

IETF CALSCH Working Group  
Issues Document  
<[draft-ietf-calsch-ical-issues-00.txt](#)>  
Expire in six months

Anik Ganguly/OnTime  
Frank Dawson/IBM  
Derik Stenerson/Microsoft  
February 2, 1997

## **Core Object Specification - Issues Document**

### **Abstract**

This document is an IETF CALSCH working group document. The document is used as a tool for recording issues and their resolution with mailing list discussions.

This Issues Document is intended to track the resolution of issues related to the \_Core Object Specification\_ deliverable.

The most current version of this document can be found on the IETF CALSCH Working Group document archive at <http://www.imc.org>.

Issues within this document are classified as either being OPEN or RESOLVED. Additionally, an issue may be found to no longer be a concern and may be classified as WITHDRAWN. Issues falling into each of these categories is listed in a separate section.

An issues is documented with a short summary title, a brief description, and a list of identified alternatives. A resolved issue will also include a description of the resolution and the date/venue that the resolution was reached.

Distribution of this document is unlimited.

### **1. Open Issues**

#### **1.1 Property Definition (Normalization)**

Description: Should data type values be defined using simple base data types or should we allow structured data types (like a 'C' struct)?

Requirements: Solution must have

- . Mechanism for grouping.
- . Named parameters for simplified parsing/ease of extensibility
- . Extensibility

Alternatives:

- 1.Type name and value consisting of multiple data types combined..

1

IETF CALSCH    Core Object Specification - Issues Document    02/03/97

For example:

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

The advantage of this format is compactness of the data. The disadvantage is the cost of more specialized parsing code that is required to break this specialized data type apart and extract the base data type components.

- 2.Type name and value consisting of a single base data type. (Normalized).

For example:

DALARM-DATE:19960415T235000  
DALARM-REMIND:000500  
DALARM-SNOOZE:2  
DALARM-MESSAGE:Your Taxes Are Due !!!

The disadvantage of this format is compactness of the data. The advantage is that generic parsing code can be used and no specialized data types beyond the base data types (string, date-time, etc) need to be defined.

- 3.Combine alternative 2 with the BEGIN:ALARM/END:ALARM, to meet the grouping requirement.

- 4.Use TERM CAPS type model. (Need someone to expand on this.)

- 5.Combine alternative 2 with MIME-DIR style grouping. Allows for unambiguous, multiple occurrences of property. Example:

A.DALARM-DATE:19960415T235000  
A.DALARM-REMIND:000500  
A.DALARM-SNOOZE:2  
A.DALARM-MESSAGE:Your Taxes Are Due !!!

## **1.2    Content Type**

Description: What content type and subtype should be specified in the specification?

Alternatives:

1. Use the "application/calendar" content type/subtype. The purpose of the "application" Content-Type as defined in [section 7.4 of \[RFC-1521\]](#) is "for data to be processed by mail-based uses of application programs." Calendaring and scheduling data falls into this category as recommended by [\[RFC-1521\]](#). The "application/calendar" Content-Type is used to hold textual calendaring and scheduling information. Applications that want to supply a textual representation for legacy systems should use multipart/alternative with a text/plain representation and an application/calendar representation.

2

IETF CALSCH Core Object Specification - Issues Document

02/03/97

2. Use the "text/calendar" content type/subtype. The specification uses a clear-text format that is reasonably readable. In addition, the default action for MIME user agents that do not understand the "calendar" subtype will be to render as a "text/plain" content type/subtype. This is what will need to be done for all legacy systems. This is a very important constituent for this content type. Otherwise, the content type would be rendered as a binary attachment.
3. Use a framework media type such as being used in MIME-DIR.

### **[1.3](#) BEGIN/END Content Sentinel Properties**

Description: Should the MIME calendaring and scheduling content include a BEGIN and END sentinel property?

Alternatives:

1. Include the BEGIN/END sentinel properties. These properties will allow the content to be identified outside of the MIME message transport. They do have any significant overhead.
2. Do not include the BEGIN/END sentinel properties. They are not needed in the MIME encapsulation of the content type. Having another mechanism for delimiting MIME objects, adds to the overhead required to parse such objects, particularly if multiple calendaring and scheduling objects are permitted in a single body part. On the other hand, using the encapsulation methods provided by MIME allows applications to leverage protocols, such as IMAP, extract individual

calendaring and scheduling content objects.

#### **1.4 Recurring Event Model**

Description: What model should be used for representing recurrences within calendar component descriptions. For example, how will recurring events be represented?

Alternatives:

1. Base the recurrences model on that specified in the vCalendar specification. The vCalendar specification includes a model that allows both the representation of recurrences as a sequence of discrete recurring dates and as a recurrence pattern with a recurrence grammar. The model also allows the inclusion of exceptions to a calendar component description using the same options of a sequence of discrete exception dates or an exception pattern. This model has been implemented by numerous vendors. It is based on previous work of the IETF CHRONOS working group and accepted practice in \_small foot print\_ machines such as Personal Digital Assistants (PDA).

2. Base the representation of recurring events on a model that describes calendar based recurrence patterns ranging from the simple to the complex in a manner that is simple to implement properly and flexible enough to allow for support non-gregorian calendars. The [draft-ietf-calsch-sch-00.txt](#) draft attempts to define such a model, implemented as a set calendar restrictions, similar to a database query, combined with reference date and time information. Each restriction, or pattern, is combined with the next pattern using the logical AND operations. The query style method allows for a wide variety of patterns to be specified without complex parsing or grammar.

#### **1.5 Date and Time Format**

Description: What date and time format should be used for properties with date and time value types?

Alternatives:

1. Use the revised [RFC 822](#) date and time format defined by [RFC 1123](#). This is the format that is used within the MIME messaging transport. It is also used by numerous IETF protocols.
2. Use the ISO 8601 based date and time format defined by the `_csct-01.txt` Internet Draft. This is the format that is used with

## **[1.6](#) DST Rule Support**

Description: Should the specification for Time Zone include support for specifying the behavior and rules DST? If so, what format should be used?

Alternatives:

1. No. The specification should not include support for calendaring functionality that depends on the specification of the behavior and rules for DST observance.
2. Yes. The specification should include a property that defines the behavior and rules for DST observance; based on the POSIX model. This would include a boolean value defining DST observance, the offset from UTC for standard time, offset from UTC for daylight savings time, date and time or recurrence rule for beginning standard time, date and time or recurrence rule for beginning DST, optional string for the standard time zone name, optional string for the DST zone name.
3. Yes. The specification should include a set of properties that define the behavior and rules for the time zone. This would include at a minimum the time zone time delta from the

prime meridian. In addition, if necessary (if DST is observed): a) a time delta used in DST, b) a Standard Time transition rule, i.e. a recurrence rule or list of absolute date/times; and c) a DST transition rule, i.e. a recurrence rule or list of absolute date/times. Additional optional properties may be desirable including a Time Zone identifiers, a time zone name for STD and DST time, a date-stamp to indicate "freshness" of the time zone rule, a URL to standardized time zone rule definitions, etc.

## **1.7 Exceptions To Recurrence Rules**

Description: How should exceptions to a recurrence rule be specified (e.g., as a sequence of dates, an exception recurrence rule, or optionally both)?

Alternatives:

1. The recurrence model should support the specification of exceptions to the recurrence as a sequence of dates.
2. The recurrence model should support the specification of exceptions to the recurrence as a sequence of dates and optionally an exception recurrence rule or both.

## **1.8 Infinite Recurring Patterns**

Description: Should the recurrence rules allow for an infinite number of recurrences?

Alternatives:

1. No. The recurrence rules need to specify a finite number of occurrences.
2. Yes. The recurrence rules need to allow for an open-ended, infinite number of recurrences.
3. Yes. The recurrence rules need to allow for open-ended recurrence rules. However, there also needs to be a way of specifying a stop date or count to the open-ended recurrence rule.

## **1.9 Non-Standard Extensibility**

Description: Should the specification support a formal method for making \_non-standard\_ extensions to the specification?

Alternatives:

1. No. The specification should limit support for extensions in order to maximize possible interoperability.

2. Yes. The specification should include the [RFC 1521](#) method of

\_X\_ tags for non-standard extensions to the property names and values and non-standard extensions to the property parameter names and values. Standard extensions should also be supported via the IANA registration process of new property names and values or new property parameter names and values.

### **1.10 Local Time Values**

Description: Should we allow local time values for date and time value types within the specification?

Requirement: Solution must have time values specified such that they are globally unambiguous to the recipient(s) of the calendar object.

Alternatives:

1. No. Only UTC values should be specified for data and time values.
2. Yes. However, the local time should always include the time zone offset from UTC.

## **2. Resolved Issues**

### **2.1 Scope and Application of Specification**

Description: Should the specification be defined with the intent that the content type is to be used solely within a SMTP/MIME message transport or should there be a broader scope and application taken into the definition of the specification?

Alternatives:

1. The specification should be designed with the scope and application target of just the SMTP/MIME messaging transport.
2. The specification should be designed with the scope and application target of just the IETF transport protocols.
3. The specification is for an industry calendaring and scheduling standard. The scope and application target should be a broad range of IETF and non-IETF transport protocols.

Resolution: Alternative 3. (Working Group Charter/1996-10)

### **2.2 Property Syntax**

Description: What syntax should be used to represent individual properties or MIME body elements within the specification?

Alternatives:

1. Properties are to be defined using the syntax from vCalendar:  
propname \*[:parmname \_=\_ parmvalue] \_:\_ propvalue
2. Properties are to be defined using the syntax from the July 1996 Microsoft document:  
propname [\_,\_ parmname \_=\_ parmvalue] \_:\_ propvalue

Resolution: Alternative 1. (Mailing List/1996-12).

### **2.3 Ordering of Properties**

Description: Should the specification require ordering of properties?

Alternatives:

1. No. There is not ordering requirement for properties, other than sentinel properties.
2. Yes. The ordering of some properties is important.

Resolution: Alternative 1. (Mailing List/1996-11)

### **2.4 Specification of Usage Profiles**

Description: How should the specification encode the originator's intent with respect to the usage profile for content information conforming to the specification?

Alternatives:

1. Include within the specification a new MIME header field CONTENT-PROFILE that will declare the intended usage profile.
2. Include within the specification a new \_profile=\_ header field parameter for the MIME Content-Type header field. This parameter will declare the intended usage profile.
3. Include within the specification both a new \_profile=\_ header field parameter for the MIME Content-Type header field and a new optional PROFILE property for the content information. These two elements will be used to declare the intended usage profile. The PROFILE property is needed within the content information in the event that the content information is



transported using a non-MIME messaging transport, but is not required when the content information is transported in MIME.

Resolution: Alternative 3 (Mailing list/1996-12).

## **2.5 Strong, Explicit Data Typing**

Description: Should the specification include the definition of properties with strong data typing?

Alternatives:

7

IETF CALSCH Core Object Specification - Issues Document

02/03/97

1. The specification should implicitly define the data types for the properties within the specification. While the content information itself does not include any explicit data typing information with the properties, it is defined within the specification.
2. The specification must include a mechanism for explicitly defining the data types for the properties within the specification. The content information includes explicit data typing with a VALUETYPE property parameter. A minimal set of value data types are defined by the specification. Additional value data types can be registered within the IANA registration procedures. The value data type must be explicitly declared on each property within the content information.
3. The specification must include a mechanism for explicitly defining the data types for the properties within the specification. Additionally, the specification must include a default value for the data type; in the event that the value data type is not explicitly specified in the content information. The content information includes explicit data typing with a VALUETYPE property parameter. A minimal set of value, data types are defined by the specification. Additional value, data types can be registered within the IANA registration procedures. The value, value data type may be explicitly declared on each property within the content information. If the value data type is not specified, it is defaulted to the default data type for the property value.

Resolution: Alternative 3. (Mailing list/1996-10)

## **2.6 Minimalization of Property Names**

Description: Should property names be specified using the verbose style of MIME or a more minimal style for \_low chat\_ and \_small foot print\_ devices?

Alternatives:

1. Use the verbose style of MIME for defining names. This is easier to read and comprehend.
2. Use a minimal, short name for properties. This is more suitable for small size datagrams. In addition, it is friendly to \_low chat\_ transports and small \_foot print\_ devices, like a PDA.

Resolution: Alternative 1. When creating property names, make them as short as possible. (Mailing List/1996-11)

## [2.7](#)    **Multi-valued Property Values**

Description: Should property values be allowed to have multiple values?

Alternatives:

1. No. A property value can only appear once in a content information with one value.
2. Yes. A property value may appear multiple times in a content information with multiple values.

Resolution: Alternative 2 (Mailing List/1996-11)

## [2.8](#)    **Data Model**

Description: What calendaring and scheduling data model should the specification use?

Alternatives:

1.    Base the model on the vCalendar specification. This includes a model of a calendar object as a conceptual container for

calendar components including events and todos. This model is heavily borrowed from the XAPIA and X/Open Calendaring and Scheduling API (CSA) which represents a data model that has some broad consensus among a group of calendaring and scheduling vendors. It has also been implemented on over four operating system platforms by numerous vendors.

2. Base the model on an architecture that is new and developed within the IETF. Where appropriate, utilize the best ideas from existing products or model specifications to derive a standard based on the consensus of the IETF Calendaring and Scheduling Working Group.

Resolution: Per the pre-working group meeting, we've decided to use the CSCT draft as our starting point for the working group spec. The entire draft is up for review and group consensus will be reflected in any modification made to the draft. Modifications will be made via postings of change text to the list.

## **2.9 Attendees In MIME Header Fields and Content Body**

Description: When calendar components are transported in the form of a MIME message, how should the attendees be specified?

Alternatives:

1. Attendees specified using the [RFC-822](#) addressing fields. For example, "To:" header are "required", those in the "Cc:" header are "optional", etc.

2. Attendees specified with the `_ATTENDEE_` properties in the content information.
3. Attendees specified by 1 and 2.
4. Attendees specified by 1 and optionally 2, where 2 has precedence over 1 if 2 is specified.

Resolution: Alternative 2, Attendees specified with the `_ATTENDEE_` properties in the content information.

## **2.10 Non-Gregorian Calendar**

Description: Should support be provided in the specification for

specifying calendar components using non-Gregorian calendar systems?

Alternatives:

1. No. Only Gregorian-based descriptions of calendar components are supported?
2. Yes. The specification should support specification of calendar components using a non-Gregorian calendar scale.

Resolution: Modification of alternative 2. A mechanism for specifying alternate calendar types will be defined: a calendar scale property will allow the calendar scale to be named. However, the specification will only define behavior for the Gregorian calendar scale. Alternate calendar types and their behaviors, conversion rules, etc. will be defined in separate documents.

### **3. Withdrawn Issues**

#### **3.1 Functional Completeness**

Description: What types of scheduling functionality should be included in the specification?

Alternatives:

1. Just include support for requesting, replying-to and canceling an event.
2. Begin with the base concepts of requesting, replying-to and canceling an event. Add additional functionality that is deemed as required by the group.
3. Start with as full a set of scheduling functionality as possible (e.g., requesting, replying-to, canceling, modifying, replacing, resending, negotiating events and todos, as well as requesting and replying-to free/busy time requests.). Let the group decide what to add and remove.

Resolution: Alternative #3 fairly closely represents the decision of the group by accepting the CSCT draft as the basis for the core object specification. However, the issue is still valid in the context of the Calendaring and Scheduling Content Type Profile

document <[draft-ietf-calsch-csp-xx.txt](#)> and should be addressed there.