

# Draft NANOG BCOP

**Shepherd:** Erik Klavon (erik.klavon@gmail.com)

**Subject Matter Expert(s) (SME):** Francisco Hidalgo, Tylar Keese, Tj Trask, Sean Stuart, Randy Neals, Peter Hoose, Dave McGaugh, Paul Schultz, Joel Wride

**Status:** Draft 0.1

**BCOP Subject:** A machine parseable standard for formatting maintenance notifications

## 1. BCOP Summary (Appeal)

The format of maintenance notification varies from sender to sender, making it difficult to automate processing of these messages. This BCOP defines conventions for machine parseable formatting of information within common forms of maintenance notifications.

## 2. BCOP Background / History

At NANOG 60, Randy Neals floated the idea of improving maintenance notifications to make it easier to write programs to consume them. Noting a willingness on the part of several attendees (Peter Hoose, Erik Klavon, Dave McGaugh, Paul Schultz, and Joel Wride) to collaborate on drafting standards for machine parseable maintenance notifications, Randy organized the initial effort. Randy held a couple conference calls and worked with the group to draft a rough charter. Unfortunately Randy had to withdraw from the role of shepherd due to competing priorities. Erik picked up where Randy left off, and - thanks to the advocacy of the other original participants - began working with Francisco Hidalgo, Tylar Keese, and Tj Trask on the project.

All of the participants generate and consume communications on maintenance activities. Some of us spend a significant amount of time manually processing these communications, while others have invested a lot of time writing programs to automate their processing. Common tasks we perform on these notifications include converting time zones, adding entries to calendars, and opening tickets to track the maintenance and our response to it. Where our processing of these notifications is manual, we run the risk of introducing errors due to human fallibility. Processing maintenance notifications is a repetitive task, and based on our experience computers handle such repetitive tasks with a greater degree of accuracy than humans.

Those of us who have automated the processing of maintenance communications have encountered several challenges. Much of the time these notifications are carried in an email, the format of which varies from sender to sender. To handle this variety in format we've had to write custom routines for each vendor. We've also found that formatting of these notifications is not always consistent from the same vendor, adding still more complexity to our software. Sometimes vendors modify their internal standards for the format of their maintenance notifications, but do not provide any sort of advanced notice, resulting in incorrectly processed messages and urgent efforts to reverse engineer the new conventions and modify our software to support them. While we've made gains from automatically processing these messages, there's a lot of room for improvement in reducing the complexity of our automation software and how much effort it takes to support it.

We believe that a machine parseable standard for the formatting of key information in maintenance notifications will enable many more operators to automate their processing of these messages, by significantly reducing the initial effort required to write software to process them, and by reducing the amount of effort required by ongoing maintenance of this software. By automating the processing of these messages, operators will improve their efficiency and the reliability of their services, both of which benefit the entire community.

## 3. BCOP

### 3.1 Introduction

Maintenance notifications contain information about events related to a service, such as a scheduled maintenance or an unplanned outage. These notifications are generated by the provider of the service and sent to consumers of the service. An essential part of the information contained in these notifications are the dates and times related to these events, general types of data which may be represented using the iCalendar data format [RFC5545]. Other essential parts of these notifications, such as impact and service identifiers, are types of data specific to maintenance notifications for which there is no structured means of representation in iCalendar.

In order to facilitate automated processing of maintenance notifications, this BCOP extends the iCalendar standard to include additional data formats for representing maintenance notification information. It also describes how iCalendar events should be used to communicate common maintenance communication sequences.

### 3.2 Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

This document closely follows the formatting conventions of [RFC5545], including the use of descriptive text and more formal notation for defining data formats for representing maintenance notification information. The formal notation used in this document is the ABNF notation of [\[RFC5234\]](#).

### 3.3 iCalendar Object

Maintenance Notification information MUST be represented using the iCalendar object as described in [RFC5545] section 3.4.

### 3.4 Calendar Properties

The following Calendar Properties are in scope for Maintenance Notifications:

#### Property Presence Required Comment

VERSION 1	Required per [RFC5545]
PRODID 1	Required per [RFC5545]
METHOD 0 or 1	See below

#### 3.4.1 METHOD Calendar Property

The METHOD Calendar Property MAY be included in Maintenance Notification Calendar Objects. By default implementations SHOULD NOT include the METHOD Calendar Property. Implementations MAY add the METHOD Calendar Property when they intend for the iCalendar Object to represent a scheduling transaction, with the value set to the desired transaction method. For more information on the METHOD Calendar Property, and other required fields depending on its value, see [RFC5546].

When an iCalendar Object represents a scheduling transaction, Human facing Calendaring systems may attempt to process the transaction. Experimentation has shown this to result in the addition of an event to a calendar contained within the Calendaring system. The desirability of this outcome will vary based on the recipient. For example, it may be desirable for maintenance notifications to auto populate to a shared calendar devoted to maintenances. It may not be desirable for maintenance notifications to auto populate to personal calendars. Providers SHOULD implement a way for recipients of their maintenance notifications to

determine individually if their notifications will include the METHOD Calendar Property. Recipients MAY need to perform some level of pre-processing in order to ensure that maintenance notifications do not interact with their Human Calendaring systems in undesirable ways.

### 3.5 Event Component and Associated Properties

Maintenance Notification information MUST be represented using the iCalendar Event Component as described in [RFC5545] section 3.6.1. All of the following Properties MUST be included with any iCalendar Event for it to be a properly formatted maintenance notification. These properties are the minimum set required to automate common processing and dispatching of maintenance notifications. Implementors MAY include and/or parse other iCal Properties, however the presence of other iCal Properties MUST NOT conflict with the use of mandatory Properties listed below.

<b>Property</b>	<b>Presence Required</b>	<b>Comment</b>
DTSTAMP	1	
DTSTART	1	
DTEND	1	
UID	1	
SUMMARY	1	
ORGANIZER	1	
SEQUENCE	1	
X-MAINTNOTE-PROVIDER	1	See below
X-MAINTNOTE-ACCOUNT	1	See below
X-MAINTNOTE-MAINTENANCE-ID	1	See below
X-MAINTNOTE-OBJECT-ID	1+	See below
X-MAINTNOTE-IMPACT	1	See below
X-MAINTNOTE-STATUS	0 or 1	See below

### 3.6 Descriptive Component Properties

The following properties extend iCalendar to specify additional descriptive information specific to the maintenance notification use case. Note that due to the status of this document as a NANOG BCOP, these extensions fall under the non-standard properties class of iCalendar properties defined in section 3.8.8.2 of [RFC5545]. To avoid possible collision with other non-standard properties, and in keeping with the suggested approach defined in section 3.8.8.2 of [RFC5545], all properties defined in this document have the prefix text "X-MAINTNOTE-" where "MAINTNOTE" is the short text that identifies the common nature and purpose of these extensions.

Testing of the effect of adding non-standard properties of this format with several consumers of iCalendar formatted information has shown that these properties will be ignored by consumers not configured to interpret them.

#### 3.6.1 Provider

Property Name: X-MAINTNOTE-PROVIDER

Purpose: This descriptive component property contains text that identifies the provider of the service that is the subject of the maintenance notification.

Value Type: TEXT

Property Parameters: IANA, non-standard, and language property parameters can be specified on this property.

Conformance: This property can be specified in "VEVENT" calendar component.

Description: This field MUST contain text that identifies the provider of the service that is the subject of the maintenance notification. The text used SHOULD be chosen to ensure uniqueness, such as by using the well known trademark of the provider or using a registered string from a globally unique namespace (for example a domain name associated with the provider, e.g. example.com).

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

```
x-maintnote-provider = "X-MAINTNOTE-PROVIDER" *(";" icalparameter) ":" text CRLF
```

Example: The following example is of a provider descriptive component property for the provider with the domain name example.com:

```
X-MAINTNOTE-PROVIDER:example.com
```

### 3.6.2 Account

Property Name: X-MAINTNOTE-ACCOUNT

Purpose: This descriptive component property contains text that identifies an account associated with the service that is the subject of the maintenance notification.

Value Type: TEXT

Property Parameters: IANA, non-standard, and language property parameters can be specified on this property.

Conformance: This property can be specified in "VEVENT" calendar component.

Description: This field MUST contain text that identifies an account associated with the service that is the subject of the maintenance notification.

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

```
x-maintnote-account = "X-MAINTNOTE-ACCOUNT" *(";" icalparameter) ":" text CRLF
```

Example: The following example is of an account descriptive component property:

```
X-MAINTNOTE-ACCOUNT:137.035999173
```

### 3.6.3 Maintenance ID

Property Name: X-MAINTNOTE-MAINTENANCE-ID

Purpose: This descriptive component property contains text that uniquely identifies the maintenance that is the subject of the notification.

Value Type: TEXT

Property Parameters: IANA, non-standard, and language property parameters can be specified on this property.

Conformance: This property can be specified in "VEVENT" calendar component.

Description: This field MUST contain text that uniquely identifies the maintenance that is the subject of the notification from all other unrelated notifications sent by the provider.

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

```
x-maintnote-maintenance-id = "X-MAINTNOTE-MAINTENANCE-ID" *(";" icalparameter) ":" text CRLF
```

Example: The following example is of an account descriptive component property:

```
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415
```

### 3.6.4 Object ID

Property Name: X-MAINTNOTE-OBJECT-ID

Purpose: This descriptive component property contains text that uniquely identifies a service object that is within the scope of the maintenance that is the subject of the notification.

Value Type: TEXT

Property Parameters: IANA, non-standard, and language property parameters can be specified on this property.

Conformance: This property can be specified in "VEVENT" calendar component.

Description: This field MUST contain text that uniquely identifies a service object that is within the scope of the maintenance that is the subject of the notification. The object identifier MUST distinguish the referenced service object from all other unrelated service objects of the provider. The consumer of the field SHOULD use both the object identifier descriptive component property and the provider descriptive component property to identify the service to protect against object identifier namespace collisions across providers. Multiple instances of this descriptive component MAY be included simultaneously where the scope of a maintenance includes multiple service objects.

Providers SHOULD choose a service object identifier which is most descriptive for the service(s) under maintenance, and which best enables automation by the recipient. In some cases, a unique (to the service provider) identifier without any technical connection to the service that is arranged at the time of initial service provisioning and communicated to the recipient as the future identifier for the service may be the best option. In other cases, some identifier associated with the technical expression of the service may be the best option. Identifiers that appear in both provider and recipient technical configuration related to the service SHOULD be used whenever possible; otherwise an identifier that appears in the recipient technical configuration related to the service MAY be used.

In cases where the provider wishes to provide additional formatted information beyond a single unique identifier, the Alternative Text Representation property parameter defined in [RFC 5545] section 3.2.1 SHOULD be used to provide a URI where additional data may be obtained. Any guidelines for the formatting of this external information is outside of the scope of this standard, however the authors recommend that the provider use a well known encoding method such as JSON [RFC 7159] and a naming standard for different types of data commonly used for the service in question. One of the examples below shows a query against the API for the popular PeeringDB service that - when populated with valid parameters - should return JSON formatted data for two networks' presences on a public Internet exchange. This example shows one way of providing additional formatted information for a particular use case, specifically BGP peering on public Internet exchanges.

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

```
x-maintnote-object-id = "X-MAINTNOTE-OBJECT-ID" *(";" icalparameter) ":" text CRLF
```

Example: The following are examples of service objects:

```
X-MAINTNOTE-OBJECT-ID:2718281828459
X-MAINTNOTE-OBJECT-ID;ALTREP="https://example.org/maintenance?
id=2718281828459":2718281828459
X-MAINTNOTE-OBJECT-ID;ALTREP="https://www.peeringdb.com/api/netixlan?
asn\_in=64496,65536&ipaddr4\_in=192.0.2.42,192.0.2.137":2718281828459
X-MAINTNOTE-OBJECT-ID:acme-widgets-as-a-service
X-MAINTNOTE-OBJECT-ID:vm-1054571726.region.example.com
X-MAINTNOTE-OBJECT-ID:2001:db8::d06:f00d
X-MAINTNOTE-OBJECT-ID:198.51.100.13
```

### 3.6.5 Impact

Property Name: X-MAINTNOTE-IMPACT

Purpose: This descriptive component property specifies the impact of the maintenance to the services within its scope.

Value Type: TEXT

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified in "VEVENT" calendar component.

Description: The X-MAINTNOTE-IMPACT property type MUST be specified on this property. This field may contain one of several labels that describes the impact of the maintenance to the services within its scope. The value "NO-IMPACT" indicates that there is no expected impact to services in scope for the maintenance. The value "REDUCED-REDUNDANCY" indicates that during the maintenance the services in scope are expected to continue operating without any consumer visible impact, however the services are without their normal level of redundancy. While operating at a reduced level of redundancy, failure of supporting infrastructure outside the scope of the maintenance occurring concurrent to the maintenance may cause consumer visible service impact. The value "DEGRADED" indicates that negative impact to services in scope for the maintenance is expected, however the maintenance will not result in a total service outage. The value "OUTAGE" indicates that the services in scope of the maintenance are expected to be completely out of service. Implementors MAY create other property parameter values that better describe their specific situations. The default value is "OUTAGE"; applications SHOULD treat x-name and iana-token values they don't recognize the same way as they would the "OUTAGE" value.

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

```
x-maintnote-impact = "X-MAINTNOTE-IMPACT" *(";" icalparameter) ":" impactvalue CRLF
```

```
impactvalue      = "NO-IMPACT" / "REDUCED-REDUNDANCY"
                  / "DEGRADED" / "OUTAGE"
                  / x-name      ; Some experimental impact property parameter value.
                  / iata-token  ; Some other IANA-registered impact property parameter value.
```

Example: The following are examples of service objects:

```
X-MAINTNOTE-IMPACT:DEGRADED
```

### 3.6.6 Status

Property Name: X-MAINTNOTE-STATUS

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

Value Type: TEXT

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified once in "VEVENT" calendar components.

Description: The property is used by the "Organizer" to provide information regarding the status of the maintenance event.

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

```
x-maintnote-status      = "X-MAINTNOTE-STATUS" x-maintnote-statparam ":" x-maintnote-statvalue
CRLF
```

```
x-maintnote-statparam  = *(";" other-param)
```

```
x-maintnote-statvalue  = "TENTATIVE" ;Indicates maintenance event is tentative.
/ "CONFIRMED" ;Indicates maintenance event is definite.
/ "CANCELLED" ;Indicates maintenance event was cancelled.
/ "IN-PROCESS" ;Indicates maintenance event is in process (e.g. open).
/ "COMPLETED" ;Indicates maintenance event completed (e.g. closed).
/ x-name ; Some experimental impact property parameter value.
/ iata-token ; Some other IANA-registered impact property parameter value.
```

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

```
X-MAINTNOTE-STATUS:TENTATIVE
```

## 3.7 Workflow

This section describes several workflows typical to maintenance notifications. It describes the sequence of steps to produce and consume properly formatted ical data extended as specified in this document.

### 3.7.1 Initial Maintenance Notification

This workflow describes the actions required by the example provider and example consumer for an initial maintenance notification.

#### 3.7.1.1 Example Provider

The example provider determine that a maintenance on a network element is required to avoid service disruption in the future. This maintenance will result in the network element going out of service. Their systems determine the impacted services provided by the network element, and the list of accounts and consumer contacts associated with those services. Their systems also generate a maintenance identifier.

Using the above information, individual maintenance notifications are generated for each consumer using existing processes. Automated systems also generate ical formatted attachments following the standards described in this document. These ical attachments include the start and end timestamps for the proposed maintenance window (DTSTART, DTEND), the same summary of the maintenance that was included in the email summary for the notification (SUMMARY), the same contact information included in the email

notification (ORGANIZER), the status of the maintenance as tentative since this is a proposed window (X

notification (X-MAINTNOTE-STATUS), the status of the maintenance as tentative since this is a proposed window (X-MAINTNOTE-STATUS), the well known domain name of the company (X-MAINTNOTE-PROVIDER), the maintenance identifier (X-MAINTNOTE-MAINTENANCE-ID) and the worst case impact (X-MAINTNOTE-IMPACT). Each consumer receives specific information relative to their impacted service(s) in the following fields: X-MAINTNOTE-ACCOUNT includes the consumer's account identifier, one or more instances of X-MAINTNOTE-OBJECT-ID list the consumer's impacted service(s).

To enable association of subsequent updates to these notifications, the following values are tracked in the systems of the example provider. A unique identifier for each notification (UID), so that subsequent updates may be associated with the original notification. A sequence number - initially zero - to serialize updates in case they are received or processed out of order (SEQUENCE).

### **3.7.1.2 Example Consumer**

The example consumer receives the example provider's notification, detects the presence of the ical attachment and routes it to a program for processing. Values of relevance are then parsed. The program examines a database to determine if a maintenance notification with the same unique identifier as this notification has been processed in the past. No entry corresponding to the unique identifier is found. The lack of a matching entry in the database indicates this is an initial notification, so the program opens a new ticket in the consumer's ticketing system. This ticket is used to track the maintenance. Because the maintenance may result in an outage, the priority of the ticket is elevated. Because the notification is tentative, the ticket is not assigned to the remediation queue. The parsing program records the UID and sequence number in a database along with an identifier for the created ticket. This state will be used to update the ticket should any further information be received from the vendor.

### **3.7.2 Updated Maintenance Notification Window**

This workflow describes the actions required by the example provider and example consumer for an updated maintenance notification window.

#### **3.7.2.1 Example Provider**

After reviewing feedback from affected consumers, the example provider negotiates a new maintenance window acceptable to consumers. The example provider's automated systems generate new maintenance notifications using the same information as the initial maintenance notification with several key fields modified. The start and end timestamps for the proposed maintenance window (DTSTART, DTEND) are updated to their new values. Of key importance is the use of the same unique identifier value as the first message, to ensure that consumers are able to associate this new notification with the previous one. The sequence number (SEQUENCE) is incremented to one so that this notification contains newer information than the initial notification that preceded it. All other information is included as it appeared in the initial notification.

#### **3.7.2.2 Example Consumer**

The example consumer receives the example provider's notification, detects the presence of the ical attachment and routes it to a program for processing. Values of relevance are then parsed. The program examines a database to determine if a maintenance notification with the same unique identifier as this notification has been processed in the past. A match is found for the unique identifier, and the sequence number in the current notification is greater than the one in the database indicating that the current notification contains updated information. The associated record includes an identifier for a ticket in the consumer's ticketing system. Given the current notification contains updates to previous information, the program updates fields in the corresponding ticket with values from the current notification. Thus the ticket is updated to contain the new start and end timestamps. The program notes that other key fields - such as the status of the notification - have not changed in a way that requires any updates to the ticket.

### **3.7.3 Canceled Maintenance Notification Window**



This workflow describes the actions required by the example provider and example consumer for a canceled maintenance notification window.

### **3.7.3.1 Example Provider**

After determining that a maintenance window is no longer needed, the example provider cancels a previously announced maintenance window. The example provider's automated systems generate new maintenance notifications using the same information as the last maintenance notification for this maintenance with several fields modified. Of key importance is the use of the same unique identifier value as the prior messages, to ensure that consumers are able to associate this new notification with previous ones. The sequence number (SEQUENCE) is incremented by one from the value used in the last messages, to indicate that this notification contains newer information than the notifications that preceded it. The maintenance status (X-MAINTNOTE-STATUS) field is set to the value CANCELLED. All other information is included as it appeared in the prior notification.

### **3.7.3.2 Example Consumer**

The example consumer receives the example provider's notification, detects the presence of the ical attachment and routes it to a program for processing. Values of relevance are then parsed. The program examines a database to determine if a maintenance notification with the same unique identifier as this notifications been processed in the past. A match is found for the unique identifier, and the sequence number in the current notification is greater than the one in the database indicating that the current notification contains updated information. The associated record includes an identifier for a ticket in the consumer's ticketing system. Given the current notification is that the maintenance previously announced is cancelled, the program cancels the corresponding ticket for the maintenance.

## **3.7.4 Open Maintenance Notification Window**

This workflow describes the actions required by the example provider and example consumer for an open maintenance notification window.

### **3.7.4.1 Example Provider**

At the beginning of the scheduled maintenance window, the example provider declares the announced maintenance window open and begins work. The example provider's automated systems generate new maintenance notifications using the same information as the last maintenance notification for this maintenance with several fields modified. Of key importance is the use of the same unique identifier value as the prior messages, to ensure that consumers are able to associate this new notification with previous ones. The sequence number (SEQUENCE) is incremented by one from the value used in the last messages, to indicate that this notification contains newer information than the notifications that preceded it. The maintenance status (X-MAINTNOTE-STATUS) field is set to the value IN-PROCESS. With the exception of modifications to the textual description stating that the maintenance window is open, all other information is included as it appeared in the prior notifications.

### **3.7.4.2 Example Consumer**

The example consumer receives the example provider's notification, detects the presence of the ical attachment and routes it to a program for processing. Values of relevance are then parsed. The program examines a database to determine if a maintenance notification with the same unique identifier as this notifications been processed in the past. A match is found for the unique identifier, and the sequence number in the current notification is greater than the one in the database indicating that the current notification contains updated information. The associated record includes an identifier for a ticket in the consumer's ticketing system. Given the current notification is an update to the maintenance previously announced maintenance window, the program updates the ticket for the maintenance with the new textual description and marks it as open.

### 3.7.5 Closed Maintenance Notification Window

This workflow describes the actions required by the example provider and example consumer for a closed maintenance notification window.

#### 3.7.5.1 Example Provider

At the end of the scheduled maintenance window, the example provider confirms all work is complete, that all affected services have returned to normal operation, then declares the announced maintenance window closed. The example provider's automated systems generate new maintenance notifications using the same information as the last maintenance notification for this maintenance with several fields modified. Of key importance is the use of the same unique identifier value as the prior messages, to ensure that consumers are able to associate this new notification with previous ones. The sequence number (SEQUENCE) is incremented by one from the value used in the last messages, to indicate that this notification contains newer information than the notifications that preceded it. The maintenance status (X-MAINTNOTE-STATUS) field is set to the value COMPLETED. With the exception of modifications to the textual description stating that the maintenance window is closed, all other information is included as it appeared in the prior notifications.

#### 3.7.5.2 Example Consumer

The example consumer receives the example provider's notification, detects the presence of the ical attachment and routes it to a program for processing. Values of relevance are then parsed. The program examines a database to determine if a maintenance notification with the same unique identifier as this notifications been processed in the past. A match is found for the unique identifier, and the sequence number in the current notification is greater than the one in the database indicating that the current notification contains updated information. The associated record includes an identifier for a ticket in the consumer's ticketing system. Given the current notification is an update to the maintenance previously announced maintenance window, the program updates the ticket for the maintenance with the new textual description and marks it as closed.

### 3.8 Examples

#### 3.8.1 Initial Maintenance Notification

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Maint Note//https://github.com/maint-notification/
BEGIN:VEVENT
SUMMARY:Maint Note Example
DTSTART;VALUE=DATE-TIME:20151010T080000Z
DTEND;VALUE=DATE-TIME:20151010T100000Z
DTSTAMP;VALUE=DATE-TIME:20151010T001000Z
UID:42
SEQUENCE:1
X-MAINTNOTE-PROVIDER:example.com
X-MAINTNOTE-ACCOUNT:137.035999173
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415
X-MAINTNOTE-OBJECT-ID:acme-widgets-as-a-service
X-MAINTNOTE-IMPACT:NO-IMPACT
X-MAINTNOTE-STATUS:TENTATIVE
ORGANIZER;CN="Example NOC":mailto:noone@example.com
END:VEVENT
END:VCALENDAR
```

#### 3.8.2 Updated Maintenance Notification Window

### 3.8.2 Updated Maintenance Notification Window

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Maint Note//https://github.com/maint-notification/
BEGIN:VEVENT
SUMMARY:Maint Note Example
DTSTART;VALUE=DATE-TIME:20151012T080000Z
DTEND;VALUE=DATE-TIME:20151012T100000Z
DTSTAMP;VALUE=DATE-TIME:20151012T001000Z
UID:42
SEQUENCE:2
X-MAINTNOTE-PROVIDER:example.com
X-MAINTNOTE-ACCOUNT:137.035999173
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415
X-MAINTNOTE-OBJECT-ID:acme-widgets-as-a-service
X-MAINTNOTE-IMPACT:NO-IMPACT
X-MAINTNOTE-STATUS:CONFIRMED
ORGANIZER;CN="Example NOC":mailto:noone@example.com
END:VEVENT
END:VCALENDAR
```

### 3.8.3 Canceled Maintenance Notification Window

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Maint Note//https://github.com/maint-notification/
BEGIN:VEVENT
SUMMARY:Maint Note Example
DTSTART;VALUE=DATE-TIME:20151012T080000Z
DTEND;VALUE=DATE-TIME:20151012T100000Z
DTSTAMP;VALUE=DATE-TIME:20151012T001000Z
UID:42
SEQUENCE:3
X-MAINTNOTE-PROVIDER:example.com
X-MAINTNOTE-ACCOUNT:137.035999173
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415
X-MAINTNOTE-OBJECT-ID:acme-widgets-as-a-service
X-MAINTNOTE-IMPACT:NO-IMPACT
X-MAINTNOTE-STATUS:CANCELLED
ORGANIZER;CN="Example NOC":mailto:noone@example.com
END:VEVENT
END:VCALENDAR
```

### 3.8.4 Open Maintenance Notification Window

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Maint Note//https://github.com/maint-notification/
BEGIN:VEVENT
SUMMARY:Open - Maint Note Example
DTSTART;VALUE=DATE-TIME:20151012T080000Z
DTEND;VALUE=DATE-TIME:20151012T100000Z
DTSTAMP;VALUE=DATE-TIME:20151012T001000Z
UID:42
SEQUENCE:3
X-MAINTNOTE-PROVIDER:examole.com
```

```
.....  
X-MAINTNOTE-ACCOUNT:137.035999173  
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415  
X-MAINTNOTE-OBJECT-ID:acme-widgets-as-a-service  
X-MAINTNOTE-IMPACT:NO-IMPACT  
X-MAINTNOTE-STATUS:IN-PROCESS  
ORGANIZER;CN="Example NOC":mailto:noon@example.com  
END:VEVENT  
END:VCALENDAR
```

### 3.8.5 Closed Maintenance Notification Window

```
BEGIN:VCALENDAR  
VERSION:2.0  
PRODID:-//Maint Note//https://github.com/maint-notification/  
BEGIN:VEVENT  
SUMMARY:Closed - Maint Note Example  
DTSTART;VALUE=DATE-TIME:20151012T080000Z  
DTEND;VALUE=DATE-TIME:20151012T100000Z  
DTSTAMP;VALUE=DATE-TIME:20151012T001000Z  
UID:42  
SEQUENCE:4  
X-MAINTNOTE-PROVIDER:example.com  
X-MAINTNOTE-ACCOUNT:137.035999173  
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415  
X-MAINTNOTE-OBJECT-ID:acme-widgets-as-a-service  
X-MAINTNOTE-IMPACT:NO-IMPACT  
X-MAINTNOTE-STATUS: COMPLETED  
ORGANIZER;CN="Example NOC":mailto:noon@example.com  
END:VEVENT  
END:VCALENDAR
```

### 3.9 Localization Considerations

Localization may be performed as needed by the implementors of this BCOP. English text used in the iCalendar data format [RFC5545] or extensions described in this BCOP may be converted to local languages as required.

### 3.10 Security Considerations

Maintenance notifications are often used as an aid in the planning of operational activities. Incorrect or undelivered maintenance notification information could, if used or required to plan operational activities, result in an operational problem. By preventing the receipt or correct processing of maintenance notifications, modifying or generating false maintenance notifications, an attacker could negatively impact the operations of targeted organizations.

It is a common practice to distribute maintenance notifications via Email. Email messages are not immune to disruption, modification or fabrication by an attacker. To mitigate possible attacks we recommend the use of existing techniques to authenticate maintenance notifications distributed via email. For example, OpenPGP [RFC 3156] could be used to sign a maintenance notification email and include a message integrity check covering the contents of the email. This technique is applied to the contents of the email, which would include the extended iCal data specified in this BCOP. This method, where implemented by both the generator and receiver of a maintenance notification, would prevent the fabrication or modification of maintenance notification emails. Use of OpenPGP to sign messages would only benefit those messages that were delivered to the receiver; it does not address situations where maintenance notifications are disrupted before they can be delivered to the recipient.

As an alternative to pushing maintenance notifications from generators to receivers via email distribution, receivers could pull maintenance notification information from generators via the CalDAV protocol [RFC 4791]. Authenticity and integrity of this communications channel could be provided by accessing CalDAV end points via TLS [RFC 5246]. CalDAV is simply a means to publish iCal formatted content, in this case the extended iCal data specified in this BCOP.

Implementors of this BCOP SHOULD ensure the correctness of all information exchanged when taking automated action based on what they receive. Comparison against out of band sources of information to confirm the correctness of information received MAY be used to detect incorrect information and prevent acting based on it.

### **3.11 Acknowledgements**

### **4. BCOP Conclusion**

[Any concluding remarks pertaining to the BCOP – no length limit] Include results of compatibility testing in this section.