Network Working Group
Internet-Draft
Updates: 4287 (if approved)
Intended status: Standards Track
Expires: October 28, 2013

E. Wilde
EMC

April 26, 2013

# Profile Support for the Atom Syndication Format
draft-wilde-atom-profile-01

## Abstract

The Atom syndication format is a generic XML format for representing collections. Profiles are one way how Atom feeds can indicate that they support specific extensions. To make this support visible on the media type level, this specification re-registers the Atom media type, and adds a "profile" media type parameter. This allows profiles to become visible at the media type level, so that servers as well as clients can indicate support for specific Atom profiles in conversations, for example when communicating via HTTP.

## Note to Readers

This draft should be discussed on the atom-syntax mailing list [7].

Online access to all versions and files is available on github [8].

## Status of this Memo

## Copyright Notice

## Table of Contents

## 1. Introduction

The Atom Syndication Format "is an XML-based document format that describes lists of related information known as 'feeds'. Feeds are composed of a number of items, known as 'entries', each with an extensible set of attached metadata. For example, each entry has a title." [1]

Profiles "can be described as additional semantics that can be used to process a resource representation, such as constraints, conventions, extensions, or any other aspects that do not alter the basic media type semantics. A profile MUST NOT change the semantics of the resource representation when processed without profile knowledge, so that clients both with and without knowledge of a profiled resource can safely use the same representation." [2]

Profiles are identified by URI, and their use can be indicated for a representation by adding a link with the registered "profile" link relation type, linking to the profile URI. While this is sufficient to represent the fact that a certain representation is using a profile, it does not make that fact visible outside of this representation. Ideally, peers communicating their media type, for example when communicating via Hypertext Transfer Protocol (HTTP) [5], should be able to indicate the support of certain profiles through the media type identifier itself, without changing the base media type.

Because Atom supports generic links through its <link/> element, "profile" links can be easily added to a feed, indicating that this feed does adhere to a certain profile. However, on the media type level, this feed would still be labeled as application/atom+xml, making the profile invisible on that level and thus not allowing it to be used in interactions such as content negotiation in HTTP.

This specification adds a "profile" media type parameter to the application/atom+xml media type, thereby making it possible for profiles to be exposed at the media type level. Apart from adding that one media type parameter, this specification does not change anything about the Atom format itself, or its media type registration.

## 2. Examples

Adding a "profile" parameter to the Atom media type adds visibility of profiles at the media type level, for example when alternative profiles are supported by a service. It might also help to further "specialize" a media type in environments where such a

"specialization" is useful. Two examples are intended to illustrate these two scenarios.

### 2.1. Profiles for Alternatives

For example, when linking to feeds of media-oriented services, it would be possible to expose two feeds, one using MediaRSS, and the other one using Podcasts. Both formats roughly cover the same functionality as media-oriented feed-based extensions, but by having the ability to expose their capabilities at the media type level, HTTP mechanisms and conversations can be used to distinguish between these formats.

In some cases it may be possible to support more than one profile, and then it is up for the service to decide whether these should be exposed in one representation (which can be exposed by linking to multiple profiles from the resource representation and/or in the media type parameter), or whether there should be two representations, one for each profile. This decision will probably depend on implementation complexity, the trade-off between navigation complexity (two representations with one profile each) and processing complexity, and also the size of the profile data, because in particular in the case of overlapping profiles, there might be many redundancies.

Thus, which way to go for multiple profiles is not a question that has one correct answer; it depends on the profiles, and on the services that are built around them.

## 2.2. Profiles for Specializations

Feed-based services may provide additional features in feeds that are represented using Atom's extension mechanisms. These additional features might be useful only for those clients that support them, and otherwise might add volume to a feed that is of no value to general consumers. In such a scenario, specialized clients might also request their specialized features via profile media type parameters, and will then get the feed being "enriched" with the additional features. If clients do not request such a profile or request one that is not known to the server, the server responds with a generic feed, still allowing them to treat the feed as a generic feed (with no additional features being represented).

Whether services respond with profiles by default or only for specific requests about a profile is a matter of policy, and will be influenced by factors such as the added volume when adding profile data, and the question whether profiles should only be exposed to those that specifically ask for them. Since profiles are not allowed

to change the semantics of the media type itself, such a decision can depend on the trade-off being a matter of expressivity, and not whether it will break clients under some circumstances.

## 3. Profile Parameter Definition

The profile parameter for the application/atom+xml media type allows one or more profile URIs to be specified. These profile URIs have the identifier semantics defined in [2], and when appearing as media type parameter, they have the same semantics as if they had been associated with the resource URI through other means, such as using one or more <link profile="" href=""/> elements as children of the <feed> element.

As a general rule, media type parameters must be quoted unless they are tokens. For the "profile" media type parameter defined here, this means that is must be quoted. It contains a non-empty list of space-separated URIs (the profile URIs).

```
profile-param = "profile=" profile-value
profile-value = <"> profile-URI 0*( 1*SP profile-URI ) <">
profile-URI   = URI
```

The "URI" in the above grammar refers to the "URI" as defined in Section 3 of [3]

## 4. IANA Considerations

The media type registration for the media type application/atom+xml should be updated according to the following registration.

### 4.1. Atom Media Type application/atom+xml

The Internet media type [6] for an Atom document is application/atom+xml.

#### 4.1.1. Media Type Name

application

#### 4.1.2. Subtype Name

atom+xml

#### 4.1.3. Required Parameters

none

#### 4.1.4. Optional Parameters

charset: This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [4].

profile: This parameter indicates that one or more profiles are used in the feed, according to the definition of profiles in [2]. The parameter syntax is specified in Section 3 of RFC XXXX

#### 4.1.5. Encoding Considerations

Identical to those of "application/xml" as described in [4], Section 3.2.

### 4.1.6. Security Considerations

As defined in [1]. In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [4], Section 10.

### 4.1.7. Interoperability Considerations

There are no known interoperability issues.

### 4.1.8. Published Specification

[1], RFC XXXX

### 4.1.9. Applications which use this media type

Many. Atom has become a common foundation for many syndication-oriented scenarios, and also has become a commonly used representation for collection contents.

### 4.1.10. Magic number(s)

As specified for "application/xml" in [4], Section 3.2.

### 4.1.11. File extension(s)

.atom

### 4.1.12. Fragment Identifiers

As specified for "application/xml" in [4], Section 5.

### 4.1.13. Base URI

As specified in [4], Section 6.

### 4.1.14. Macintosh File Type Code(s)

TEXT

### 4.1.15. Person & email address to contact for further information

Mark Nottingham <mnot@mnot.net> and Erik Wilde <erik.wilde@emc.com>

### 4.1.16. Intended Usage

Common

### 4.1.17. Author/Change Controller

IESG

## 5. Change Log

Note to RFC Editor: Please remove this section before publication.

### 5.1. From -00 to -01

- o Fixed typos.
- o Removed the requirement to percent-encode URIs in the profile parameter.
- o Added example for media type specialization.

## 6. References

### 6.1. Normative References

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

[2] Wilde, E., "The 'profile' Link Relation Type", RFC 6906, March 2013.

[3] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform

Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[4] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

## 6.2. Non-Normative References

[5] 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.

[6] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

## URIs

[7] <http://www.imc.org/atom-syntax/>

[8] <https://github.com/dret/I-D/tree/master/atom-profile>

## Appendix A. Acknowledgements

Thanks for comments and suggestions provided by Markus Lanthaler.

## Author's Address

Erik Wilde
EMC
6801 Koll Center Parkway
Pleasanton, CA 94566
U.S.A.

Phone: +1-925-6006244
Email: erik.wilde@emc.com
URI: http://dret.net/netdret/