Network Working Group
Request for Comments: 5005
Category: Standards Track

M. Nottingham

September 2007

Feed Paging and Archiving

Status of This Memo

Abstract

This specification defines three types of syndicated Web feeds that enable publication of entries across one or more feed documents. This includes "paged" feeds for piecemeal access, "archived" feeds that allow reconstruction of the feed's contents, and feeds that are explicitly "complete".

Table of Contents

1. Introduction

Syndicated Web feeds (using formats such as Atom [3]) are often split into multiple documents to save bandwidth, allow "sliding window" access, or for other purposes.

This specification formalizes two types of feeds that can span one or more feed documents; "paged" feeds and "archived" feeds. Additionally, it defines "complete" feeds to cover the case when a single feed document explicitly represents all of the feed's entries.

Each has different properties and trade-offs:

- Complete feeds contain the entire set of entries in one document, and can be useful when it isn't desirable to "remember" previously-seen entries.
- Paged feeds split the entries among multiple temporary documents. This can be useful when entries in the feed are not long-lived or stable, and the client needs to access an arbitrary portion of them, usually in close succession.
- Archived feeds split the entries among multiple permanent documents and can be useful when entries are long-lived, and it is important for clients to see every one.

The semantics of a feed that combines these types is undefined by this specification.

Although they refer to Atom normatively, the mechanisms described herein can be used with similar syndication formats; see Appendix B for one such use.

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

This specification uses XML Namespaces [4] to uniquely identify XML element names. It uses the following namespace prefix for the indicated namespace URI;

"fh": "http://purl.org/syndication/history/1.0"

1.2. Terminology

In this specification, "feed document" refers to an Atom Feed Document or similar syndication instance document. It may contain any number of entries, and may or may not be a complete representation of the logical feed.

A "logical feed" is the complete set of entries associated with a feed (as contrasted with a feed document, which may contain a subset of entries).

"Head section" refers to a document's feed-wide metadata container; e.g., the child elements of the atom:feed element in an Atom Feed Document.

This specification uses terms from the XML Infoset [5]. However, this specification uses a shorthand; the phrase "Information Item" is omitted when naming Element Information Items. Therefore, when this specification uses the term "element," it is referring to an Element Information Item in Infoset terms.

This specification also uses Atom link relations to identify different types of links; see the Atom specification [3] for information about their syntax, and the IANA link relation registry for more information about specific values.

Note that URI references in link relation values may be relative, and when they are used they must be absolutised, as described in Section 5.1 of [2].

2. Complete Feeds

A complete feed is a feed document that contains all of the entries of a logical feed; any entry not actually in the feed document SHOULD NOT be considered part of that feed.

For example, a feed that represents a ranking that varies over time (such as "Top Twenty Records" or "Most Popular Items") should not have newer entries displayed alongside older ones. By marking this feed as complete, old entries are discarded when it is refreshed.

The fh:complete element, when present in a feed's head section, indicates that the feed document it occurs in is a complete representation of the logical feed's entries. It is an empty element; this specification does not define any content for it.

Example: Atom-formatted Complete Feed

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
 xmlns:fh="http://purl.org/syndication/history/1.0">
 <title>NetMovies Queue</title>
 <subtitle>The DVDs you'll receive next.</subtitle>
 <link href="http://example.org/"/>
 <fh:complete/>
 <link rel="self"
  href="http://netmovies.example.org/jdoe/queue/index.atom"/>
 <updated>2003-12-13T18:30:02Z</updated>
 <author>
   <name>John Doe</name>
 </author>
 <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
 <entry>
   <title>Casablanca</title>
   <link href="http://netmovies.example.org/movies/Casablanca"/>
   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
   <updated>2003-12-13T18:30:02Z</updated>
   <summary>Here's looking at you, kid...</summary>
 </entry>
</feed>
```

This specification does not address duplicate entries in complete feeds.

3. Paged Feeds

A paged feed is a set of linked feed documents that together contain the entries of a logical feed, without any guarantees about the stability of each document's contents.

Paged feeds are lossy; that is, it is not possible to guarantee that clients will be able to reconstruct the contents of the logical feed at a particular time. Entries may be added or changed as the pages of the feed are accessed, without the client becoming aware of them.

Therefore, clients SHOULD NOT present paged feeds as coherent or complete, or make assumptions to that effect.

Paged feeds can be useful when the number of entries is very large, infinite, or indeterminate. Clients can "page" through the feed, only accessing a subset of the feed's entries as necessary.

For example, a search engine might make query results available as a paged feed, so that queries with very large result sets do not overwhelm the server, the network, or the client.

The feed documents in a paged feed are tied together with the following link relations:

- "first" - A URI that refers to the furthest preceding document in a series of documents.
- "last" - A URI that refers to the furthest following document in a series of documents.
- "previous" - A URI that refers to the immediately preceding document in a series of documents.
- "next" - A URI that refers to the immediately following document in a series of documents.

Paged feed documents MUST have at least one of these link relations present, and should contain as many as practical and applicable.

Example: Atom-formatted Paged Feed

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
 <title>Example Feed</title>
 <link href="http://example.org/"/>
 <link rel="self" href="http://example.org/index.atom"/>
 <link rel="next" href="http://example.org/index.atom?page=2"/>
 <updated>2003-12-13T18:30:02Z</updated>
 <author>
   <name>John Doe</name>
 </author>
 <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
 <entry>
   <title>Atom-Powered Robots Run Amok</title>
   <link href="http://example.org/2003/12/13/atom03"/>
   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
   <updated>2003-12-13T18:30:02Z</updated>
   <summary>Some text.</summary>
 </entry>
</feed>
```

This specification does not address duplicate entries in paged feeds.

4. Archived Feeds

An archived feed is a set of feed documents that can be combined to accurately reconstruct the entries of a logical feed.

Unlike paged feeds, archived feeds enable clients to do this without losing entries. This is achieved by publishing a single subscription document and (potentially) many archive documents.

A subscription document is a feed document that always contains the most recently added or changed entries available in the logical feed.

Archive documents are feed documents that contain less recent entries in the feed. The set of entries contained in an archive document published at a particular URI SHOULD NOT change over time. Likewise, the URI for a particular archive document SHOULD NOT change over time.

The following link relations are used to tie subscription and archived feeds together:

- "prev-archive" - A URI that refers to the immediately preceding archive document.
- "next-archive" - A URI that refers to the immediately following archive document.
- "current" - A URI that, when dereferenced, returns a feed document containing the most recent entries in the feed.

Subscription documents and archive documents MUST have a "prev-archive" link relation, unless there are no preceding archives available. Archive documents SHOULD also have a "next-archive" link relation, unless there are no following archives available.

Archive documents SHOULD indicate their associated subscription documents using the "current" link relation.

Archive documents SHOULD also contain an fh:archive element in their head sections to indicate that they are archives. fh:archive is an empty element; this specification does not define any content for it.

Example: Atom-formatted Subscription Document

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
 <title>Example Feed</title>
 <link href="http://example.org/"/>
 <link rel="self" href="http://example.org/index.atom"/>
 <link rel="prev-archive"
  href="http://example.org/2003/11/index.atom"/>
 <updated>2003-12-13T18:30:02Z</updated>
 <author>
   <name>John Doe</name>
 </author>
 <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
 <entry>
   <title>Atom-Powered Robots Run Amok</title>
   <link href="http://example.org/2003/12/13/atom03"/>
   <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
   <updated>2003-12-13T18:30:02Z</updated>
   <summary>Some text.</summary>
 </entry>
</feed>
```

Example: Atom-formatted Archive Document

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
 xmlns:fh="http://purl.org/syndication/history/1.0">
 <title>Example Feed</title>
 <link rel="current" href="http://example.org/index.atom"/>
 <link rel="self" href="http://example.org/2003/11/index.atom"/>
 <fh:archive/>
 <link rel="prev-archive"
  href="http://example.org/2003/10/index.atom"/>
 <updated>2003-11-24T12:00:00Z</updated>
 <author>
   <name>John Doe</name>
 </author>
 <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
 <entry>
   <title>Atom-Powered Robots Scheduled To Run Amok</title>
   <link href="http://example.org/2003/11/24/robots_coming"/>
   <id>urn:uuid:cdef5c6d5-gff8-4ebb-assa-80dwe44efkjo</id>
   <updated>2003-11-24T12:00:00Z</updated>
   <summary>Some text from an old, different entry.</summary>
 </entry>
</feed>
```

In this example, the feed archives are split into monthly chunks, and the subscription document points to the most recent complete archive "http://example.org/2003/11/index.atom" using the "prev-archive" relation. That document, in turn points to the previous archive "http://example.org/2003/10/index.atom", and so on. Note that the "2003/11" archive does not have a "next-archive" relation, because it is the most recent complete archive; although another archive ("2003/12") may be under construction, it would be an error to link to it before completion.

4.1. Publishing Archived Feeds

The requirement that archive documents be stable allows clients to safely assume that if they have retrieved one in the past, it will not meaningfully change in the future. As a result, if an archive document's contents are changed, some clients may not become aware of the changes.

Therefore, if a publisher requires a change to be visible to all users (e.g., correcting factual errors), they should consider publishing the revised entry in the subscription document, in addition to (or instead of) the appropriate archive document. Conversely, unimportant changes (e.g., spelling corrections) might be only effected in archive documents.

Publishers SHOULD construct their feed documents in such a way as to make duplicate removal unambiguous (see Section 4.2).

Publishers are not required to make all archive documents available; they may refuse to serve (e.g., with HTTP status code 403 or 410) or be unable to serve (e.g., with HTTP status code 404) an archive document.

4.2. Consuming Archived Feeds

Typically, clients will "subscribe" to an archived feed by polling the subscription document for recent changes. If a URI contained in the prev-archive link relation has not been processed in the past, the client can "catch up" with any missed entries by dereferencing it and adding the contained entries to the logical feed. This process should be repeated recursively until the client encounters a prev-archive link relation that has been processed (the end of the archive is indicated by a missing prev-archive link relation) or an error is encountered.

If duplicate entries are found, clients SHOULD consider only the most recently updated entry to be part of the logical feed. If duplicate entries have the same update time-stamp, or no time-stamps are available, the entry sourced from the most recently updated feed document SHOULD replace all other duplicates of that entry.

In Atom-formatted archived feeds, two entries are duplicates if they have the same atom:id element. The update time of an entry is determined by its atom:updated element, and likewise the update time of a feed document is determined by its feed-level atom:updated element.

Clients SHOULD warn users when they are not able to reconstruct the entire logical feed (e.g., by alerting the user that an archive document is unavailable, or displaying pseudo-entries that inform the user that some entries may be missing).

5. IANA Considerations

This specification defines the following new relations that have been added to the Link Relations registry:

- Attribute Value: prev-archive
- Description: A URI that refers to the immediately preceding archive document.
- Security considerations: See [RFC5005]
- Attribute Value: next-archive
- Description: A URI that refers to the immediately following archive document.
- Expected display characteristics: none
- Security considerations: See [RFC5005]

Additionally, the "previous," "next", and "current" link relations should be updated to refer to this document.

6. Security Considerations

Feeds using this mechanism have the same security considerations as Atom [3]. Encryption and authentication security services can be obtained by encrypting and/or signing the feed, as described in [3], and may also be obtained through channel-based mechanisms (e.g., TLS [9], HTTP authentication [7]) and/or transport (e.g., IPsec [8]).

Feeds using these mechanisms could be crafted in such a way as to cause a client to initiate excessive (or even an unending sequence of) network requests, causing denial of service (either to the client, the target server, and/or intervening networks). Clients can mitigate this risk by requiring user intervention after a certain number of requests, or by limiting requests either according to a hard limit, or with heuristics. Servers can mitigate this risk by denying requests that they consider abusive (e.g., by closing the connection or generating an error).

Clients should be mindful of resource limits when storing feed documents. To reiterate, they are not required to always store or reconstruct the feed when conforming to this specification; they only need to inform the user when the reconstructed feed is not complete.

This specification does not define what it means when a logical feed's component feed documents have different security mechanisms applied.

7. References

7.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[3] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, December 2005.

[4] Bray, T., Hollander, D., and A. Layman, "Namespaces in XML", World Wide Web Consortium FirstEdition REC-xml-names-19990114, January 1999, <http://www.w3.org/TR/1999/REC-xml-names-19990114>.

[5] Tobin, R. and J. Cowan, "XML Information Set (Second Edition)", World Wide Web Consortium Recommendation REC-xml-infoset-20040204, February 2004, <http://www.w3.org/TR/2004/REC-xml-infoset-20040204>.

7.2. Informative References

[6] Winer, D., "RSS 2.0 Specification", 2005, <http://www.rssboard.org/rss-specification>.

[7] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[8] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[9] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

Appendix A. Acknowledgements

The author would like to thank the following people for their contributions, comments, and help: Danny Ayers, Thomas Broyer, Lisa Dusseault, Stefan Eissing, David Hall, Bill de Hora, Vidya Narayanan, Aristotle Pagaltzis, John Panzer, Dave Pawson, Garrett Rooney, Robert Sayre, James Snell, Henry Story, and Franklin Tse.

Any errors herein remain the author's, not theirs.

Appendix B. Use in RSS 2.0

As previously noted, while this specification's extensions are described in terms of the Atom feed format, they are also useful in similar formats. This informative appendix demonstrates how they can be used in an RSS 2.0-formatted [6] feed.

In RSS 2.0-formatted feeds, two entries are duplicates if they have the same guid element. The update time of an entry is not defined by RSS 2.0, but the feed-level update time can be determined by the lastBuildDate element, if present.

RSS 2.0-formatted Complete Feed

```
<?xml version="1.0"?>
<rss version="2.0"
 xmlns:fh="http://purl.org/syndication/history/1.0">
 <channel>
  <title>NetMovies Queue</title>
  <link>http://netmovies.example.org/</link>
  <description>The DVDs you'll receive next.</description>
  <fh:complete/>
  <item>
   <title>Casablanca</title>
   <link>http://netmovies.example.org/movies/Casablanca</link>
   <description>Here's looking at you, kid...
   </description>
   <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
   <guid isPermaLink="false"
   >urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</guid>
  </item>
 </channel>
</rss>
```

RSS 2.0-formatted Paged Feed

```
<?xml version="1.0"?>
<rss version="2.0"
 xmlns:atom="http://www.w3.org/2005/Atom">
 <channel>
  <title>Liftoff News</title>
  <link>http://liftoff.example.net/</link>
  <description>Liftoff to Space Exploration.</description>
  <atom:link rel="next"
   href="http://liftof.example.net/index.rss?page=2"/>
  <item>
   <title>Star City</title>
   <link>http://liftoff.example.net/2003/06/news-starcity</link>
   <description>How do Americans get ready to work with Russians
   aboard the International Space Station? They take a crash course
   in culture, language and protocol at Russia's Star City.
   </description>
   <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
   <guid>http://liftoff.example.net/2003/06/03/starcity</guid>
  </item>
 </channel>
</rss>
```

RSS 2.0-formatted Subscription Document

```
<?xml version="1.0"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
 <channel>
  <title>Liftoff News</title>
  <link>http://liftoff.example.net/</link>
  <description>Liftoff to Space Exploration.</description>
  <atom:link rel="prev-archive"
   href="http://liftoff.example.net/2003/05/index.rss"/>

  <item>
   <title>Star City</title>
   <link>http://liftoff.example.net/2003/06/news-starcity</link>
   <description>How do Americans get ready to work with Russians
   aboard the International Space Station? They take a crash course
   in culture, language and protocol at Russia's Star City.
   </description>
   <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
   <guid>http://liftoff.example.net/2003/06/03/starcity</guid>
  </item>
 </channel>
</rss>
```

RSS 2.0-formatted Archive Document

```
<?xml version="1.0"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom"
 xmlns:fh="http://purl.org/syndication/history/1.0">
 <channel>
  <title>Liftoff News</title>
  <link>http://liftoff.example.net/</link>
  <description>Liftoff to Space Exploration.</description>
  <lastBuildDate>Fri, 30 May 2003 11:06:42 GMT</lastBuildDate>
  <fh:archive/>
  <atom:link rel="current"
   href="http://liftoff.example.net/index.rss"/>
  <atom:link rel="prev-archive"
   href="http://liftoff.example.net/2003/04/index.rss"/>

  <item>
   <title>Upcoming Eclipse</title>
   <link>http://liftoff.example.net/2003/05/30/eclipse</link>
   <description>Sky watchers in Europe, Asia, and parts of
   Alaska and Canada will experience a partial eclipse of the Sun
   on Saturday, May 31st.</description>
   <pubDate>Fri, 30 May 2003 11:06:42 GMT</pubDate>
   <guid>http://liftoff.example.net/2003/05/30/eclipse</guid>
  </item>
  <item>
   <title>The Engine That Does More</title>
   <link>http://liftoff.example.net/2003/05/27/vasmir</link>
   <description>Before man travels to Mars, NASA hopes to
   design new engines that will let us fly through the Solar
   System more quickly. The proposed VASIMR engine would do
   that.</description>
   <pubDate>Tue, 27 May 2003 08:37:32 GMT</pubDate>
   <guid>http://liftoff.example.net/2003/05/27/vasmir</guid>
  </item>
 </channel>
</rss>
```

Index

Author's Address

Mark Nottingham
Email: mnot@pobox.com
URI: http://www.mnot.net/

Full Copyright Statement

Intellectual Property

[1] http://www.ietf.org/ipr
[2] mailto:ietf-ipr@ietf.org