CAP Index Format Proposal

Draft: July 30, 2007
Author: Jacob Westfall

The following proposal outlines best practices for creating a Common Alerting Protocol (CAP) index in RSS 2.0 and Atom 1.0. This is an initial draft created to gather feedback from other CAP index users and creators.

The CAP standard defines how individual CAP messages are created, but does not provide a method for indexing or grouping these messages. An index is required so that CAP clients can discover newly posted messages and the required information to download these new messages. Using multiple CAP info segments to create such an index is not valid as these segments refer to distinct incidents and should be created as individual messages. Previous index formats have included text listings of new message links, and a number of different implementations using RSS and Atom as special CAP only feeds.

This proposal seeks to establish a standard method of creating CAP message indexes in either RSS or Atom. This allows for a creator's current feeds to be easily “CAP enabled” through the addition of CAP specific elements. No special CAP only feed is required allowing both CAP enabled clients and regular RSS/Atom feed readers to use the same feed. This reduces duplication, enhances the capabilities of CAP enabled clients, and will hopefully spur the uptake of CAP by other RSS/Atom feed readers.

A key component of this index format is the registration of a CAP mime-type. At this time the proposed mime-type is the following,

application/common-alerting-protocol+xml

Also some additional guidelines to follow: The file extension .cap should be used for CAP messages. (*ex. 12312312.cap*). Atom entry ids should use tagURIs. RSS feeds should use a guid for each entry and may use the message url as a permalink, or better yet a tagURI as a non-permalink. A default value for the RSS enclosure length can also be used. Stylesheets can be used for both CAP messages and CAP feeds, but creators should not assume that all readers will use them.

There are 3 parts to creating a CAP message index using RSS/Atom. Only part 1 is required, the remainder are optional.

1. Each entry in the index should provide a link to the full CAP message. This allows CAP enabled clients to discover any new messages and download them. Using the default link type for either RSS or Atom is not suitable however, since many sites may want normal feed readers to use this default link for an HTML page instead of a CAP message. Some sites overcome this by including a stylesheet with their CAP messages, however not all sites do and it should not be the expectation. So a CAP specific link type should be used. For both RSS and Atom this is accomplished through the use of a mime-type for CAP messages. See the following examples,

Atom <link rel="enclosure" type="application/common-alerting-protocol+xml" href="http://message.cap" />

RSS <enclosure url="http://message.cap" length="500" type="application/common-alerting-protocol+xml" />

2. Elements from the CAP message itself can be added to each index entry. These elements can be used by CAP enabled clients to filter and act upon the index entries prior to downloading the full CAP messages. Some sites also use a stylesheet with their index, which in turn uses these elements for presentation purposes. The use and selection of these elements is left up to the feed creator. Future discussion may develop a list of commonly used elements. The CAP namespace and cap prefix are to be used for these elements and applies equally to RSS and Atom. See the following example,

RSS <rss xmlns:cap="urn:oasis:names:tc:emergency:cap:1.1" version="2.0">
<channel>
<title>Sample CAP RSS Feed</title>
<link>http://feed.rss</link>
<description>Sample feed</description>
<pubDate>Mon, 30 Jul 2007 12:00:00 EST</pubDate>
<item>
<title>Sample CAP message</title>
<description>Sample message</description>
<link>http://message.html</link>
<enclosure url="http://message.cap" length="500" type="application/common-alerting-protocol+xml" />
<pubDate>Mon, 30 Jul 2007 12:00:00 EST</pubDate>
<guid isPermaLink="false">tag:test.com,2007-07-30:/cap/12345</guid>

```
    <cap:identifier>12345</cap:identifier>
    <cap:sender>test@test.com</cap:sender>
    <cap:sent>2007-07-30T12:00:00-05:00</cap:sent>
  </item>
</channel>
</rss>
```

3. The majority of feeds will be accessible online and download the index and the CAP messages individually. However some feeds may need to be accessible in an offline format, or be used for archival purposes. In these cases a single file index that contains both the index entries and the CAP messages is required. RSS is not suitable for this purpose since it does not include an acceptable method for denoting content types. Atom is the only recommended format for creating single file indexes. While it is possible to include the whole CAP message in an RSS entry using the method outlined in part 2, this will cause confusion for those clients expecting only CAP elements and instead receiving a full CAP message. Further discussion could develop an alternative method for RSS. Each entry's CAP message in the index should be included in the content element with the type being the CAP mime type. The elements from part 1 and part 2 should still be added if the CAP messages are also available online. See the following example,

Atom
```
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:cap="urn:oasis:names:tc:emergency:cap:1.1>
  <id>http://index.atom</id>
  <title>Sample Index</title>
  <updated>2007-07-30T12:00:00-05:00</updated>
  <entry>
    <id>tag:test.com,2007-07-30:/cap/12345</id>
    <title>Sample message</title>
    <updated>2007-07-30T12:00:00-05:00</updated>
    <link rel="alternate" href="http://message.html" />
    <link rel="enclosure" type="application/common-alerting-
protocol+xml" href="http://message.cap" />
    <cap:identifier>12345</cap:identifier>
    <cap:sender>test@test.com</cap:sender>
    <cap:sent>2007-07-30T12:00:00-05:00</cap:sent>
    <content type="application/common-alerting-protocol+xml">
      <alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
        <identifier>12345</identifier> ... and so on ....
    </content>
  </entry>
</feed>
```