

Network Working Group	E. Hammer-Lahav
Internet-Draft	Yahoo!
Intended status: Standards Track	B. Cook
Expires: November 10, 2011	May 09, 2011

Web Host Metadata

draft-hammer-hostmeta-15

## Abstract

This specification describes a method for locating host metadata as well as information about individual resources controlled by the host.

## **Editorial Note (to be removed by RFC Editor)**

Please discuss this draft on the [apps-discuss@ietf.org](mailto:apps-discuss@ietf.org) mailing list.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on November 10, 2011.

## Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

\*1. [Introduction](#)

\*1.1. [Example](#)

\*1.1.1. [Processing Resource-Specific Information](#)

\*1.2. [Notational Conventions](#)

\*2. [Obtaining host-meta Documents](#)

\*3. [The host-meta Document](#)

\*3.1. [XML Document format](#)

\*3.1.1. [The 'Link' Element](#)

\*3.1.1.1. [Template Syntax](#)

\*4. [Processing host-meta Documents](#)

\*4.1. [Host-Wide Information](#)

\*4.2. [Resource-Specific Information](#)

\*5. [Security Considerations](#)

\*6. [IANA Considerations](#)

\*6.1. [The 'host-meta' Well-Known URI](#)

\*6.2. [The 'lrdd' Relation Type](#)

\*7. [References](#)

\*[Authors' Addresses](#)

## **[1. Introduction](#)**

Web-based protocols often require the discovery of host policy or metadata, where "host" is not a single resource but the entity controlling the collection of resources identified by Uniform Resource Identifiers (URI) with a common URI host [\[RFC3986\]](#).

While web protocols have a wide range of metadata needs, they often use metadata that is concise, has simple syntax requirements, and can benefit from storing their metadata in a common location used by other related protocols.

Because there is no URI or representation available to describe a host, many of the methods used for associating per-resource metadata (such as HTTP headers) are not available. This often leads to the overloading of the root HTTP resource (e.g. 'http://example.com/') with host metadata that is not specific or relevant to the root resource itself.

This specification registers the well-known URI suffix host-meta in the Well-Known URI Registry established by [\[RFC5785\]](#), and specifies a

simple, general-purpose metadata document format for hosts, to be used by multiple web-based protocols.

In addition, there are times when a host-wide scope for policy or metadata is too coarse-grained. host-meta provides two mechanisms for providing resource-specific information:

- \*Link Templates - links using a URI template instead of a fixed target URI, providing a way to define generic rules for generating resource-specific links by applying the individual resource URI to the template.

- \*Link-based Resource Descriptor Documents (LRDD, pronounced 'lard') - descriptor documents providing resource-specific information, typically information that cannot be expressed using link templates. LRDD documents are linked to resources or host-meta documents using link templates with the lrdd relation type.

### **1.1. Example**

The following is a simple host-meta document including both host-wide and resource-specific information for the 'example.com' host:

```
<?xml version='1.0' encoding='UTF-8'?>
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>

  <!-- Host-wide Information -->

  <Property type='http://protocol.example.net/version'>1.0</Property>

  <Link rel='copyright'
    href='http://example.com/copyright' />

  <!-- Resource-specific Information -->

  <Link rel='hub'
    template='http://example.com/hub' />

  <Link rel='lrdd'
    type='application/xrd+xml'
    template='http://example.com/lrdd?uri={uri}' />

  <Link rel='author'
    template='http://example.com/author?q={uri}' />

</XRD>
```

The host-wide information which applies to host in its entirety provided by the document includes:

- \*A `http://protocol.example.net/version` host property with a value of `1.0`.

- \*A link to the host's copyright policy (copyright).

The resource-specific information provided by the document includes:

- \*A link template for receiving real-time updates (hub) about individual resources. Since the template does not include a template variable, the target URI is identical for all resources.

- \*A LRDD document link template (lrdd) for obtaining additional resource-specific information contained in a separate document for each individual resource.

- \*A link template for finding information about the author of individual resources (author).

#### 1.1.1. Processing Resource-Specific Information

When looking for information about the an individual resource, for example, the resource identified by `'http://example.com/xy'`, the resource URI is applied to the templates found, producing the following links:

```
<Link rel='hub'
  href='http://example.com/hub' />
```

```
<Link rel='lrdd'
  type='application/xrd+xml'
  href='http://example.com/lrdd?uri=http%3A%2F%2Fexample.com%2Fxy' />
```

```
<Link rel='author'
  href='http://example.com/author?q=http%3A%2F%2Fexample.com%2Fxy' />
```

The LRDD document for `'http://example.com/xy'` is obtained using an HTTP GET request:

```

<?xml version='1.0' encoding='UTF-8'?>
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>

  <Subject>http://example.com/xy</Subject>

  <Property type='http://spec.example.net/color'>red</Property>

  <Link rel='hub'
    href='http://example.com/another/hub' />

  <Link rel='author'
    href='http://example.com/john' />
</XRD>

```

Together, the information available about the individual resource (presented as an XRD document for illustration purposes) is:

```

<?xml version='1.0' encoding='UTF-8'?>
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>

  <Subject>http://example.com/xy</Subject>

  <Property type='http://spec.example.net/color'>red</Property>

  <Link rel='hub'
    href='http://example.com/hub' />

  <Link rel='hub'
    href='http://example.com/another/hub' />

  <Link rel='author'
    href='http://example.com/john' />

  <Link rel='author'
    href='http://example.com/author?q=http%3A%2F%2Fexample.com%2Fxy' />

</XRD>

```

Note that the order of links matters and is based on their original order in the host-meta and LRDD documents. For example, the hub link obtained from the host-meta link template has a higher priority than the link found in the LRDD document because the host-meta link appears before the lrdd link.

On the other hand, the author link found in the LRDD document has a higher priority than the link found in the host-meta document because it appears after the lrdd link.

## **1.2. 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 [\[RFC2119\]](#).

This document uses the Augmented Backus-Naur Form (ABNF) notation of [\[RFC5234\]](#). Additionally, the following rules are included from [\[RFC3986\]](#): reserved, unreserved, and pct-encoded.

## **2. Obtaining host-meta Documents**

The client obtains the host-meta document for a given host by sending an HTTP [\[RFC2616\]](#) or an HTTPS [\[RFC2818\]](#) GET request to the host for the /.well-known/host-meta path, using the default ports defined for each protocol (e.g. port 80 for HTTP and port 443 for HTTPS). The scope and meaning of host-meta documents obtained via other protocols or ports is undefined.

The server MUST support at least one protocol but MAY support both. If both protocols are supported, they MUST produce the same document. The decision which protocol is used to obtain the host-meta document have significant security ramifications as described in [Section 5](#). For example, the following request is used to obtain the host-meta document for the 'example.com' host:

```
GET /.well-known/host-meta HTTP/1.1
Host: example.com
```

If the server response indicates that the host-meta resource is located elsewhere (a 301, 302, or 307 response status code), the client MUST try to obtain the resource from the location provided in the response. This means that the host-meta document for one host MAY be retrieved from another host. Likewise, if the resource is not available or does not exist (e.g. a 404 or 410 response status codes) using both the HTTP and HTTPS protocols, the client should infer that metadata is not available via this mechanism.

The host-meta document SHOULD be served with the application/xrd+xml media type. [[ media type registration pending ]]

## **3. The host-meta Document**

The host-meta document uses the XRD 1.0 document format as defined by [\[OASIS.XRD-1.0\]](#), which provides a simple and extensible XML-based schema for describing resources. This specification defines additional

processing rules needed to describe hosts. Documents MAY include any XRD element not explicitly excluded.

The server MAY offer alternative representations of any XRD document it serves (host-meta, LRDD, or other XRD-based documents). The client MAY request a particular representation using the HTTP Accept request header field. If no Accept request header field is included with the request, or if the client requests a `application/xrd+xml` representation, the server MUST respond using the REQUIRED XRD 1.0 XML representation described in [Section 3.1](#).

The XRD 1.0 XML representation is the only canonical representation for any XRD document. If there is any discrepancy between the content of the XRD 1.0 XML representation and any other representation for the same resource, the client MUST only use the XRD 1.0 XML representation. Applications using the host-meta document MAY require the server to provide a specific alternative representation in addition to the XRD 1.0 XML representation when explicitly requested by the client. A JavaScript Object Notation (JSON) XRD 1.0 representation is described in [\[json\]](#).

### [3.1. XML Document format](#)

The host-meta document root MUST be an XRD element. The document SHOULD NOT include a Subject element, as at this time no URI is available to identify hosts. The use of the Alias element in host-meta is undefined and NOT RECOMMENDED.

The subject (or "context resource" as defined by [\[RFC5988\]](#)) of the XRD Property and Link elements is the host described by the host-meta document. However, the subject of Link elements with a `template` attribute is the individual resource whose URI is applied to the link template as described in [Section 3.1.1](#).

#### [3.1.1. The 'Link' Element](#)

The XRD Link element, when used with the `href` attribute, conveys a link relation between the host described by the document and a common target URI.

For example, the following link declares a common copyright license for the entire scope:

```
<Link rel='copyright' href='http://example.com/copyright' />
```

However, a Link element with a `template` attribute conveys a relation whose context is an individual resource within the host-meta document scope, and whose target is constructed by applying the context resource URI to the template. The template string MAY contain a URI string without any variables to represent a resource-level relation that is identical for every individual resource.

For example, a blog with multiple authors can provide information about each article's author by providing an endpoint with a parameter set to the URI of each article. Each article has a unique author, but all share the same pattern of where that information is located:

```
<Link rel='author'
  template='http://example.com/author?article={uri}' />
```

### 3.1.1.1. Template Syntax

This specification defines a simple template syntax for URI transformation. A template is a string containing brace-enclosed ("{}") variable names marking the parts of the string that are to be substituted by the corresponding variable values.

Before substituting template variables, values **MUST** be encoded using UTF-8 and any character other than unreserved (as defined by [\[RFC3986\]](#)) **MUST** be percent-encoded per [\[RFC3986\]](#).

This specification defines a single variable - uri - as the entire context resource URI. Protocols **MAY** define additional relation-specific variables and syntax rules, but **SHOULD** only do so for protocol-specific relation types, and **MUST NOT** change the meaning of the uri variable. If a client is unable to successfully process a template (e.g. unknown variable names, unknown or incompatible syntax) the parent Link element **SHOULD** be ignored.

The template syntax ABNF:

```
URI-Template = *( uri-char / variable )
variable     = "{" var-name "}"
uri-char     = ( reserved / unreserved / pct-encoded )
var-name     = %x75.72.69 / ( 1*var-char ) ; "uri" or other names
var-char     = ALPHA / DIGIT / "." / "_"
```

For example:

```
Input:   http://example.com/r?f=1
Template: http://example.org/?q={uri}
Output:  http://example.org/?q=http%3A%2F%2Fexample.com%2Fr%3Ff%3D1
```



## **4. Processing host-meta Documents**

Once the host-meta document has been obtained, the client processes its content based on the type of information desired: host-wide or resource-specific.

Clients usually look for a link with a specific relation type or other attributes. In such cases, the client does not need to process the entire host-meta document and all linked LRDD documents, but instead, process the various documents in their prescribed order until the desired information is found.

Protocols using host-meta must indicate whether the information they seek is host-wide or resource-specific. For example, "obtain the first host-meta resource-specific link using the 'author' relation type". If both types are used for the same purpose (e.g. first look for resource-specific, then look for host-wide), the protocol must specify the processing order.

### **4.1. Host-Wide Information**

When looking for host-wide information, the client **MUST** ignore any Link elements with a template attribute, as well as any link using the `lrdd` relation type. All other elements are scoped as host-wide.

### **4.2. Resource-Specific Information**

Unlike host-wide information which is contained solely within the host-meta document, resource-specific information is obtained from host-meta link templates, as well as from linked LRDD documents.

When looking for resource-specific information, the client constructs a resource descriptor by collecting and processing all the host-meta link templates. For each link template:

1. The client applies the URI of the desired resource to the template, producing a resource-specific link.
2. If the link's relation type is other than `lrdd`, the client adds the link to the resource descriptor in order.
3. If the link's relation type is `lrdd`:
  - 3.1** The client obtains the LRDD document by following the scheme-specific rules for the LRDD document URI. If the document URI scheme is `http` or `https`, the document is obtained via an HTTP GET request to the identified URI. If the HTTP response status code is 301, 302, or 307, the client **MUST** follow the redirection response and repeat the request with the provided location.
  - 3.2** The client adds any links found in the LRDD document to the resource descriptor in order, except for any link using

the lrdd relation type (processing is limited to a single level of inclusion). When adding links, the client SHOULD retain any extension attributes and child elements if present (e.g. <Property> or <Title> elements).

- 3.3 The client adds any resource properties found in the LRDD document to the resource descriptor in order (e.g. <Alias> or <Property> child elements of the LRDD document <XRD> root element).

## **5. Security Considerations**

The host-meta document is designed to be used by other applications explicitly "opting-in" to use the facility. Therefore, any such application MUST review the specific security implications of using host-meta documents. By itself, this specification does not provide any protections or guarantees that any given host-meta document is under the control of the appropriate entity as required by each application. The metadata returned by the host-meta resource is presumed to be under the control of the appropriate authority and representative of all the resources described by it. If this resource is compromised or otherwise under the control of another party, it may represent a risk to the security of the server and data served by it, depending on the applications using it.

Applications utilizing the host-meta document for sensitive or security related information MUST require the use of the HTTPS protocol and MUST NOT produce a host-meta document using other means. In addition, such applications MUST require that any redirection leading to the retrieval of a host-meta document also utilize the HTTPS protocol.

Since the host-meta document is authoritative for the entire host, not just the authority (combination of scheme, host, and port) of the host-meta document server, applications MUST ensure that using a host-meta document for another URI authority does not represent a potential security exploit.

Protocols using host-meta templates must evaluate the construction of their templates as well as any protocol-specific variables or syntax to ensure that the templates cannot be abused by an attacker. For example, a client can be tricked into following a malicious link due to a poorly constructed template which produces unexpected results when its variable values contain unexpected characters.

## **6. IANA Considerations**

### **6.1. The 'host-meta' Well-Known URI**

This specification registers the host-meta well-known URI in the Well-Known URI Registry as defined by [\[RFC5785\]](#).

**URI suffix:** host-meta

**Change controller:**

IETF

**Specification document(s):** [[ this document ]]

**Related information:** The host-meta documents obtained from the same host using the HTTP and HTTPS protocols (using default ports) MUST be identical.

## **6.2. The 'lrdd' Relation Type**

This specification registers the lrdd relation type in the Link Relation Type Registry defined by [\[RFC5988\]](#):

**Relation Name:** lrdd

**Description:** lrdd (pronounced 'lard') is an acronym for Link-based Resource Descriptor Document. It is used by the host-meta document processor to locate resource-specific information about individual resources. When used elsewhere (e.g. in HTTP Link header fields or in HTML <LINK> elements), it operates as an include directive, identifying the location of additional links and other metadata. Multiple links with the 'lrdd' relation indicate multiple sources to include, not alternative sources of the same information. An application/xrd+xml representation MUST be available, and this media type MAY appear in a link's type attribute. Additional representations MAY be available (using the HTTP Accept request header field), in which case the link's type attribute SHOULD be omitted.

**Reference:** [[ This specification ]]

## **7. References**

<b>[RFC2119]</b>	<a href="#">Bradner, S.</a> , " <a href="#">Key words for use in RFCs to Indicate Requirement Levels</a> ", BCP 14, RFC 2119, March 1997.
<b>[RFC2616]</b>	<a href="#">Fielding, R.</a> , <a href="#">Gettys, J.</a> , <a href="#">Mogul, J.</a> , <a href="#">Frystyk, H.</a> , <a href="#">Masinter, L.</a> , <a href="#">Leach, P.</a> and <a href="#">T. Berners-Lee</a> , " <a href="#">Hypertext Transfer Protocol -- HTTP/1.1</a> ", RFC 2616, June 1999.
<b>[RFC2818]</b>	<a href="#">Rescorla, E.</a> , " <a href="#">HTTP Over TLS</a> ", RFC 2818, May 2000.
<b>[RFC3986]</b>	<a href="#">Berners-Lee, T.</a> , <a href="#">Fielding, R.</a> and <a href="#">L. Masinter</a> , " <a href="#">Uniform Resource Identifier (URI): Generic Syntax</a> ", STD 66, RFC 3986, January 2005.
<b>[RFC4627]</b>	<a href="#">Crockford, D.</a> , " <a href="#">The application/json Media Type for JavaScript Object Notation (JSON)</a> ", RFC 4627, July 2006.

[RFC5234]	Crocker, D. and P. Overell, " <a href="#">Augmented BNF for Syntax Specifications: ABNF</a> ", STD 68, RFC 5234, January 2008.
[RFC5785]	Nottingham, M. and E. Hammer-Lahav, " <a href="#">Defining Well-Known Uniform Resource Identifiers (URIs)</a> ", RFC 5785, April 2010.
[RFC5988]	Nottingham, M., " <a href="#">Web Linking</a> ", RFC 5988, October 2010.
[OASIS.XRD-1.0]	Hammer-Lahav, E.H and W.N Norris, "Extensible Resource Descriptor (XRD) Version 1.0", .

### [Authors' Addresses](#)

Eran Hammer-Lahav Hammer-Lahav Yahoo! EMail: [eran@hueniverse.com](mailto:eran@hueniverse.com)  
URI: <http://hueniverse.com>

Blaine Cook Cook EMail: [romeda@gmail.com](mailto:romeda@gmail.com) URI: <http://romeda.org>