=" "DATE-TIME"
              ":" [WSP] date-time)

```

Example: A trigger set 15 minutes relative to the start of the event or to-do.

```
TRIGGER:P15M
```

A trigger set 5 minutes relative to the end of the event or to-do.

```
TRIGGER;RELATED=END:P5M
```

A trigger set to an absolute date/time.

```
TRIGGER;VALUE=DATE-TIME:19980101T050000Z
```

4.8.7 Change Management Component Properties

The following properties specify change management information in calendar components.

4.8.7.1 Date/Time Created

Property Name: CREATED

Purpose: This property specifies the date and time that the calendar information was created in the calendar user agent.

Value Type: DATE-TIME

Conformance: The property MAY be specified in any of the calendar components. The property MAY only be specified once.

Description: The date and time is a UTC value.

No property parameters MAY be specified on this property.

Dawson/Stenerson

96

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

Format Definition: The property is defined by the following notation:

```
created    = "CREATED" ":" [WSP] date-time CRLF
```

Example: The following is an example of this property:

```
CREATED:19960329T133000Z
```

4.8.7.2 Date/Time Stamp

Property Name: DTSTAMP

Purpose: The property indicates the date/time that the instance of the iCalendar object was created.

Value Type: DATE-TIME

Conformance: This property MUST be included in the "VEVENT", "VTOD0", "VJOURNAL" and "VFREEBUSY" calendar components.

Description: The value must be specified in the UTC time format.

This property is also useful to protocols such as [\[IMIP\]](#) that have inherent latency issues with the delivery of content. This property will assist in the proper sequencing of messages containing iCalendar objects.

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.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
dtstamp    = "DTSTAMP" ":" [WSP] date-time CRLF
```

Example:

```
DTSTAMP:19971210T080000Z
```

4.8.7.3 Last Modified

Property Name: LAST-MODIFIED

Purpose: The property specifies the date and time that the information associated with the calendar component was last revised.

Value Type: DATE-TIME

Conformance: This property MAY be specified in the "EVENT", "VTOD0", "VJOURNAL" or "VFREEBUSY" calendar components.

Description: The property value MUST be specified in the UTC time format.

No property parameters MAY be specified on this property.

Format Definition: The property is defined by the following notation:

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

Example: The following is are examples of this property:

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

[4.8.7.4](#) Sequence Number

Property Name: SEQUENCE

Purpose: This property defines the revision sequence of the calendar component used in a request.

Value Type: INTEGER

Conformance: The property MAY only be specified in a "VEVENT", "VTODO", "VJOURNAL" or "VFREEBUSY" calendar component.

Description: This property is needed to properly handle the receipt and processing of a sequence of 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 incremented each time the ORGANIZER or OWNER issues a revision to the request. The sequence number MUST be monotonically incremented when one of the following properties in a calendar component is changed:

- . "DTSTART"
- . "DTEND"
- . "DUE"
- . "RDATE"
- . "RRULE"
- . "EXDATE"
- . "EXRULE"

No property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
seq         = "SEQUENCE" ":" [WSP] integer CRLF
```

;Default is "0".

Example: The following is an example of this property:

SEQUENCE:1

Dawson/Stenerson

98

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

4.8.8 Miscellaneous Component Properties

The following properties specify information about a number of miscellaneous features of calendar components.

4.8.8.1 Non-standard Properties

Property Name: Any property name with a "X-" prefix

Purpose: This class of property provides a framework for defining non-standard properties.

Value Type: TEXT

Conformance: This property MAY be specified in any calendar component.

Description: 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 memo. 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.

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.

Any of the general property parameters MAY be specified on this

property.

Format Definition: The property is defined by the following notation:

```
extension = "X-" [vendorid] "-" text *(";" [WSP] param) ":"  
          [WSP] value-list CRLF
```

; Lines longer than 75 characters should be folded

```
vendorid  = 3*char          ;Vendor identification prefix text
```

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

X-ABC-MMSUBJ;TYPE=wave; VALUE=URI: <http://load.noise.org/mysubj.wav>

Dawson/Stenerson

99

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

4.8.8.2 Request Status

Property Name: REQUEST-STATUS

Purpose: This property defines the status code returned for a scheduling request.

Value Type: TEXT

Conformance: The property MAY be specified in a "VEVENT", "VTODO", "VJOURNAL" or "VFREEBUSY" calendar component.

Description: 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 short return status is a PERIOD character (ASCII decimal 46) separated 3-tuple of integers. For example, "3.1.1". The successive levels of integers provide for a successive level of status code granularity.

The following are initial classes for the return status code. Individual iCalendar object methods will define specific return status codes for these classes. In addition, other classes for the

return status code may be defined using the registration process defined later in this memo.

| Short Return Status Code | Longer Return Status Description |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.xx | Preliminary success. This class of status of status code indicates that the request has been initially processed but that completion is pending. |
| 2.xx | 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. |
| 3.xx | 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. |
| 4.xx | Scheduling Error. This class of status code indicates that the request was not successful. |

Dawson/Stenerson

100

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

| | |
|--|------------------------------------------------------------------------------------------------------------------------|
| | Some sort of error occurred within the calendaring and scheduling service, not directly related to the request itself. |
|--|------------------------------------------------------------------------------------------------------------------------|

Only the "LANGUAGE" property parameter MAY be specified on this property.

Format Definition: The property is defined by the following notation:

```
rstatus      = "REQUEST-STATUS" [rstatparam] ":" [WSP]
               statcode ";" [WSP] statdesc [";" [WSP] extdata]
```

```
rstatparam = ";" [WSP] languageparm
```

```
statcode    = 1*DIGIT *("." 1*DIGIT)
;Hierarchical, numeric return status code
```

statdesc = text
;Textual status description

extdata = text
;Textual exception data. For example, the offending property
;name and value or complete property line.

Example: The following are some possible examples of this property
(Note: The BACKSLASH character escapement of separator characters in
the property because it has a text value type):

REQUEST-STATUS:2.0;Success

REQUEST-STATUS:3.1;Invalid property value;DTSTART:96-Apr-01

REQUEST-STATUS:2.8; Success\, repeating event ignored. Scheduled
as a single event.;RRULE:FREQ=WEEKLY\;INTERVAL=2

REQUEST-STATUS:4.1;Event conflict. Date/time is busy.

REQUEST-STATUS:3.7;Invalid calendar user;ATTENDEE:
MAILTO:jsmith@host.com

5 iCalendar Object Examples

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

The following example specifies a three day conference that begins at
8:00 AM EDT, September 18, 1996 and end at 6:00 PM EDT, September 20,
1996.

```
BEGIN:VCALENDAR
PRODID:-//xyz Corp//NONSGML PDA Calendar Verson 1.0//EN
VERSION:2.0
BEGIN:VEVENT
```

Dawson/Stenerson 101 Expires September 1998

Internet Draft C&S Core Object Specification March 6, 1998

```
DTSTAMP:19960704T120000Z
UID:uid1@host.com
DTSTART:19960918T143000Z
DTEND:19960920T220000Z
STATUS:CONFIRMED
CATEGORIES:CONFERENCE
SUMMARY:Networld+Interop Conference
DESCRIPTION:Networld+Interop Conference
```


and Exhibit\nAtlanta World Congress Center\
Atlanta, Georgia
END:VEVENT
END:VCALENDAR

The following example specifies a group scheduled meeting that begin at 8:30 AM EST on March 12, 1998 and end at 9:30 AM EST on March 12, 1998. The "Organizer" has scheduled the meeting with one or more calendar users in a group. A time zone specification for Eastern United States has been specified.

```
BEGIN:VCALENDAR
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VTIMEZONE
TZID:Eastern US
BEGIN:STANDARD
DTSTART:19981025T020000
RDATE:19981025T020000
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19990404T020000
RRULE:19990404T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTAMP:19980309T231000Z
UID:guid-1.host1.com
ORGANIZER;ROLE=CHAIR:mrbig@host.com
ATTENDEE;RSVP=TRUE;ROLE=REQ-PARTICIPANT;TYPE=GROUP:
  MAILTO:employee-A@host.com
DESCRIPTION:Project XYZ Review Meeting
CATEGORIES:MEETING
CLASS:PUBLIC
CREATED:19980309T130000Z
SUMMARY:XYZ Project Review
DTSTART;TZID=EST:19980312T083000
DTEND;TZID=EST:19980312T0930
LOCATION:1CP Conference Room 4350
```

END:VEVENT
END:VCALENDAR

The following is an example of an iCalendar object passed in a MIME message with a single body part consisting of a "text/calendar" Content Type.

```
TO:jsmith@host1.com
FROM:jdoe@host1.com
MIME-VERSION:1.0
MESSAGE-ID:<id3@host1.com>
CONTENT-TYPE:text/calendar

BEGIN:VCALENDAR
METHOD:xyz
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT
DTSTAMP:19970324T1200Z
SEQUENCE:0
UID:uid3@host1.com
ORGANIZER:jdoe@host1.com
ATTENDEE;RSVP=TRUE:MAILTO:jsmith@host1.com
DTSTART:19970324T123000Z
DTEND:19970324T210000Z
CATEGORIES:MEETING,PROJECT
CLASS:PUBLIC
SUMMARY:Calendaring Interoperability Planning Meeting
DESCRIPTION:Discuss how we can test c&s interoperability\n
  using iCalendar and other IETF standards.
LOCATION:LDB Lobby
ATTACH:file:///xyzCorp.com/conf/bkgrnd.ps
END:VEVENT
END:VCALENDAR
```

The following is an example of a to-do due on April 15, 1998. An audio alarm has been specified to remind the calendar user at noon, the day before the to-do is expected to be completed and repeat hourly, four additional times. The to-do definition has been modified twice since it was initially created.

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VTODO
DTSTAMP:19980130T134500Z
SEQUENCE:2
UID:uid4@host1.com
```

ORGANIZER:unclesam@us.gov
ATTENDEE;ATTSTAT=ACCEPTED;ATTSTAT=ACCEPTED:MAILTO:jqppublic@host.com
DUE:19980415T235959
STATUS:NEEDS-ACTION
SUMMARY:Submit Income Taxes
BEGIN:VALARM

Dawson/Stenerson

103

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

ALARM-TYPE:AUDIO
TRIGGER:19980403T120000
ATTACH:http://host.com/pub/audio-files/ssbanner.wav
REPEAT:4
DURATION:PT1H
END:VALARM
END:VTODO
END:VCALENDAR

The following is an example of a journal entry.

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VJOURNAL
DTSTAMP:19970324T120000Z
UID:uid5@host1.com
STATUS:DRAFT
CLASS:PUBLIC
CATEGORY:Project Report, XYZ, Weekly Meeting
DESCRIPTION:Project xyz Review Meeting Minutes\
Agenda\
1. Review of project version 1.0 requirements.\n2.
Definition
of project processes.\n3. Review of project schedule.\nParticipants: John Smith, Jane Doe, Jim Dandy\n-It was
decided that the requirements need to be signed off by
product marketing.\n-Project processes were accepted.\n-Project schedule needs to account for scheduled holidays
and employee vacation time. Check with HR for specific
dates.\n-New schedule will be distributed by Friday.\n-Next weeks meeting is cancelled. No meeting until 3/23.
END:VJOURNAL
END:VCALENDAR

The following is an example of published busy time information. The
iCalendar object might be placed in the network resource
www.host.com/calendar/busytime/jsmith.ifb.

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//RDU Software//NONSGML HandCal//EN
BEGIN:VFREEBUSY
ATTENDEE:jsmith@host.com
DTSTART:19980313T141711Z
DTEND:19980410T141711Z
FREEBUSY:19980314T233000Z/19980315T003000Z
FREEBUSY:19980316T153000Z/19980316T163000Z
FREEBUSY:19980318T030000Z/19980318T040000Z
URL:http://www.host.com/calendar/busytime/jsmith.ifb
END:VFREEBUSY
END:VCALENDAR
```

Dawson/Stenerson

104

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

6 Recommended Practices

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

1. Content lines longer than 75 character SHOULD be folded.
2. A calendar entry with a "DTSTART" property but no "DTEND" property - 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 NOT be used to record busy time no matter what the value for the "TRANSP" property.
3. When the "DTSTART" and "DTEND", for "VEVENT", "VJOURNAL" and "VFREEBUSY" calendar components, and "DTSTART" and "DUE", for "VTODO" calendar components, have the same value data type (e.g., DATE-TIME), they should specify values in the same time format (e.g., UTC time format).
4. A combination of "RRULE" and "RDATE" properties 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.
5. A particular iCalendar object method that specifies "ATTENDEE" properties with the "MEMBER" property parameter, for which the recipient has multiple memberships - Recipient should reply to

only the first "MEMBER" property parameter value that it can match.

6. An implementation MAY truncate a "SUMMARY" property value to 255 characters.
7. If seconds of the minute are not supported by an implementation, then a value of "00" should be specified for the seconds component in a time value.

7 Registration of Content Type Elements

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

7.1 Registration of New and Modified iCalendar Object Methods

New MIME Calendaring and Scheduling Content Type iCalendar object methods are registered by the publication of an IETF Request for Comment (RFC). Changes to an iCalendar object method are registered by the publication of a revision of the RFC defining the method.

| | | |
|------------------|-------------------------------|------------------------|
| Dawson/Stenerson | 105 | Expires September 1998 |
| Internet Draft | C&S Core Object Specification | March 6, 1998 |

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

Dawson/Stenerson

106

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

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 Method Reviewer for approval. The Method 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 Method Reviewer to the IANA for inclusion in the official IANA method 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 Method 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.

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 memo. The Method 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 References

The following document are referred to within this memo.

[ICMS] "Internet Calendaring Model Specification", Internet-Draft, July 1997, <ftp://ftp.ietf.org/internet-drafts/draft-ietf-calsch-mod-03.txt>.

[IMIP] "iCalendar Message-based Interoperability Protocol (IMIP)", Internet Draft, February 1998, <http://www.imc.org/draft-ietf-calsch-imip-04.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.

[ITIP] "iCalendar Transport-Independent Interoperability Protocol (iTIP) : Scheduling Events, Busy Time, To-dos and Journal Entries ", Internet-Draft, February 1998, <http://www.imc.org/draft-ietf-calsch-itip-03.txt>.

[MIME DIR] Howes, T., Smith, M., "A MIME Content-Type for Directory Information", Internet-draft-ietf-asid-mime-direct-07.txt, November, 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 1983] "Internet Users' Glossary", [RFC 1983](#), August 1996.

[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.

Internet Draft

C&S Core Object Specification

March 6, 1998

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

[RFC 2111] "Content-ID and Message-ID Uniform Resource Locators", [RFC 2111](#), March 1997.

[RFC 2119] "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 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>.

[VCARD] Internet Mail Consortium, "vCard - The Electronic Business Card Version 2.1", <http://www.imc.org/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.

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

9 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, Philippe Boucher, Steve Carter, Andre Courtemanche, Dave Crocker, David Curley, Alec Dun, John Evans, Ross Finlayson, Randell Flint, Ned Freed, Patrik Falstrom, Chuck Grandgent, Mark Handley, Steve Hanna, Paul B. Hill, Paul Hoffman, Ross Hopson, Mark Horton, Bruce Kahn, C. Harald Koch, Ryan Jansen, 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, Doug Royer, Andras Salamar, Ted Schuh, Vinod Seraphin, Derrick Shadel, Ken Shan,

Andrew Shuman, Steve Silverberg, William P. Spencer, John Sun, Mark Towfiq, Yvonne Tso, Robert Visnov, James L. Weiner, Mike Weston, William Wyatt.

10 Author's Address

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

The authors of this draft are:

| | | |
|------------------|-----|------------------------|
| Dawson/Stenerson | 109 | Expires September 1998 |
|------------------|-----|------------------------|

| | | |
|----------------|-------------------------------|---------------|
| Internet Draft | C&S Core Object Specification | March 6, 1998 |
|----------------|-------------------------------|---------------|

```
BEGIN:VCARD
VERSION:2.1
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;PREF:Frank_Dawson@Lotus.com
EMAIL;INTERNET:fdawson@earthlink.net
URL:http://home.earthlink.net/~fdawson
END:VCARD
```

```
BEGIN:VCARD
VERSION:2.1
FN:Derik Stenerson
ORG:Microsoft Corporation
ADR;WORK;POSTAL;PARCEL:;One Microsoft Way;
  Redmond;WA;98052-6399;USA
TEL;WORK;MSG:+1-425-936-5522
TEL;WORK;FAX:+1-425-936-7329
EMAIL;INTERNET:deriks@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:

```
BEGIN:VCARD
VERSION:2.1
FN:Anik Ganguly
ORG:OnTime, Inc.
```

ADR;WORK;POSTAL;PARCEL:10 Floor;21700 Northwestern Highway;
Southfield;MI;48075;USA
TEL;WORK;MSG:+1-248-559-5955
TEL;WORK;FAX:+1-248-559-5034
EMAIL;INTERNET:anik@ontime.com
END:VCARD

11 Full Copyright Statement

"Copyright (C) The Internet Society (date). All Rights Reserved.

This document and translations of it MAY be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation MAY be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself MAY not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process MUST be

Dawson/Stenerson

110

Expires September 1998

Internet Draft

C&S Core Object Specification

March 6, 1998

followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Dawson/Stenerson

111

Expires September 1998