



# Example Practices: CAP Feeds

## Version 1.0

Committee Note 02

04 March 2014

### Specification URIs

#### This version:

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cn02/cap-feeds-v1.0-cn02.doc> (Authoritative)

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cn02/cap-feeds-v1.0-cn02.html>

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cn02/cap-feeds-v1.0-cn02.pdf>

#### Previous version:

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cn01/cap-feeds-v1.0-cn01.doc> (Authoritative)

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cn01/cap-feeds-v1.0-cn01.html>

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cn01/cap-feeds-v1.0-cn01.pdf>

#### Latest version:

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cap-feeds-v1.0.doc> (Authoritative)

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cap-feeds-v1.0.html>

<http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cap-feeds-v1.0.pdf>

### Technical Committee:

[OASIS Emergency Management Adoption TC](#)

#### Chair:

Rex Brooks ([rex.brooks@ncoic.org](mailto:rex.brooks@ncoic.org)), [Network Centric Operations Industry Consortium](#)

#### Editors:

Yu Chen ([yuch@google.com](mailto:yuch@google.com)), [Google Inc.](#)

Anthony Mancuso ([amancuso@google.com](mailto:amancuso@google.com)), [Google Inc.](#)

### Related work:

This document is related to:

This is a Non-Standards Track Work Product. The patent provisions of the OASIS IPR Policy do not apply.

- *Example Practices: CAP Elements Version 1.0.* Edited by Yu Chen and Anthony Mancuso. Latest version. <http://docs.oasis-open.org/emergency-adopt/cap-elements/v1.0/cap-elements-v1.0.html>.
- *Common Alerting Protocol Version 1.2.* Edited by Jacob Westfall. 01 July 2010. OASIS Standard. <http://docs.oasis-open.org/emergency/cap/v1.2/os/CAP-v1.2-os.html>.

### **Abstract:**

This document provides example practices for public dissemination of CAP alerts.

### **Status:**

This document was last revised or approved by the OASIS Emergency Management Adoption TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this document to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "[Send A Comment](#)" button on the Technical Committee's web page at <https://www.oasis-open.org/committees/emergency-adopt/>.

### **Citation format:**

When referencing this document the following citation format should be used:

#### **[CAP-Feeds-v1.0]**

*Example Practices: CAP Feeds Version 1.0.* Edited by Yu Chen and Anthony Mancuso. 04 March 2014. OASIS Committee Note 02. <http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cn02/cap-feeds-v1.0-cn02.html>. Latest version: <http://docs.oasis-open.org/emergency-adopt/cap-feeds/v1.0/cap-feeds-v1.0.html>.

Copyright © OASIS Open 2014. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

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 section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be 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 OASIS or its successors or assigns.

This is a Non-Standards Track Work Product.  
The patent provisions of the OASIS IPR Policy do not apply.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

---

## Table of Contents

1	Introduction .....	5
1.1	References (non-normative).....	5
2	Web Feeds of CAP Alerts .....	6
2.1	General Feed Practices .....	6
2.2	Choice of Web Feed Formats: Atom or RSS.....	7
2.3	Alert Delivery via Feeds .....	7
2.3.1	Atom Feed Delivery .....	8
2.3.1.1	Atom feed elements .....	8
2.3.1.2	Atom entry-to-CAP mapping .....	8
2.3.1.2.1	Examples.....	9
2.3.1.2.2	Example .....	10
2.3.1.3	Embedding CAP elements .....	10
2.3.2	RSS Feed Delivery.....	11
2.3.2.1	RSS channel elements.....	11
2.3.2.2	RSS item-to-CAP mapping .....	11
2.3.2.2.1	Example .....	12
2.4	Feed Security .....	12
2.5	Web Feed Language .....	13
2.6	Updating Alerts.....	13
Appendix A.	Acknowledgments .....	14
Appendix B.	Revision History .....	15

# 1 Introduction

This note provides example practices to distribute CAP alerts publicly over the Internet through web feeds. It was written, in part, in response to [Comments & Questions, Emergency Alerting Policy Workshop](#) (Comment 6), made at the Emergency Alerting Policy Workshop, Montreal, Canada, 1-3 May, 2012, which expressed a “Need for international good/example practices used for alert generation.”

Note: This note is limited to the discussion of distributing CAP alerts through common web-feed mechanisms. It does not discuss other means of publicly distributing CAP alerts, such as the distribution of CAP alerts within EDXL-DE message containers or through the use of social media applications.

## 1.1 References (non-normative)

[CAP-1.2]

*Common Alerting Protocol Version 1.2*. 01 July 2010. OASIS Standard. <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>

## 2 Web Feeds of CAP Alerts

CAP alerts can be made available on the Internet to various clients and user agents (web browsers, mobile applications, crawlers, etc.) through standard web feeds. Once the web feed is established and made public, any member of the public can fetch the alerts referenced in the web feed using widely-available web browsers, alerting applications, web-feed readers, and libraries.

Web feeds of CAP alerts typically link to the XML content of the full CAP alert, and may embed (for display or summary purposes) one or more elements or sub-elements of the CAP alert. For example, the U.S. National Weather Service maintains an [NWS Public Alerts](#) index page that contains feeds to current CAP alerts. Users can subscribe to a feed or click on links to display formatted portions of each feed or alert (the URL of the original, raw XML CAP alert is embedded in each feed entry, and can be displayed from a browser by selecting 'View Source').

Note: Feed clients should not process or filter alerts based on separate CAP alert elements or sub-elements embedded in the feed for display or summary purposes. Feed clients must parse or filter the elements and sub-elements of the original CAP alert, which should be linked in the feed.

### 2.1 General Feed Practices

Alert issuers and feed implementers have wide discretion on how to create and manage alert feeds. The practices discussed in this section are intended to provide practical guidelines to help decrease client confusion and complexity, and improve feed consistency. While issuers are encouraged to adopt these guidelines, different practices will fit best with different issuers, and the ultimate decision on how to structure and manage CAP alert feeds, of course, rests with each implementer.

- Feed servers should support "[Conditional GET](#)" so that Feed clients can obey and use these HTTP values in order to significantly reduce polling bandwidth.
- Valid GeoRSS uses a single geometry. Faced with a CAP Alert with multiple area blocks or polygons/geocodes, an easy feed-generator solution is to create a centroid or other representative point for the entire alert. Points are the best-supported GeoRSS element, and should properly appear on most maps. Another way to properly encompass the alert area is to create a summary polygon that surrounds the entire alert area.
- CAP feeds open to public consumption should be limited to CAP alerts whose <scope> is "Public" and <status> is "Actual."
- The feed <title> should identify the subject matter or type of alert. If the feed contains one alert, it may be possible to construct a useful title from the CAP alert/info/event and alert/msgType elements (e.g., "Industrial Fire, Update, Bridge Closure, Alert"). CAP alert/scope and alert/info/language may also be included (e.g., "NB MASAS Alert Feed: English Public").

- Provide a separate feed of current (including updated) alerts, even if you also provide an "all-alerts" feed that includes alerts for some period after their updated, cancellation, or expiration date for log or archival purposes.
- While it is allowable to include multiple entries in one feed to produce multiple language-specific CAP alert content, it is best to produce one feed for language-specific CAP alert content. Doing this can reduce filtering work for feed clients.
- If possible, when an alert has separate <info> blocks in the same language for an event with different alert content (for example, a storm with areas having different severity), provide separate feed entries (Atom <entry>s or RSS <channels>s) that align with separate <info> blocks (a Severe Storm Warning entry and a Moderate Storm Warning entry in the feed ). Note: this recommendation makes life easier for the LMD (Last Mile Distributor) of the alert, but more difficult for the feed originator. Hence, this is a qualified recommendation, and should be implemented if the benefit outweighs the effort.

## 2.2 Choice of Web Feed Formats: Atom or RSS

Two common web feed formats are Atom and RSS. CAP alerts can be published using either format. When a CAP feed may have life-critical content, the comparative simplicity, prevalence, and stability of each format are important considerations:

**Simplicity** - Simplicity is an obvious virtue in technology, provided that the technology is not so simplistic that it lacks essential features. RSS is a simpler (less expressive) format than Atom, yet RSS does have all features essential for CAP alerting.

**Prevalence** - Prevalence is important in the CAP context because it is more likely that a more prevalent technology will be supported by the target audiences of the alert. RSS is the more prevalent web-feed format, estimated at 80% versus 20% for Atom.

**Stability** - Stability of a technology is important in the CAP context because the more stable technology is more likely to be implemented completely and correctly. Atom is an evolving technology, while the RSS specification (RSS version 2.0) is stable (it has been frozen for many years).

Note that the above comparative points are made in the context of a simple feed solution for a life-threatening alert (where RSS may be the best feed to distribute quickly to wide audience). Generally, however, Atom offers better support for CAP elements, and is the preferred feed to use to publish CAP content.

## 2.3 Alert Delivery via Feeds

While some feeds embed elements of a CAP alert for display purposes (using XSLT), note that only a separate XML file with "alert" as its top-level element that conforms in all respects with a

standard CAP specification is a "CAP message." Hence, the recommended practice is to always include a link to the full, conformant CAP alert in web feeds.

### 2.3.1 Atom Feed Delivery

Atom encompasses a pair of related web standards - an [XML-based syndication format](#) for structuring data for web feeds, and an [HTTP-based publishing protocol](#) for creating and updating those web feeds.

#### 2.3.1.1 Atom feed elements

- Required feed sub-elements. An Atom feed must have the following sub-elements: <title>, <updated>, <id>, and <author> (the feed <author> sub-element is not required if each <entry> in the feed contains an <author> sub-element). It is good practice to fill in the <updated> sub-element with latest <entry>:<published> value in the feed (this value should reflect the <sent> value of the most recent CAP alert included in the feed).
- Required entry sub-elements. An Atom feed can contain one or more <entry> elements. Each <entry> should correspond to a particular CAP alert published by the altering source. Each <entry> must have the following sub elements: <title>, <updated>. <id>.
- Optional entry sub-elements. The following optional sub-elements should also be included in the <entry> to describe the CAP alert: <link>, <published>, <rights>, <source>, <summary>. Note that a <link> sub-element (with a *rel* attribute of "alternate") is required if the <entry> does not contain a <content> sub-element. Also, a <summary> sub-element is required if the <content> sub-element is empty (contains a *src* attribute).

#### 2.3.1.2 Atom entry-to-CAP mapping

Some CAP alert elements correspond (loosely in some cases) to counterpart elements in web feeds. Hence, in some cases, web feed generators may (programmatically or manually) populate one or more feed elements with CAP element values. We list some possible correspondences between CAP and Atom feed elements below. Use these correspondences with caution, however, since, as noted below, there may be subtle differences between the meaning or intent of a CAP element and a similar feed element. Also note that a CAP alert may have multiple <info> blocks for the same alert (for example, to provide different alert content for different locations affected by the alert). If the feed has only one entry for each alert, there may not be a satisfactory (easily parsed) mapping between multiple info block content and a feed entry element.

Note: Feed clients should not process or filter alerts based on separate CAP alert elements or sub-elements embedded in the feed for display or summary purposes. Feed clients must parse or filter the elements and sub-elements of the original CAP alert, which should be linked in the feed.

- entry/title = cap:alert:info:headline



- entry/id = cap:alert:identifier  
Caution: Using the cap:alert:identifier as the entry/id can lead to duplicate (hence invalid) feed entry ids if more than one entry in the feed contains content from the same CAP alert. To better ensure uniqueness among entry ids, create a [tag URI](#) or a [UUID](#) for each entry id.
- entry/category = cap:alert:info:category
- entry/summary = cap:alert:info:headline or excerpt from cap:alert:info:description
- entry/published = cap:alert:sent  
Note: The CAP sent element is specified as the date and time of the origination of the CAP alert. You may wish to use the date and time of the publication of the feed instead, since this is the expected value for <published>.
- entry/author/name = cap:alert:info:senderName
- entry/author/email = cap:sender (sender's email, if provided)  
Note: CAP sender may not include the sender's email, and even if it does, an individual sender may not anticipate distribution of the email address in a widely-disseminated public feed.
- entry/link = (URL of full CAP alert)  
Linking to where the full CAP alert is hosted is recommended. Note the following example practices when linking to the CAP alert:
  - Use an absolute (not a relative) URL.
  - The alert page must exist when the alert link is added to the feed. Otherwise, feed clients that try to load the alert will encounter errors at load time.
  - The link to the CAP alert must be correctly identified (type="application/cap+xml"). Similarly, if other links are provided to content, they must be appropriately associated with the correct MIME type and other attributes.

#### 2.3.1.2.1 Examples

##### [U.S. Geological Survey Earthquake feed](#)

The feed links to the original XML CAP alert and to an HTML version of the alert:

```
<entry>
<id>urn:earthquake-usgs-gov:us:c000f1r5</id>
<title>M 6.3, Santa Cruz Islands</title>
<updated>2013-02-06T00:07:22Z</updated>

<link rel="related" type="application/cap+xml"
href="http://earthquake.usgs.gov/earthquakes/catalogs/cap/usc000f1r5"/>
```

```
<link rel="alternate" type="text/html"
href="http://earthquake.usgs.gov/earthquakes/recenteqsww/Quakes/usc000f1r5.php"/>
```

```
...
</entry>
```

- entry/content: = (embedded cap <alert>)  
You can include the entire CAP <alert> inside an Atom's feed entry's content element (<content type = "cap+xml">). This method allows a feed client to filter the entries it fetches based on the directly-embedded full alert, which may eliminate the need for extra HTTP requests by a feed client. For example, a feed-reader application may allow a user to only fetch alerts that affects a designated town.

Caution: This method may be appropriate for streaming clients or for archival purposes. And while it also may be useful to allow feed clients to eliminate the need for additional HTTP requests (by filtering directly from the feed content), it can create feed "bloat," particularly when the feed includes a large CAP alert with multiple info blocks or multiple entries with multiple CAP alerts. Hence, to avoid feed bloat, simply provide a link in the entry to the CAP message (see "entry/link" above).

#### 2.3.1.2.2 Example

##### [NOAA \(National Oceanic and Atmospheric Administration\)](#)

Links to CAP alerts instead of embedding the alerts in the feed, which keeps the size of this national feed manageable and more easily downloadable by clients.

#### 2.3.1.3 Embedding CAP elements

Some feeds (such as the [U.S. National Weather Service Public Alerts](#)) directly embed one or more CAP alert elements in the feed, use the "cap" namespace followed by the CAP element name (<cap:element\_name>):

```
<cap:event>Special Weather Statement</cap:event>
<cap:effective>2013-02-05T20:35:00-08:00</cap:effective>
<cap:expires>2013-02-06T10:30:00-08:00</cap:expires>
<cap:status>Actual</cap:status>
<cap:msgType>Alert</cap:msgType>
<cap:category>Met</cap:category>
<cap:urgency>Expected</cap:urgency>
<cap:severity>Minor</cap:severity>
<cap:certainty>Observed</cap:certainty>
<cap:area>
  <cap:areaDesc>Apple and Lucerne Valleys; Coachella Valley;
Orange County Coastal Areas;
  </cap:areaDesc>
  <cap:geocode>
    <valueName>FIPS6</valueName>
    <value>006059 006065 006071 006073</value>
    <valueName>UGC</valueName>
```

```
<value>CAZ042 CAZ043 CAZ048 CAZ050 CAZ055 CAZ056 CAZ057  
CAZ058 CAZ060</value>  
</cap:geocode>  
<cap:area>
```

Caution: While embedding CAP alert elements in a feed for display in web browsers can be convenient to offer human viewers a summary of selected CAP elements, this practice is normally not advised for the following reasons:

- Selecting some CAP elements can cause feed consumers to assume that the displayed elements represent the entire CAP alert. This misunderstanding can lead to inadequate alert responses as well as potential legal liability issues.
- Using a "cap" prefix for embedding each element does not represent the fully qualified CAP element name (for example, "cap:event" above is represented in the CAP alert as "alert:info:event"). Hence, there can be element name collisions when elements from a CAP alert with multiple <info> blocks are directly embedded in a feed.

## 2.3.2 RSS Feed Delivery

### 2.3.2.1 RSS channel elements

Channel elements. There is a single <channel> defined in an RSS news feed. There are three required sub-elements about the RSS <channel> itself: <title>, <link>, and <description>. There are optional sub-elements you can use to further describe the channel, such as <language>, <pubDate>, and <lastBuildDate>.

Item elements. Each RSS news< item> is another element within the news <channel>. That RSS item should correspond to a particular CAP alert published by the altering source. Each RSS <item> must have at least one of the <title> or <description> sub-elements. Although this is the only required sub-element, there are additional (optional) sub-elements that can be used to characterize a CAP alert, including <guid>, <category>, <pubDate>, <author>, and <enclosure>.

### 2.3.2.2 RSS item-to-CAP mapping

Some CAP alert elements correspond (loosely in some cases) to counterpart elements in web feeds. Hence, in some cases, web feed generators may (programmatically or manually) populate one or more feed elements with CAP element values. We list some possible correspondences between CAP and RSS feed elements below. Use these correspondences with caution, however, since, as noted below, there may be subtle differences between the meaning or intent of a CAP element and a similar feed element. Also note that a CAP alert may have multiple <info> blocks for the same alert (for example, to provide different alert content for different locations affected by the alert). If the feed has only one item for each alert, there may not be a satisfactory (easily parsed) mapping between multiple info block content and a feed item element.

Note: Feed clients should not process or filter alerts based on separate CAP alert elements or sub-elements embedded in the feed for display or summary purposes. Feed clients must parse or filter the elements and sub-elements of the original CAP alert, which should be linked in the feed.

- item/title = cap:alert:info:headline
- item/guid = cap:alert:identifier  
Caution: Using the cap:alert:identifier as the item/guid can lead to duplicate (hence invalid) feed item guides if more than one item in the feed contains content from the same CAP alert. To better ensure uniqueness among item guides, create a [tag URI](#) or a [UUID](#) for each item guid.  
Also Note. If you provide both Atom and RSS feeds to the same CAP alert, you should use the same id for both feeds (Atom entry/id = RSS item/guid for the same CAP alert).
- item/category = cap:alert:info:category
- item/pubDate = cap:alert:sent  
Note: The CAP sent element is specified as the date and time of the origination of the CAP alert. You may wish to use the date and time of the publication of the feed instead, since this is the expected <pubDate> value.
- item/description = cap:alert:info:description
- item/author = cap:alert:info:senderName
- item/enclosure= (URL of full cap alert message)  
Linking to where the full CAP alert is hosted is recommended. Using the item/enclosure element is preferable to using the item/link element since <enclosure> can include the MIME type.  
Note the following example practices when linking to the CAP alert:
  - Use an absolute (not a relative) URL.
  - The alert page must exist when the alert link is added to the feed. Otherwise, feed clients that try to load the alert will encounter errors at load time.
  - The link to the CAP alert should be identified by mimeType (type="application/cap+xml").  

```
<link rel="related" type="application/cap+xml"
href="http://earthquake.usgs.gov/earthquakes/catalogs/cap/u
sc000f1r5"/>
```

#### 2.3.2.2.1 Example

[NTWC \(National Tsunami Warning Center\)](#)

Contains CAP and non-CAP event pages. CAP file distinguished by <link rel="related" type="application/cap+xml"

## 2.4 Feed Security

To ensure security and authenticity, the web feed can be served:

- over HTTPS

- over HTTP if each message (CAP <alert>) is digitally signed or if the feed itself is digitally signed

## 2.5 Web Feed Language

Often, authorities issue alerts to multi-lingual populations. There are three methods for handling multilingual content in a CAP feed. Choose the method that's most appropriate for your situation.

- 1) Include multilingual content in one CAP alert message. The CAP specification allows an alert to be provided in multiple languages through the use of multiple <info> blocks. One feed entry can link to the entire CAP alert (but feed consumers will have to parse out the language-specific <info> blocks).
- 2) Issue a separate CAP alert message for each language. Link to those alerts from language-specified entries or links in the same web feed. In Atom feeds, a single <entry> allows multiple <links>, each with its own hreflang attribute.
- 3) As in 2) above, issue a separate CAP alert message for each language. Create separate web feeds for each language. In each feed, link to the CAP alert message that is in the same language as the feed. Note that this method may require the least amount of work and filtering by Last Mile Distributors that parse the feed.

## 2.6 Updating Alerts

- Whenever a CAP alert is changed, a new Alert (a new XML file) is issued. When this occurs, a new Atom feed <entry> or RSS feed <item> should be created that links to the new alert. A feed should not edit information in an already-issued CAP alert. This practice helps ensure that there's a trailing record of changes to an alert.
- After an appropriate delay (to allow polling clients to sync up), it is good practice to remove a canceled, updated, or expired alert from a feed. One practice is to keep a canceled, expired, or updated alert message up for 24 or 48 hours after its cancellation, update, or expiration. Removal of outdated feeds helps prevents the feed from growing in size indefinitely. Note that feed readers often allow you to view deleted entries (to view previous alerts).

## Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Doug Allport	Canadian Association for Public Alerting and Notification (CAPAN)
Art Botterell	Individual
Rex Brooks	Network Centric Operations Industry Consortium
Elliot Christian	Individual
Phil Coakley	Google
Steve Hakusa	Google
Gary Ham	Individual
Elysa Jones	Individual
Camille Osterloh	Individual
Norm Paulsen	Environment Canada
Ezra Resnick	Google
Jacob Westfall	Individual

---

## Appendix B. Revision History

Revision	Date	Editor	Changes Made
1.0 wd01 - 08	11/5/12 - 12/21/12	Tony Mancuso	Initial Drafts
1.0 wd02 - 16	01/07/13 - 04/02/13	Tony Mancuso	Deleted Cap element material and renamed doc for feeds-only content. Edited text in response to comments from Elliot Christian, Jacob Westfall, Yu Chen, and Steve Hakusa