

Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 11, 2007

M. Nottingham
Yahoo! Inc.
May 10, 2007

HTTP Cache Channels
draft-nottingham-http-cache-channels-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on November 11, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This specification defines "cache channels" to propagate out-of-band events from HTTP origin servers (or their delegates) to subscribing caches. It also defines an event payload that gives finer-grained control over cache freshness.

Table of Contents

| | | |
|-----------------------------|--|--------------------|
| 1. | Introduction | 3 |
| 2. | Notational Conventions | 3 |
| 3. | Cache Channels | 4 |
| 3.1. | Channel Subscription | 5 |
| 3.2. | Event Application | 5 |
| 3.3. | Atom Cache Channels | 6 |
| 3.3.1. | The 'cc:precision' Feed Extension | 7 |
| 3.3.2. | The 'cc:lifetime' Feed Extension | 7 |
| 3.3.3. | Example | 8 |
| 4. | Managing Freshness with Cache Channels | 8 |
| 4.1. | The 'channel-max-age' Response Cache-Control Extension | 9 |
| 4.2. | The 'stale' Cache Channel Event | 9 |
| 5. | Security Considerations | 10 |
| 6. | Normative References | 10 |
| Appendix A. | Acknowledgements | 10 |
| Appendix B. | Operational Considerations | 10 |
| Appendix C. | Implementation Notes | 11 |
| | Author's Address | 12 |
| | Intellectual Property and Copyright Statements | 13 |

[1.](#) Introduction

This specification defines "cache channels" to propagate out-of-band events from HTTP [\[1\]](#) origin servers (or their delegates) to subscribing caches. It also defines an event payload that gives finer-grained control over cache freshness.

Typically, a cache will discover channels of interest by examining Cache-Control response headers for the "channel" extension; when present, it indicates that the response is associated with that channel. Upon subscription, caches will receive all events in that channel.

To allow use of a variety of underlying protocols, the process of subscription and the means of propagating events in the channel are specific to the transport in use. This specification does define one such channel transport, using the Atom Syndication Format [\[2\]](#).

Likewise, channels may be used to convey a variety of events from origin servers to caches. This specification defines one such payload, the 'stale' event, that affords finer-grained control over freshness than available in HTTP alone.

Together, cache channels and the stale event enable an origin server to maintain control over the content of a set of caches while increasing their efficiency. For example, "reverse proxies" are often used to accelerate HTTP servers by caching their content; cache channels and stale events can be used to more closely control their behaviour.

This use of cache channels is similar to an invalidation protocol, except that the protocol described here operates by extending cached responses' freshness lifetime, rather than invalidating them. This preserves the semantics of the HTTP caching model and assures that the failure modes are safe.

Additionally, the 'group' functionality of cache channels enables stale events to apply to several cached responses, thereby offering control over freshness of responses whose request-URIs may not be known. For example, a HTTP search interface may have given several responses containing the same information; if they are grouped together, it is possible to force them to become stale without knowing their request-URIs.

[2.](#) Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

Nottingham

Expires November 11, 2007

[Page 3]

Internet-Draft

Cache Channels

May 2007

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [3], as scoped to those conformance targets.

This specification uses the augmented Backus-Naur Form of [RFC2616](#) [1], and includes the delta-seconds rule from that specification, and the absolute-URI rule from [RFC3986](#) [4].

Elements defined by this specification use the XML namespace [5] URI "http://purl.org/syndication/cache-channel". In this specification, that URI is assumed to be bound to the prefix "cc".

[3.](#) Cache Channels

A cache channel is an out-of-band path from an origin server (or its delegate) to one or more interested caches. It is identified by a URI [4].

A channel contains events, each of which may be associated with one or more URIs. The payload of each event is applied to its associated URIs when it is received.

Typically, a cache channel will convey events applicable to a variety of URIs in an administrative domain; for example, it might carry all events that apply to a single origin server, or a group of origin servers.

From the perspective of a cache, a channel has several interesting

states;

- o unsubscribed: The channel is not being monitored.
- o connected: The channel is being monitored, and events are able to be propagated.
- o disconnected: The channel is being monitored, but events are not able to be propagated.

Channels MUST make the events that they contain available for an advertised amount of time, known as the lifetime of the channel. This allows clients that have been disconnected to re-synchronise themselves with the contents of the channel upon becoming re-connected.

Channels SHOULD also advertise the approximate propagation delay, known as their precision.

Two general processes facilitate the operation of cache channels; subscription and event application.

[3.1.](#) Channel Subscription

Channels are advertised using the "channel" HTTP response cache-control extension.

```
channel-extension = "channel" "=" <"> channel-URI <">
    channel-URI = absolute-URI
```

The recipient of this cache-control extension can subscribe to the channel-URI when interested in receiving events associated with the response.

The specific mechanism for subscription is determined by the channel-URI's scheme and other factors (e.g., the media type of the representation obtained by dereferencing that URI).

If a subsequent response has a different channel-URI, or no channel-URI, and there are no other responses associated with the same channel-URI, a subscriber SHOULD unsubscribe from the channel.

A response MUST NOT have more than one channel-extension.

For example,

```
Cache-Control: max-age=300, channel="http://example.org/channels/a"
```

[3.2.](#) Event Application

An event carries one or more event-URIs, which are used to determine what cached responses the event applies to. This includes responses whose request-URIs match an event-URI, and those with a group-URI matching an event-URI.

Responses can be associated with a group-URI using the "group" response Cache-Control extension;

```
group-extension = "group" "=" <"> group-URI <">  
group-URI = absolute-URI
```

A response MAY have any number of group-extensions.

For example, a response carrying the following Cache-Control header;

```
Cache-Control: max=age=600, channel="http://example.com/channels/a",  
group="urn:uuid:30A909D9-BC7A-4257-BE09-6F781AD6471F"
```

will not only have those events which match the response's request-URI applied, but also events who match the URI

"urn:uuid:30A909D9-BC7A-4257-BE09-6F781AD6471F".

This mechanism allows application of events to arbitrary sets of responses using a synthetic identifier.

For example, if "http://example.com/", "http://example.com/top.html" and "http://example.com/index.html" are all associated with the group identified by "urn:uuid:30A909D9-BC7A-4257-BE09-6F781AD6471F", an event with that group-URI will be applied to all three.

By default, an event will apply to a matching response regardless of the headers used to select it (as indicated by the Vary response header). However, a particular event type can be specified to override this behaviour.

Note that an event **MUST NOT** be applied to responses whose channel-extension does not indicate that the response is associated with the channel that the event occurred in.

[3.3.](#) Atom Cache Channels

The Atom Syndication Format [\[2\]](#) -- when used with channel-specific extensions in a specific pattern over HTTP -- is one suitable cache channel transport.

A channel is subscribed to by commencing regular polling of the channel-URI. Polling **MUST** be attempted at least as often as the channel's precision, as indicted by the cc:precision feed extension.

The feed's 'current' link relation value **MUST** contain the channel-URI, which **MUST** be a HTTP URI.

The channel is considered disconnected when the last successful poll occurred more than channel-precision seconds ago, or a successful poll has not yet occurred. A successful poll is defined as one that results in a fresh (according to the HTTP caching model) copy of a well-formed, valid feed document with a 'self' link relation whose value is character-for-character identical to the channel-URI.

The feed **MAY** be archived [\[6\]](#) to allow the full state of the channel to be reconstructed, while minimising the amount of bandwidth that polling the channel consumes.

The cc:lifetime feed extension indicates the minimum span of time that events are available in the feed for.

Entries in the feed represent events, in reverse-chronological order. The atom:link element(s) with the "alternate" relation determines the

URI(s) that an event is applicable to.

[3.3.1.](#) The 'cc:precision' Feed Extension

An Atom cache channel's precision is indicated by the cc:precision element. This is an feed-level extension element whose content indicates the precision in seconds.

For example;

```
<cc:precision>60</cc:precision>
```

indicates that the precision of the channel it occurs in is one minute.

[3.3.2.](#) The 'cc:lifetime' Feed Extension

An Atom cache channel's lifetime is indicated by the cc:lifetime element. This is an feed-level extension element whose content indicates the lifetime in seconds.

For example;

```
<cc:lifetime>2592000</cc:lifetime>
```

indicates that the lifetime of the channel it occurs in is 30 days.

[3.3.3.](#) Example


```

<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:cc="http://purl.org/syndication/cache-channel">
  <title>Invalidations for www.example.org</title>
  <id>http://admin.example.org/events/</id>
  <link rel="self"
    href="http://admin.example.org/events/current"/>
  <link rel="prev-archive"
    href="http://admin.example.org/events/archive/1234"/>
  <updated>2007-04-13T11:23:42Z</updated>
  <author>
    <name>Administrator</name>
    <email>web-admin@example.org</email>
  </author>
  <cc:precision>60</cc:precision>
  <cc:lifetime>2592000</cc:lifetime>
  <entry>
    <title>stale</title>
    <id>http://admin.example.org/events/1124</id>
    <updated>2007-04-13T11:23:42Z</updated>
    <link href="urn:uuid:50D3565C-97A8-40E1-A5C8-CFA070166FEF"/>
    <cc:stale/>
  </entry>
  <entry>
    <title>stale</title>
    <id>http://admin.example.org/events/1125</id>
    <updated>2007-04-13T10:31:01Z</updated>
    <link href="http://www.example.org/img/123.gif" type="image/gif"/>
    <link href="http://www.example.org/img/123.png" type="image/png"/>
    <cc:stale/>
  </entry>
</feed>

```

This feed document contains two events, the most recent applying to the URI "urn:uuid:50D3565C-97A8-40E1-A5C8-CFA070166FEF" and the other to "http://www.example.org/img/123.gif" and "http://www.example.org/img/123.png". The previous archive document is located at "http://admin.example.org/events/archive/1234", to allow the client to reconstruct missed events if necessary. The channel it represents has a precision of 60 seconds (thereby telling clients that they need to poll at least that often), and a lifetime of 30 days.

4. Manging Freshness with Cache Channels

One use of cache channels is the management of cached responses'

freshness lifetime. This is achieved through use of the 'channel-max-age' response Cache-Control extension, which allows subscribed caches to extend the freshness of a response until an applicable 'stale' cache channel event is received.

[4.1.](#) The 'channel-max-age' Response Cache-Control Extension

The channel-max-age response cache-control extension allows controlled extension of the freshness lifetime of a cached response.

```
channel-max-age = "channel-max-age" "=" delta-seconds
```

A response containing this extension MAY be considered fresh when all of the following conditions are true;

- o The cache is connected to the channel indicated by the channel-URI.
- o The channel does not contain a 'stale' event applicable to the cached response.
- o The response's current_age (as per HTTP [\[1\]](#), Section 13.2.3) is no more than delta-seconds.

For example a response with this Cache-Control header;

```
Cache-Control: channel="http://admin.example.org/events/current",  
               group="urn:uuid:50D3565C-97A8-40E1-A5C8-CFA070166FEF",  
               channel-max-age=86400, max-age=30
```

will be considered fresh for 30 seconds by a cache that is not subscribed to the channel "http://admin.example.org/events/current", but can be considered fresh for up to a day by one that is, as long as the channel does not contain an applicable 'stale' event.

[4.2.](#) The 'stale' Cache Channel Event

Cache channels can indicate that all cached responses associated with a URI are to be considered stale with the 'stale' event. In Atom channels, this event is indicated by the cc:stale entry-level extension;

```
<cc:stale/>
```

When received, this event indicates that any cached response associated with the event's URI(s) MUST be considered stale (for purposes of extension by channel-max-age) within channel-precision seconds.

[5.](#) Security Considerations

Publishers of cache channels MAY require credentials to be presented.

[6.](#) Normative References

- [1] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [2] Nottingham, M. and R. Sayre, "The Atom Syndication Format", [RFC 4287](#), December 2005.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [4] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [5] Bray, T., Hollander, D., and A. Layman, "Namespaces in XML", W3C REC REC-xml-names-19990114, January 1999.
- [6] Nottingham, M., "Feed Paging and Archiving", [draft-nottingham-atompub-feed-history-09](#) (work in progress), April 2007.

[Appendix A.](#) Acknowledgements

Thanks to Jeff McCarrell, John Nienart, Henrik Nordstrom, Evan Torrie, and Chris Westin for their suggestions. The author takes all responsibility for errors and omissions.

[Appendix B.](#) Operational Considerations

There are a number of aspects to considered when using cache

channels;

- o They are most efficient when a small number of channels (ideally, one) is used to convey events about a large number of associated resources (e.g., an entire Web site, or a number of related sites).
- o In particular, when using Atom channels care should be taken to assure that the additional requests necessary to poll the channel are offset by the load reduction achieved. In doing so, the

anticipated number of clients, channel-precision, change rate for cached responses and number of responses being monitored by the channel need to be considered.

- o Feed documents from Atom-based channels should be cacheable, to allow clusters of caches and cache hierarchies to share them more efficiently.
- o When using channels to update freshness information, it is critical to assure that any new content is actually available before events are propagated; if the event is too early and stale cached content forces revalidation, it is possible that the updated content will not be loaded into cache.
- o Responses that use the channel-max-age mechanism should also specify a max-age, both to allow channel-naive caches to store them in a limited fashion, and also to allow some types of channel implementations to initially store the response before subscribing.

[Appendix C](#). Implementation Notes

Handling of the 'stale' event in order to extend freshness can often be effected in an existing cache implementation with only small changes.

In particular, most caches can be easily modified to call a function (whether in-process or in a separate process) when a stale response is found, before the decision to validate it on the origin server is made. Using the request-URI, the stored Cache-Control response header and the age of the cached response as input, such a function should return either STALE, which indicates that the cached response is in fact stale, or FRESH, along with an indication of how much the freshness lifetime should be extended by.

This function determines its response based upon its application of the following rules to the state that is collected about the channel in question;

- o If the channel-URI is missing from the Cache-Control response header, the response is assumed to not be associated with any channel, and a STALE can immediately be returned.
- o If the channel is unsubscribed, it should be scheduled for subscription, and STALE returned.
- o If the channel is disconnected, return STALE.
- o If there is a 'stale' event in the channel that applies to the request-URI or group-URI (if present), and cached response's age is greater than the age of the of that event (calculated using the invalidate-date), return STALE.

Nottingham

Expires November 11, 2007

[Page 11]

Internet-Draft

Cache Channels

May 2007

- o If the cached response's age is greater than channel-max-age, return STALE.
- o If the cached response's age is greater than the channel's lifetime, return STALE.
- o Otherwise, return FRESH freshness=[channel-precision].

Author's Address

Mark Nottingham
Yahoo! Inc.

Email: mnot@yahoo-inc.com
URI: <http://www.mnot.net/>

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).