

**Cool URIs for the Semantic Web**

**W3C Interest Group Note 03 December 2008**

**This version:**

<http://www.w3.org/TR/2008/NOTE-cooluris-20081203/>

**Latest version:**

<http://www.w3.org/TR/cooluris/>

**Previous version:**

<http://www.w3.org/TR/2008/NOTE-cooluris-20080331/>

**Editors:**

[Leo Sauermann](http://www.dfki.uni-kl.de/~sauermann/) ([DFKI GmbH](http://www.dfki.de))

[Richard Cyganiak](http://richard.cyganiak.de/) ([DERI, NUI Galway](http://www.deri.ie/) and [Freie Universität Berlin](http://www.fu-berlin.de/en/))

**Contributors:**

[Danny Ayers](http://dannyayers.com/) ([Talis Information Ltd.](http://talis.com))

[Max Völkel](http://www.xam.de/) ([FZI Karlsruhe](http://www.fzi.de/))

Please refer to the [**errata**](http://www.w3.org/2001/sw/sweo/public/2008/Errata-in-CoolURIs.html) for this document, which may include some corrections.

[Copyright](http://www.w3.org/Consortium/Legal/ipr-notice#Copyright) © 2008 [W3C](http://www.w3.org/)® ([MIT](http://www.csail.mit.edu/), [ERCIM](http://www.ercim.org/), [Keio](http://www.keio.ac.jp/)), All Rights Reserved. W3C [liability](http://www.w3.org/Consortium/Legal/ipr-notice#Legal_Disclaimer), [trademark](http://www.w3.org/Consortium/Legal/ipr-notice#W3C_Trademarks) and [document use](http://www.w3.org/Consortium/Legal/copyright-documents) rules apply.

**Abstract**

The *Resource Description Framework* RDF allows users to describe both Web documents and concepts from the real world—people, organisations, topics, things—in a computer-processable way. Publishing such descriptions on the Web creates the *Semantic Web*. URIs (Uniform Resource Identifiers) are very important, providing both the core of the framework itself and the link between RDF and the Web. This document presents guidelines for their effective use. It discusses two strategies, called *303 URIs* and *hash URIs*. It gives pointers to several Web sites that use these solutions, and briefly discusses why several other proposals have problems.

**Status of this document**

*This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the* [*W3C technical reports index*](http://www.w3.org/TR/) *at http://www.w3.org/TR/.*

This is a W3C Interest Group Note giving a tutorial explaining decisions of the TAG for newcomers to Semantic Web technologies. It was initially based on the [DFKI Technical Memo TM-07-01, *Cool URIs for the Semantic Web*](http://www.dfki.uni-kl.de/dfkidok/publications/TM/07/01/tm-07-01.pdf) and was subsequently published as a W3C Working draft in [December 2007](http://www.w3.org/TR/2007/WD-cooluris-20071217/), and again in [March 2008](http://www.w3.org/TR/2008/WD-cooluris-20080321/) by the [Semantic Web Education and Outreach (SWEO) Interest Group](http://www.w3.org/2001/sw/sweo/) of the W3C, part of the [W3C Semantic Web Activity](http://www.w3.org/2001/sw/). The drafts were publicly reviewed, especially by the [Technical Architecture Group (TAG)](http://www.w3.org/2001/tag/) and the [Semantic Web Deployment Group (SWD)](http://www.w3.org/2006/07/SWD/). The only change from the previous version of this document is the addition of a link to an [errata page](http://www.w3.org/2001/sw/sweo/public/2008/Errata-in-CoolURIs.html).

The charter of the [Semantic Web Education and Outreach (SWEO) Interest Group](http://www.w3.org/2001/sw/sweo/) expired at the end of March, 2008. Nevertheless, this document may be taken up by some other groups in the future for further development. Feedbacks on this documents is therefore encouraged. Please send comments about this document to public-sweo-ig@w3.org (with [public archive](http://lists.w3.org/Archives/Public/public-sweo-ig/)). A complete [list of changes](http://www.w3.org/TR/cooluris/#changelog) is available.

Publication as an Interest Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](http://www.w3.org/Consortium/Patent-Policy-20040205/). The group does not expect this document to become a W3C Recommendation. W3C maintains a [public list of any patent disclosures](http://www.w3.org/2004/01/pp-impl/39572/status) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim(s)](http://www.w3.org/Consortium/Patent-Policy-20040205/#def-essential) must disclose the information in accordance with [section 6 of the W3C Patent Policy](http://www.w3.org/Consortium/Patent-Policy-20040205/#sec-Disclosure).

The disclosure obligations of the Participants of this group are described in the [charter](http://www.w3.org/2006/07/sweoig-charter.html#pd).

**Scope**

This document is a practical guide for implementers of the RDF specification. The title is inspired by Tim Berners-Lee's article "Cool URIs don't change" [[Cool](http://www.w3.org/TR/cooluris/#ref-Cool)].  It explains two approaches for RDF data hosted on [HTTP](http://www.w3.org/TR/cooluris/#ref-RFC2616) servers. Intended audiences are Web and ontology developers who have to decide how to model their RDF URIs for use with HTTP. Applications using non-HTTP URIs are not covered. This document is an informative guide covering selected aspects of previously published, detailed technical specifications. The 303 URIs are based on the [httpRange-14 resolution](http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html) [*[httpRange](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-httpRange)*] by the [Technical Architecture Group (TAG)](http://www.w3.org/2001/tag/). We assume that you are familiar with the [basics of the RDF data model](http://www.w3.org/TR/2004/REC-rdf-primer-20040210/) [*[RDFPrimer](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-RDFPrimer)*]. We also assume some familiarity with the [HTTP protocol](http://www.ietf.org/rfc/rfc2616.txt) [[*RFC2616*](http://www.w3.org/TR/cooluris/#ref-RFC2616)]. [Wikipedia's article](http://en.wikipedia.org/wiki/HTTP) [[*WP-HTTP*](http://www.w3.org/TR/cooluris/#ref-WP-HTTP)] serves as a good primer.

**Table of Contents**

* [1. Introduction](http://www.w3.org/TR/cooluris/#intro)
* [2. URIs for Web Documents](http://www.w3.org/TR/cooluris/#oldweb)
	+ [2.1. HTTP and Content Negotiation](http://www.w3.org/TR/cooluris/#conneg)
* [3. URIs for Real-World Objects](http://www.w3.org/TR/cooluris/#semweb)
	+ [3.1 Distinguishing between Representations and Descriptions](http://www.w3.org/TR/cooluris/#distinguishing)
* [4. Two Good Solutions](http://www.w3.org/TR/cooluris/#solutions)
	+ [4.1. Hash URIs](http://www.w3.org/TR/cooluris/#hashuri)
	+ [4.2. 303 URIs forwarding to One Generic Document](http://www.w3.org/TR/cooluris/#r303gendocument)
	+ [4.3. 303 URIs forwarding to Different Documents](http://www.w3.org/TR/cooluris/#r303uri)
	+ [4.4. Choosing Between 303 and Hash](http://www.w3.org/TR/cooluris/#choosing)
	+ [4.5. Cool URIs](http://www.w3.org/TR/cooluris/#cooluris)
	+ [4.6. Linking](http://www.w3.org/TR/cooluris/#linking)
	+ [4.7. Implementing Content Negotiation](http://www.w3.org/TR/cooluris/#implementation)
* [5. Examples from the Web](http://www.w3.org/TR/cooluris/#examples)
* [6. Other Resource Naming Proposals](http://www.w3.org/TR/cooluris/#otherideas)
	+ [6.1. New URI Schemes](http://www.w3.org/TR/cooluris/#schemes)
	+ [6.2. Reference By Description](http://www.w3.org/TR/cooluris/#blanknodes)
* [7. Conclusion](http://www.w3.org/TR/cooluris/#conclusion)
* [8. Acknowledgements](http://www.w3.org/TR/cooluris/#acknowledgements)
* [9. References](http://www.w3.org/TR/cooluris/#references)
* [10. Change log](http://www.w3.org/TR/cooluris/#changelog)

**1. Introduction**

The Semantic Web is envisioned as a decentralised world-wide information space for sharing machine-readable data with a minimum of integration costs. Its two core challenges are the distributed modelling of the world with a shared data model, and the infrastructure where data and schemas can be published, found and used. Users benefit from getting information *"raw and now"* [[Give](http://www.w3.org/TR/cooluris/#ref-Give)] and in portable data formats [[DP](http://www.w3.org/TR/cooluris/#ref-DP)]. Providers often publish data embedded in a fixed user interface, in HTML. A basic question is thus how to publish information about resources in a way that allows interested users and software applications to find and interpret them.

On the Semantic Web, all information has to be expressed as *statements* about *resources*, like *the members of the company Example.com are Alice and Bob* or *Bob's telephone number is "+1 555 262"* or *this Web page was created by Alice*. Resources are identified by *Uniform Resource Identifiers* ([URIs](http://www.ietf.org/rfc/rfc3986.txt)) [[*RFC3986*](http://www.w3.org/TR/cooluris/#ref-RFC3986)]. This modelling approach is at the heart of *Resource Description Framework* ([RDF](http://www.w3.org/TR/2004/REC-rdf-primer-20040210/)) [*[RDFPrimer](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-RDFPrimer)*]. A nice introduction is given in the N3 primer [[N3Primer](http://www.w3.org/TR/cooluris/#ref-N3Primer)].

Using RDF, the statements can be published on the Web site of the company. Others can read the data and publish their own information, linking to existing resources. This forms a distributed model of the world. It allows the user to pick any application to view and work with the same data, for example to see Alice's published address in your address book.

At the same time, Web documents have always been addressed with URIs (in common parlance often referred as Uniform Resource Locators, URLs). This is useful because it means we can easily make RDF statements about Web pages, but also dangerous because we can easily mix up Web pages and the things, or resources, described on the page.

So the question is, what URIs should we use in RDF? As an example, to identify the frontpage of the Web site of Example Inc., we may use http://www.example.com/. But what URI identifies the company as an organisation, not a Web site? Do we have to serve any content—HTML pages, RDF files—at those URIs? In this document we will answer these questions according to relevant specifications. We explain how to use URIs for things that are not Web pages, such as people, products, places, ideas and concepts such as ontology classes. We give detailed examples as to how the Semantic Web can (and should) be realised as a part of the Web.

**2. URIs for Web Documents**

Let us begin with an example. Assume that Example Inc., a fictional company producing "**Ex**treme Guitar **Ampl**ifi**e**rs", has a Web site at http://www.example.com/. Part of the site is a white-pages service listing the names and contact details of the employees. Alice and Bob both work at Example Inc. The structure of the Web site might thus be:

**http://www.example.com/**

the homepage of Example Inc.

**http://www.example.com/people/alice**

the homepage of Alice

**http://www.example.com/people/bob**

the homepage of Bob

Like everything on the traditional Web, each of the pages mentioned above are *Web documents*. Every Web document has its own URI. Note that a Web document is not the same as a file: a single Web document can be available in many different formats and languages, and a single file, for example a PHP script, may be responsible for generating a large number of Web documents with different URIs. A Web document is defined as something that has a URI and can return *representations* (responses in a format such as HTML or JPEG or RDF) of the identified resource in response to HTTP requests. In technical literature, such as [*Architecture of the World Wide Web, Volume One*](http://www.w3.org/TR/2004/REC-webarch-20041215/) [[*AWWW*](http://www.w3.org/TR/cooluris/#ref-AWWW)], the term *Information Resource* is used instead of *Web document*.

On the traditional Web, URIs were used *primarily* for Web documents—to link to them, and to access them in a browser. The notion of resource *identity* was not so important on the traditional Web, a URL simply identified whatever we see when we type it into a browser.

**2.1. HTTP and Content Negotiation**

Web clients and servers use the [HTTP protocol](http://www.ietf.org/rfc/rfc2616.txt) [[*RFC2616*](http://www.w3.org/TR/cooluris/#ref-RFC2616)] to request representations of Web documents and send back the responses. HTTP has a powerful mechanism for offering different formats and language versions of the same Web document known as *content negotiation*.

When a user agent (such as a browser) makes an HTTP request, it sends along some HTTP headers to indicate what data formats and language it prefers. The server then selects the best match from its file system or generates the desired content on demand, and sends it back to the client. For example, a browser could send this HTTP request to indicate that it wants an HTML or XHTML representation of http://www.example.com/people/alice in English or German:

GET /people/alice HTTP/1.1

Host: www.example.com

Accept: text/html, application/xhtml+xml

Accept-Language: en, de

The server could answer:

HTTP/1.1 200 OK

Content-Type: text/html

Content-Language: en

Content-Location: http://www.example.com/people.en.html

followed by the content of the HTML document in English.

Here we see [Content negotiation](http://www.w3.org/2001/tag/doc/alternatives-discovery-20061101.html) [[*TAG-Alt*](http://www.w3.org/TR/cooluris/#ref-TAG-Alt)] in action. The server interprets the Accept-Language headers in the request and decides to return the English representation of the resource in question. Note that the URI of this representation is passed back in the Content-Location header, this is not required but a recommended good practice (see [[CHIPS](http://www.w3.org/TR/cooluris/#ref-CHIPS)], [7.2](http://www.w3.org/TR/chips/#gl7)). Clients see that this URI is connected to the specific representation (in this case English) and search engines can refer to the different representations by using the different URIs. This implies that it is possible to have multiple representations of the same resource.

Content negotation is often implemented with a twist: Instead of a direct answer, the server *redirects* to another URL where the appropriate representation is found:

HTTP/1.1 302 Found

Location: http://www.example.com/people/alice.en.html

The redirect is indicated by a special *Status Code*, here 302 Found. The client would now send another HTTP request to the new URL. By having separate URLs for different representations, this approach allows Web authors to link directly to a specific representation.

RDF/XML, the standard serialisation format of RDF, has its own content type, application/rdf+xml. Content negotiation thus allows publishers to serve HTML representations of a Web document to traditional Web browsers and RDF representations to Semantic Web-enabled user agents. This also allows servers to provide alternative RDF serialisation formats like [Notation3](http://www.w3.org/DesignIssues/Notation3) [[*N3*](http://www.w3.org/TR/cooluris/#ref-N3)] or [TriX](http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Stickler01/EML2004Stickler01.html) [*[TriX](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-TriX)*].

**3. URIs for Real-World Objects**

On the Semantic Web, URIs identify not just Web documents, but also real-world objects like people and cars, and even abstract ideas and non-existing things like a mythical unicorn. We call these *real-world objects* or *things*.

Given such a URI, how can we find out what it identifies? We need some way to answer this question, because otherwise it will be hard to achieve interoperability between independent information systems. We could imagine a service where we can look up a description of the identified resource, similar to today's search engines. But such a single point of failure is against the Web's decentralised nature.

Instead, we should use the Web itself—an extremely robust and scalable information publishing system—as a lookup service for resource descriptions. Whenever a URI is mentioned, we can look it up to retrieve a description containing relevant information and links to related data. This is so important that we make it our number one requirement for *cool* URIs:

**1. Be on the Web.**

Given only a URI, machines and people should be able to retrieve a description about the resource identified by the URI from the Web. Such a look-up mechanism is important to establish shared understanding of what a URI identifies. Machines should get RDF data and humans should get a readable representation, such as HTML. The standard Web transfer protocol, HTTP, should be used.

Let's assume Example Inc. wants to publish contact data of their employees on the Semantic Web so their business partners can import it into their address books. For example, the published data would contain these statements about Alice, written here in [N3 syntax](http://www.w3.org/DesignIssues/Notation3) [[*N3*](http://www.w3.org/TR/cooluris/#ref-N3)]:

**<URI-of-alice>** a **foaf:Person**;

 foaf:name **"Alice"**;

 foaf:mbox **<mailto:alice@example.com>**;

 foaf:homepage **<http://www.example.com/people/alice>** .

What URI should we use instead of the placeholder <URI-of-alice>? Certainly not http://www.example.com/people/alice, because that would confuse a person with a Web document, leading to misunderstandings: Is the homepage of Alice also named “Alice”? Can a homepage itself have an e-mail address? And does it make sense for a home-page to have itself as its home-page? So we need another URI. (For in-depth treatments of this issue, see [What HTTP URIs Identify?](http://www.w3.org/DesignIssues/HTTP-URI2.html) [[*HTTP-URI2*](http://www.w3.org/TR/cooluris/#ref-HTTP-URI2)] and [Four Uses of a URL: Name, Concept, Web Location and Document Instance](http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm) [[*Booth*](http://www.w3.org/TR/cooluris/#ref-Booth)]).

Therefore our second requirement:

**2. Be unambiguous.**

There should be no confusion between identifiers for Web documents and identifiers for other resources. URIs are meant to identify only one of them, so one URI can't stand for both a Web document and a real-world object.

We note that our requirements seem to conflict with each other. If we can't use URIs of documents to identify real-world object, then how can we retrieve a representation about real-world objects based on their URI? The challenge is to find a solution that allows us to find the describing documents if we have just the resource's URI, using standard Web technologies.

The following picture shows the desired relationships between a resource and its representing documents:



**3.1 Distinguishing between Representations and Descriptions**

It is important to understand that using URIs, it is possible to identify both a thing (which may exist outside of the Web) and a Web document *describing* the thing. For example the person Alice is described on her homepage. Bob may not like the look of the *homepage*, but fancy the person Alice. So two URIs are needed, one for Alice, one for the homepage or a RDF document describing Alice. The question is where to draw the line between the case where either is possible and the case where *only* descriptions are available.

According to W3C guidelines ([[AWWW](http://www.w3.org/TR/cooluris/#ref-AWWW)], section 2.2.), we have a Web document (there called *information resource*) if *all its essential characteristics can be conveyed in a message*. Examples are a Web page, an image or a product catalog.

In HTTP, because a 200 response code should be sent when a Web document has been accessed, but a different setup is needed when publishing URIs that are meant to identify entities which are *not* Web documents.

In the next section, solutions are described that allow you to mint URIs for things and also allow clients to get a description of the thing using standard Web technologies.

**4. Two Solutions**

There are two solutions that meet our requirements for identifying real-world objects: *303 URIs* and *hash URIs*. Which one to use depends on the situation, both have advantages and disadvantages.

The solutions described in the following apply to deployment scenarios in which the RDF data and the HTML data is served separately, such as a standalone RDF/XML document along with an HTML document. The metadata can also be embedded in HTML, using technologies such as RDFa [[RDFa Primer](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-RDFaPrimer)], microformats and other documents to which the GRDDL [[GRDDL](http://www.w3.org/TR/cooluris/#ref-GRDDL)] mechanisms can be applied. In those cases the RDF data is extracted from the returned HTML document.

**4.1. Hash URIs**

The first solution is to use “hash URIs” for non-document resources. URIs can contain a *fragment*, a special part that is separated from the rest of the URI by a hash symbol (“#”).

When a client wants to retrieve a hash URI, then the HTTP protocol requires the fragment part to be stripped off before requesting the URI from the server. This means a URI that includes a hash cannot be retrieved directly, and therefore does not necessarily identify a Web document. But we can use them to identify other, non-document resources, without creating ambiguity.

If Example Inc. adopts this solution, then they could use these URIs to represent the company, Alice, and Bob:

**http://www.example.com/about#exampleinc**

Example Inc., the company

**http://www.example.com/about#bob**

Bob, the person

**http://www.example.com/about#alice**

Alice, the person

Clients will always strip off the fragment part before requesting any of these URIs, resulting in a request to this URI:

**http://www.example.com/about**

RDF document describing Example Inc., Bob, and Alice

At this URI, Example Inc. could serve an RDF document that contains descriptions of all three resources, using the original hash URIs to identify the resources.

The following picture shows the hash URI approach without content negotiation:



Alternatively, content negotiation (see [Section 2.1.](http://www.w3.org/TR/cooluris/#conneg)) could be employed to redirect from the about URI to either a HTML or an RDF representation. The decision which to return is based on client preferences and server configuration, as explained below in [Section 4.7](http://www.w3.org/TR/cooluris/#implementation). The Content-Location header should be set to indicate if the hash URI refers to a part of the HTML document or RDF document.

The following picture shows the hash URI approach with content negotiation:



**4.2. 303 URIs forwarding to One Generic Document**

The second solution is to use a special HTTP status code, 303 See Other, to give an indication that the requested resource is not a regular Web document. Web architecture tells you that for a thing resource (URI) it is inappropriate to return a 200 because there is, in fact, no suitable representation for those resources. However, it is useful to provide information about those resources. The W3C's Technical Architecture Group proposes in its [httpRange-14 resolution](http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html) [*[httpRange](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-httpRange)*] document a solution that is to direct you to a document which has information *about* the thing you asked about. By doing this we avoid ambiguity between the original, real-world object and the resource that represents it.

Since 303 is a redirect status code, the server can give the location of a document that represents the resource. If, on the other hand, a request is answered with one of the usual status codes in the 2XX range, like 200 OK, then the client knows that the URI identifies a Web document.

If Example Inc. adopts this solution, they could use these URIs to represent the company, Alice and Bob:

**http://www.example.com/id/exampleinc**

Example Inc., the company

**http://www.example.com/id/bob**

Bob, the person

**http://www.example.com/id/alice**

Alice, the person

The Web server would be configured to answer requests to all these URIs with a 303 status code and a Location HTTP header that provides the URL of a document that represents the resource.For example, to redirect from http://www.example.com/id/alice to http://www.example.com/doc/alice.

Content-negotiation is then used when retrieving a representation from the document URI using a HTTP request. The server decides (see [Section 4.7](http://www.w3.org/TR/cooluris/#implementation)) to return either HTML or RDF (or more alternative forms) and sets the Content-Location header to the URI where the specific representation can be retrieved.

This setup should be used when the RDF and HTML (and possibly more alternative representations) convey the *same information in different forms*. When the information in the variations differs considerably, the 303 approach as described [below](http://www.w3.org/TR/cooluris/#r303uri) should be used.

See the following illustration for the solution providing the generic document URI.



In this setup, the server forwards from the identification URI to the generic document URI. This has the advantage that clients can bookmark and further work with the generic document. A user having a RDF-capable client could bookmark the document, and mail it to another user (or device) which then dereferences it and gets the HTML *or* the RDF view. Also, the server can add representations in new languages in the future. Just because the client started with the URI of a thing, it doesn't mean that the document involved is not a first class document on the WWW. The background of generic document resources is described in [[GenRes](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-GenRes)].

**4.3. 303 URIs forwarding to Different Documents**

When the RDF and HTML representations of the resource differ substantially, the previous setup should not be used. They are not two versions of the same document, but different documents altogether. Again, the Web server would be configured to answer requests with a 303 status code and a Location HTTP header that provides the URL of a document that represents the resource.

The following picture shows the redirects for the 303 URI solution without the generic document URI:



The server could employ content negotiation (see [Section 2.1.](http://www.w3.org/TR/cooluris/#conneg)) to send either the URL of an HTML description or RDF. HTTP requests for HTML content would be redirected to the HTML URLs we gave in [Section 2](http://www.w3.org/TR/cooluris/#oldweb). Requests for RDF data would be redirected to RDF documents, such as:

**http://www.example.com/data/exampleinc**

RDF document describing Example Inc., the company

**http://www.example.com/data/bob**

RDF document describing Bob, the person

**http://www.example.com/data/alice**

RDF document describing Alice, the person

Each of the RDF documents would contain statements about the appropriate resource, using the original URI, e.g. http://www.example.com/id/alice, to identify the described resource.

**4.4. Choosing between 303 and Hash**

Which approach is better? It depends. The hash URIs have the advantage of reducing the number of necessary HTTP round-trips, which in turn reduces access latency. A family of URIs can share the same non-hash part. The descriptions of http://www.example.com/about#exampleinc, http://www.example.com/about#alice, and http://www.example.com/about#bob are retrieved with a single request to http://www.example.com/about. However this approach has a downside. A client interested only in #product123 will inadvertently load the data for all other resources as well, because they are in the same file. 303 URIs, on the other hand, are very flexible because the redirection target can be configured separately for each resource. There could be one describing document for each resource, or one large document for all of them, or any combination in between. It is also possible to change the policy later on.

When using 303 URIs for an ontology, like FOAF, network delay can reduce a client's performance considerable. The large number of redirects may cause higher latency. A client looking up a set of terms through 303 may use many requests, even though the first request has already loaded everything there is to know.

When hosting large-scale datasets with the 303 solution, clients may be tempted to download all data using many requests. We advise to additionally provide SPARQL endpoints or comparable services to answer complex queries on the server directly, rather than to let the client download a large set of data via HTTP.

Note also, that both *303 and Hash can be combined*, allowing a large dataset to be separated  into multiple parts and have an identifier for a non-document resource. An example for a combination of 303 and Hash is:

**http://www.example.com/bob#this**

Bob, the person with a combined URI.

Any fragment identifier is valid, this in the above URI is a suggestion you may want to copy for your implementations.

**Conclusion.**

Hash URIs should be preferred for rather small and stable sets of resources that evolve together. The ideal case are RDF Schema vocabularies and OWL ontologies, where the terms are often used together, and the number of terms is unlikely to grow out of control in the future.

Hash URIs without content negotiation can be implemented by simply uploading static RDF files to a Web server, without any special server configuration. This makes them popular for quick-and-dirty RDF publication.

URIs of the bob#this form can be used for large sets of data that are, or may grow, beyond the point where it is practical to serve all related resources in a single document. 303 URIs may also be used for such data sets, making neater-looking URIs, but with an impact on run-time performance and server load.

If in doubt, follow your nose.

**4.5. Cool URIs**

The best resource identifiers don't just provide descriptions for people and machines, but are designed with simplicity, stability and manageability in mind, as explained by Tim Berners-Lee in [*Cool URIs don't change*](http://www.w3.org/Provider/Style/URI) and by the W3C Team in [*Common HTTP Implementation Problems*](http://www.w3.org/TR/2003/NOTE-chips-20030128/) (sections 1 and 3):

**Simplicity.**

Short, mnemonic URIs will not break as easily when sent in emails and are in general easier to remember, e.g. when debugging your Semantic Web server.

**Stability.**

Once you set up a URI to identify a certain resource, it should remain this way as long as possible. Think about the next ten years. Maybe twenty. Keep implementation-specific bits and pieces such as .php and .asp out of your URIs, you may want to change technologies later.

**Manageability.**

Issue your URIs in a way that you can manage. One good practice is to include the current year in the URI path, so that you can change the URI-schema each year without breaking older URIs. Keeping all 303 URIs on a dedicated subdomain, e.g. http://id.example.com/alice, eases later migration of the URI-handling subsystem.

**4.6. Linking**

All the URIs related to a single real-world object—resource identifier, RDF document URL, HTML document URL—should also be explicitly linked with each other to help information consumers understand their relation. For example, in the 303 URI solution for Example Inc., there are three URIs related to Alice:

**http://www.example.com/id/alice**

Identifier for Alice, the person

**http://www.example.com/people/alice**

Alice's homepage

**http://www.example.com/data/alice**

RDF document with description of Alice

Two of them are Web document URLs. The RDF document located at http://www.example.com/data/alice might contain these statements (expressed in N3):

**<http://www.example.com/id/alice>**

 foaf:page **<http://www.example.com/people/alice>**;

 rdfs:isDefinedBy **<http://www.example.com/data/alice>**;

 a **foaf:Person**;

 foaf:name **"Alice"**;

 foaf:mbox **<mailto:alice@example.com>**;

 ...

The document makes statements about Alice, the person, using the resource identifier. The first two properties relate the resource identifier to the two document URIs. The foaf:page statement links it to the HTML document. This allows RDF-aware clients to find a human-readable resource, and at the same time, by linking the page to its topic, defines useful metadata about that HTML document. The rdfs:isDefinedBy statement links the person to the document containing its RDF description and allows RDF browsers to distinguish this main resource from other auxiliary resources that just happen to be mentioned in the document. We use rdfs:isDefinedBy instead of its weaker superproperty rdfs:seeAlso because the content at /data/alice is authoritative. The remaining statements are the actual white pages data.

The HTML document at http://www.example.com/people/alice should contain in its header a <link> element that points to the corresponding RDF document:

<html xmlns="http://www.w3.org/1999/xhtml" lang="en">

 <head>

 <title>Alice's Homepage</title>

 <link rel="alternate" type="application/rdf+xml"

 title="RDF Representation"

 href="**http://www.example.com/data/alice**" />

 </head> ...

This allows RDF-aware Web clients to discover the RDF information. The approach is [recommended in the RDF/XML specification](http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/#section-rdf-in-HTML) ([[*RDFXML*](http://www.w3.org/TR/cooluris/#ref-RDFXML)], section 9). If the RDF data is *about* the Web page, rather than an expression of the information in it, then we recommend using rel="meta" instead of rel="alternate".

The client also can deduce similar link information directly from the HTTP headers: that a thing is described by a Web document which can be found at the end of a 303 redirect; that the Content-Location resource is a content-specific version of the generic document, and more. Ontologies for these relations are not discussed here.

The following illustration shows how the RDF and HTML documents should relate the three URIs to each other:



**4.7. Implementing Content Negotiation**

The W3C's Semantic Web Best Practices and Deployment Working Group has published a document that describes how to implement the solutions presented here on the Apache Web server. The [*Best Practice Recipes for Publishing RDF Vocabularies*](http://www.w3.org/TR/swbp-vocab-pub/) [[*Recipes*](http://www.w3.org/TR/cooluris/#ref-Recipes)] mostly discuss the publication of *RDF vocabularies*, but the ideas can also be applied to other kinds of small RDF datasets that are published from static files.

However, especially when it comes to content negotiation, the Recipes document doesn't cover some important details. Content negotiation is a bit more difficult in practice because of mixed-mode clients that can deal with both HTML and RDF, such as Firefox with the [Tabulator extension](http://dig.csail.mit.edu/2007/tab/).

These browsers announce their ability to consume both RDF and HTML through Accept headers that use q (quality) values:

Accept: application/rdf+xml;q=0.7, text/html

This browser accepts RDF with a q value of 0.7 and HTML with a q value of 1.0 (the default). This means the browser has a slight preference for HTML over RDF.

Now, a client preference for HTML doesn't necessarily mean that every server should send HTML. The server has to look at the client's preferences, and then it must make a decision based on the quality of the different variants it could offer. For example:

* If the HTML variant is a simple low-quality rendering of the RDF, like a property-value table or a list of triples, then the server should send the RDF, unless the client has a very strong preference for HTML.
* If HTML and RDF variant contain the same information, and both are of high quality, then the server should treat both variants with equal preference, and leave the choice to the client's preferences.
* If the RDF variant is only a part of the information offered in the HTML, or is scraped from the HTML, then the server should probably send the HTML, unless the client has a strong preference for RDF.

There are algorithms for choosing the best match by comparing client preferences with the quality of the server's available variants. For example, the Apache server can be configured with server-side qs values that specify their relative quality.

A qs value of 1.0 for application/rdf+xml and 0.5 for text/html, would mean that the HTML variant has only approximately half the quality of the RDF and might be appropriate in the first case from the list above. If the HTML is a news article and the RDF contains just minimal information such as title, date and author, then 1.0 for the HTML and 0.1 for the RDF would be appropriate.

To determine the best variant for a particular client, Apache multiplies the client's q value for HTML with the configured qs value for HTML; and the same for RDF. The variant with the higher number wins. Apache's documentation has a [section](http://httpd.apache.org/docs/2.0/content-negotiation.html#methods) with a detailed description of its content negotiation algorithm [[ApCN](http://www.w3.org/TR/cooluris/%22%20%5Cl%20%22ref-ApCN)]. HTTP's Accept header is described in detail in [section 14.1](http://www.rfc.net/rfc2616.html#s14.1) of the HTTP specification [[HTTP-SPEC](http://www.w3.org/TR/cooluris/#ref-HTTP-SPEC)].

Content negotiation, with all its details, is fairly complex, but it is a powerful way of choosing the best variant for mixed-mode clients that can deal with HTML and RDF.

**5. Examples from the Web**

Not all projects that work with Semantic Web technologies make their data available on the Web. But a growing number of projects follow the practices described here. This section gives a few examples.

**ECS Southampton.** The [School of Electronics and Computer Science](http://www.ecs.soton.ac.uk/) at University of Southampton has a Semantic Web site that employs the 303 solution and is a great example of Semantic Web engineering. It is documented in the [*ECS URI System Specification*](http://id.ecs.soton.ac.uk/docs/) [[*ECS*](http://www.w3.org/TR/cooluris/#ref-ECS)]. Separate subdomains are used for HTML documents, RDF documents, and resource identifiers. Take these examples:

**http://id.ecs.soton.ac.uk/person/1650**

URI for Wendy Hall, the person

**http://www.ecs.soton.ac.uk/people/wh**

HTML page about Wendy Hall

**http://rdf.ecs.soton.ac.uk/person/1650**

RDF about Wendy Hall

Entering the first URI into a normal Web browser redirects to an HTML page about Wendy Hall. It presents a Web view of all available data on her. The page also links to her URI and to her RDF document.

[**D2R Server**](http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/) is an open-source application that can be used to publish data from relational databases on the Semantic Web in accordance with these guidelines. It employs the 303 solution and content negotiation. For example, the [*D2R Server publishing the DBLP Bibliography Database*](http://www4.wiwiss.fu-berlin.de/dblp/) publishes several thousand bibliographical records and information about their authors. Example URIs, again connected via 303 redirects:

**http://www4.wiwiss.fu-berlin.de/dblp/resource/person/315759**

URI for Chris Bizer, the person

**http://www4.wiwiss.fu-berlin.de/dblp/page/person/315759**

HTML page about Chris Bizer

The RDF document for Chris Bizer is a SPARQL query result from the server's SPARQL endpoint:

http://www4.wiwiss.fu-berlin.de/dblp/sparql?query=

DESCRIBE+\%3Chttp\%3A\%2F\%2Fwww4.wiwiss.fu-berlin.de

\%2Fdblp\%2Fresource\%2Fperson\%2F315759\%3E

The SPARQL query encoded in this URI is:

DESCRIBE <http://www4.wiwiss.fu-berlin.de/dblp/resource/person/315759>

This shows how a SPARQL endpoint can be used as a convenient method of serving resource descriptions.

[**Semantic MediaWiki**](http://semantic-mediawiki.org/wiki/Semantic_MediaWiki) is an open-source Semantic wiki engine. Authors can use special wiki syntax to put semantic attributes and relationships into wiki articles. For each article, the software generates a 303 URI that identifies the article's topic, and serves RDF descriptions generated from the attributes and relationships. Semantic MediaWiki drives the [OntoWorld wiki](http://ontoworld.org/wiki/Main_Page). It has an article about the city of Karlsruhe:

**http://ontoworld.org/wiki/Karlsruhe**

the article, an HTML document

**http://ontoworld.org/wiki/\_Karlsruhe**

the city of Karlsruhe

**http://ontoworld.org/wiki/Special:ExportRDF/Karlsruhe**

RDF description of Karlsruhe

The URI of the RDF description is less than ideal, because it exposes the implementation (php) and refers redundantly to RDF in the path and in the query. A much cooler URI would be for example http://ontoworld.org/data/Karlsruhe, as it allows content negotiation to be used to serve the data in RDF, RIF (Rule Interchange Format), or whatever else we think of next.

**6. Other Resource Naming Proposals**

Many other approaches have been suggested over the years. While most of them are appropriate in special circumstances, we feel that they do not fit the criteria from [Section 3](http://www.w3.org/TR/cooluris/#semweb), which are to *be on the Web* and *don't be ambiguous*. Therefore they are not adequate as general solutions for building a standards-based, non-fragmented, decentralized Semantic Web. We will discuss two of these approaches in some detail.

**6.1. New URI Schemes**

HTTP URIs already identify Web resources and Web documents, not other kinds of resources. Shouldn't we create a new URI scheme to identify other resources? Then we could easily distinguish them from Web documents just by looking at the first characters of the URI. For example, the *info* scheme can be used to identify books based on a LCCN number: info:lccn/2002022641.

Here are examples of such new URI schemes. A longer list is provided by Thompson and Orchard in [*URNs, Namespaces and Registries*](http://www.w3.org/2001/tag/doc/URNsAndRegistries-50-2006-08-17.html) [[*TAG-URNs*](http://www.w3.org/TR/cooluris/#ref-TAG-URNs)].

* [**Magnet**](http://magnet-uri.sourceforge.net/) is an open URI scheme enabling seamless integration between Web sites and locally-running utilities, such as file-management tools. It is based on hash-values, a URI looks like this:
magnet:?xt=urn:sha1:YNCKHTQCWBTRNJIV4WNAE52SJUQCZO5C.
* The [**info: URI scheme**](http://www.ietf.org/rfc/rfc4452.txt) is proposed to identify information assets that have identifiers in existing public namespaces. Examples are URIs for LCCN numbers (info:lccn/2002022641) and the Dewey decimal system (info:ddc/22/eng//004.678).
* The idea of [**Tag URIs**](http://www.taguri.org/) is to generate collision-free URIs by using a domain name and the date when the URI was allocated. Even if the domain changes ownership at a later date, the URI remains unambiguous. Example: tag:hawke.org,2001-06-05:Taiko.
* [**XRI**](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xri) defines a scheme and resolution protocol for abstract identifiers. The idea is to use URIs that contain wildcards, to adapt to changes of organizations, servers, etc.
Examples are @Jones.and.Company/(+phone.number) or xri://northgate.library.example.com/(urn:isbn:0-395-36341-1).

To be truly useful, a new scheme must be accompanied by a protocol defining how to access more information about the identified resource. For example, the ftp:// URI scheme identifies resources (files on an FTP server), and also comes with a protocol for accessing them (the FTP protocol).

Some of the new URI schemes provide no such protocol at all. Others provide a Web Service that allows retrieval of descriptions using the HTTP protocol. The identifier is passed to the service, which looks up the information in a central database or in a federated way. The problem here is that a failure in this service renders the system unusable.

Another drawback can be a dependence on a standardization body. To register new parts in the info: space, a standardization body has to be contacted. This, or paying a license fee before creating a new URI, slows down adoption. In such cases a standardization body is desirable to ensure that all URIs are unique (e.g. with ISBNs). But this can be achieved using HTTP URIs inside an HTTP namespace owned and managed by the standardization organization.

Independent of standardization body and retrievability, pending patents and legal issues can influence the adoption of a new URI scheme.  When using patented technology, implementers should verify that a Royalty-Free license is available.

The problems with new URI schemes are discussed at length in [*URNs, Namespaces and Registries*](http://www.w3.org/2001/tag/doc/URNsAndRegistries-50.html).

**6.2. Reference by Description**

*"Reference by Description"* radically solves the URI problem by doing away with URIs altogether: Instead of *naming* resources with a URI, *anonymous nodes* are used, and are *described* with information that allows us to find the right one. A person, for example, could be described by name, date of birth, and social security number. These pieces of information should be sufficient to uniquely identify a person.

A popular practice is the use of a person's email address as a uniquely identifying piece of information. The foaf:mbox property is used in [Friend of a Friend](http://xmlns.com/foaf/spec/20070524.html) ([*FOAF*](http://www.w3.org/TR/cooluris/#ref-FOAF)) profiles for this purpose. In OWL, this kind of property is known as an *Inverse Functional Property* (IFP). When an agent encounters two resources with the same email address, it can infer that both refer to the same person and can treat them as one.

But how to *be on the Web* with this approach? How to enable agents to download more data about resources we mention? There is a best practice to achieve this goal: Provide not only the IFP of the resource (e.g. the person's email address), but also an rdfs:seeAlso property that points to a Web address of an RDF document with further information about it. We see that HTTP URIs are still used to identify the location where more information can be downloaded.

Furthermore, we now need several pieces of information to refer to a resource, the IFP value and the RDF document location. The simple act of linking by using a URI has become a process involving several moving parts, and this increases the risk of broken links and makes implementation more cumbersome.

Regarding FOAF's practice of avoiding URIs for people, we agree with [Tim Berners-Lee's advice](http://dig.csail.mit.edu/breadcrumbs/node/71): “Go ahead and give yourself a URI. You deserve it!”

**7. Conclusion**

Resource names on the Semantic Web should fulfill two requirements: First, a description of the identified resource should be retrievable with standard Web technologies. Second, a naming scheme should not confuse things and the documents representing them.

We have described two approaches that fulfill these requirements, both based on the HTTP URI scheme and protocol. One is to use the 303 HTTP status code to redirect from the resource identifier to the describing document. One is to use “hash URIs” to identify resources, exploiting the fact that hash URIs are retrieved by dropping the part after the hash and retrieving the other part.

The requirement to distinguish between resources and their descriptions increases the need for coordination between multiple URIs. Some useful techniques are: embedding links to RDF data in HTML documents, using RDF statements to describe the relationship between the URIs, and using content negotiation to redirect to an appropriate description of a resource.

**8. Acknowledgements**

Many thanks to Tim Berners-Lee who invested much time and helped us understanding the [TAG](http://www.w3.org/2001/tag/) solution by answering [chat requests](http://chatlogs.planetrdf.com/swig/2006-10-29#T17-42-28) and contributing many emails with clarifications and detailled reviews of this document. Special thanks go to Stuart Williams, Norman Walsh and all the other members from TAG, who reviewed this document and provided essential feedback in [June 2007](http://lists.w3.org/Archives/Public/www-tag/2007Jun/0075.html) and [September 2007](http://lists.w3.org/Archives/Public/www-tag/2007Sep/0090.html) about many formulations that were (accidentially) contrary to the TAG's view. Also special thanks to the [Semantic Web Deployment Group](http://www.w3.org/2006/07/SWD/)'s members Michael Hausenblas, Vit Novacek, and Ed Summers' reviews and their review summary sent in [October 2007](http://www.w3.org/2006/07/SWD/wiki/ReviewCoolURIs). We wish to thank everyone else who has reviewed drafts of this document, especially Chris Bizer, Gunnar AAstrand Grimnes, Harry Halpin, Xiaoshu Wang, Henry S. Thompson, Jonathan Rees, and Christoph Päper. Susie Stephens reviewed the document, managed SWEO, and helped us to stay on track. Ivan Herman did much to verify that the W3C requirements are met and submitted the note.

This work was supported by the German Federal Ministry of Education, Science, Research and Technology (BMBF), (Grants 01 IW C01, Project EPOS: Evolving Personal to Organizational Memories; and 01 AK 702B, Project InterVal: Internet and Value Chains) and by the European Union IST fund (Grant FP6-027705, Project Nepomuk).

**9. References**

**[AWWW]**

[*Architecture of the World Wide Web, Volume One*](http://www.w3.org/TR/2004/REC-webarch-20041215/), Ian Jacobs, Norman Walsh, Editors. World Wide Web Consortium, 15 December 2004. This edition is http://www.w3.org/TR/2004/REC-webarch-20041215/. The [latest edition](http://www.w3.org/TR/webarch/) is available at <http://www.w3.org/TR/webarch/>.

**[****ApCN]**

[Apache HTTP Server Version 2.0 Documentation, Chapter Content Negotiation](http://httpd.apache.org/docs/2.0/content-negotiation.html). This document is available at <http://httpd.apache.org/docs/2.0/content-negotiation.html>.

**[Booth]**

[*Four Uses of a URL: Name, Concept, Web Location and Document Instance*](http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm), David Booth. 28 January 2003. This document is available at http://www.w3.org/2002/11/dbooth-names/dbooth-names\_clean.htm.

**[CHIPS]**

[*Common HTTP Implementation Problems*](http://www.w3.org/TR/2003/NOTE-chips-20030128/), Olivier Théreaux, Editor. World Wide Web Consortium, 28 January 2003. This edition is http://www.w3.org/TR/2003/NOTE-chips-20030128/. The [latest edition](http://www.w3.org/TR/2003/NOTE-chips-20030128/) is available at http://www.w3.org/TR/chips/.

**[Cool]**

[*Cool URIs don't change*](http://www.w3.org/Provider/Style/URI), Tim Berners-Lee, 1998. This document is available at <http://www.w3.org/Provider/Style/URI>.

**[****DP]**

The DataPortability Project. <http://dataportability.org/>

**[ECS]**

[*ECS URI System Specification*](http://id.ecs.soton.ac.uk/docs/), Colin Williams, Nick Gibbins. ECS Southampton, 2006. This document is available at <http://id.ecs.soton.ac.uk/docs/>.

**[FOAF]**

[*FOAF Vocabulary Specification 0.9*](http://xmlns.com/foaf/spec/20070524.html), Dan Brickley, Libby Miller. 24 May 2007. This edition is http://xmlns.com/foaf/spec/20070524.html. The [latest edition](http://xmlns.com/foaf/spec/) is available at <http://xmlns.com/foaf/spec/>.

**[****Give]**

[*Give Us the Data Raw, and Give it to Us Now*](http://blog.okfn.org/2007/11/07/give-us-the-data-raw-and-give-it-to-us-now/). Rufus Pollock. 7th November 2007.

**[****GenRes]**

[*Generic Resources*](http://www.w3.org/DesignIssues/Generic.html), Tim Berners-Lee. This document is available at <http://www.w3.org/DesignIssues/Generic.html>.

**[****GRDDL]**

[Gleaning Resource Descriptions from Dialects of Languages (GRDDL)](http://www.w3.org/TR/grddl/), Dan Connolly, Editor, W3C Recommendation 11 September 2007. This edition is http://www.w3.org/TR/2007/REC-grddl-20070911/. The latest edition is available at <http://www.w3.org/TR/grddl/>.

**[HTTP-URI2]**

[*What HTTP URIs Identify*](http://www.w3.org/DesignIssues/HTTP-URI2.html), Tim Berners-Lee. 9 June 2005. This document is available at http://www.w3.org/DesignIssues/HTTP-URI2.html.

**[httpRange]**

[*[httpRange-14] Resolved*](http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html), Roy Fielding. 18 June 2005. This archived [www-tag](http://lists.w3.org/Archives/Public/www-tag/) email message is available at <http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html>.

**[****HTTP-SPEC]**

[RFC2616](http://www.rfc.net/rfc2616.html#s14.1), Hypertext Transfer Protocol -- HTTP/1.1, <http://www.rfc.net/rfc2616.html#s14.1>

**[N3]**

[*Notation 3*](http://www.w3.org/DesignIssues/Notation3), Tim Berners-Lee, Dan Connolly, 2008. This document is available at <http://www.w3.org/TeamSubmission/n3/>.

**[****N3Primer]**

Primer: Getting into RDF & Semantic Web using N3. Tim Berners-Lee, 2005. <http://www.w3.org/2000/10/swap/Primer>

**[****RDFa Primer]**

RDFa Primer 1.0 - Embedding Structured Data in Web Pages (see <http://www.w3.org/2006/07/SWD/RDFa/primer/>.)

**[RDFPrimer]**

[*RDF Primer*](http://www.w3.org/TR/2004/REC-rdf-primer-20040210/), Frank Manola, Eric Miller, Editors. World Wide Web Consortium, 10 February 2004. This edition is http://www.w3.org/TR/2004/REC-rdf-primer-20040210/. The [latest edition](http://www.w3.org/TR/rdf-primer/) is available at http://www.w3.org/TR/rdf-primer/.

**[RDFXML]**

[*RDF/XML Syntax Specification (Revised)*](http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/), Dave Beckett, Editor. World Wide Web Consortium, 10 February 2004. This edition is http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/. The [latest edition](http://www.w3.org/TR/rdf-syntax-grammar/) is available at http://www.w3.org/TR/rdf-syntax-grammar/.

**[Recipes]**

[*Best Practice Recipes for Publishing RDF Vocabularies*](http://www.w3.org/TR/2008/WD-swbp-vocab-pub-20080123/), Alistair Miles, Thomas Baker, Ralph Swick, Editors. World Wide Web Consortium, 23 January 2008. This edition is http://www.w3.org/TR/2008/WD-swbp-vocab-pub-20080123/. It is a work in progress. The [latest edition](http://www.w3.org/TR/swbp-vocab-pub/) is available at http://www.w3.org/TR/swbp-vocab-pub/.

**[RFC2616]**

[*RFC 2616: Hypertext Transfer Protocol - HTTP/1.1*](http://www.ietf.org/rfc/rfc2616.txt), J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. IETF, 1999. This document is available at http://www.ietf.org/rfc/rfc2616.txt.

**[RFC3986]**

[*RFC 3986: Uniform Resource Identifier (URI): Generic Syntax*](http://www.ietf.org/rfc/rfc3986.txt), T. Berners-Lee, R. Fielding, L. Masinter. IETF, 2005. This document is available at http://www.ietf.org/rfc/rfc3986.txt.

**[SMW]**

[*Semantic Wikipedia*](http://www.aifb.uni-karlsruhe.de/WBS/hha/papers/SemanticWikipedia.pdf), Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, Rudi Studer. University of Karlsruhe, 2006. This document is available at http://www.aifb.uni-karlsruhe.de/WBS/hha/papers/SemanticWikipedia.pdf.

**[TAG-Alt]**

[*On Linking Alternative Representations To Enable Discovery And Publishing*](http://www.w3.org/2001/tag/doc/alternatives-discovery-20061101.html), T.V. Raman. World Wide Web Consortium, 1 November 2006. This edition is http://www.w3.org/2001/tag/doc/alternatives-discovery-20061101.html. The [latest edition](http://www.w3.org/2001/tag/doc/alternatives-discovery.html) is available at http://www.w3.org/2001/tag/doc/alternatives-discovery.html.

**[TAG-URNs]**

[*URNs, Namespaces and Registries*](http://www.w3.org/2001/tag/doc/URNsAndRegistries-50-2006-08-17.html), Henry S. Thompson, David Orchard. World Wide Web Consortium, 17 August 2006. This edition is http://www.w3.org/2001/tag/doc/URNsAndRegistries-50-2006-08-17.html. It is a work in progress. The [latest edition](http://www.w3.org/2001/tag/doc/URNsAndRegistries-50.html) is available at http://www.w3.org/2001/tag/doc/URNsAndRegistries-50.html.

**[TriX]**

[*RDF Triples in XML*](http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Stickler01/EML2004Stickler01.html), Jeremy J. Carroll, Patrick Stickler, 2004. This document is available at http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Stickler01/EML2004Stickler01.html.

**[WP-HTTP]**

[*Hypertext Transfer Protocol*](http://en.wikipedia.org/wiki/HTTP), Wikipedia contributors. Wikipedia, 8 October 2007. The latest version of this document is available at http://en.wikipedia.org/wiki/HTTP.

**10. Change log**

**29 November 2006**

1.0 Initial Version.

**9 August 2007**

1.1 Revised Version. Changes based on [TAG review](http://lists.w3.org/Archives/Public/www-tag/2007Jun/0075.html).

**28 November 2007**

Leo Sauermann included more feedback from reviews contributed by TAG, SWD, and Tim Berners-Lee.

**8 December 2007**

Danny Ayers did proofreading, minor grammar/idiomatic/editorial changes (I've tried not to make any changes that substantively modify the content, though some come close...). XHMTL validated with nxml-mode emacs

**12 December 2007**

Leo Sauermann included link to GRDDL as suggested by Danny Ayers, minor changes of todo notes. Document was remodelled to Working Draft status - all feedback by SWD, TAG, and Tim Berners Lee either has been addressed or is listed in this document as todos using @@-symbols and the css class "todo".

**17 December 2007**

Document published as Working Draft at <http://www.w3.org/TR/2007/WD-cooluris-20071217/>

**23 Februar 2008**

All feedback received on Working Draft.

**20 March 2008**

All feedback incorporated, issues are listed and addressed in [this document](https://gnowsis.opendfki.de/repos/gnowsis/papers/2006_11_concepturi/feedback/index.htm#feedback_v20071217).

**21 March 2008**

Document published as Last Call Working Draft at <http://www.w3.org/TR/2008/WD-cooluris-20080321/>

**31 March 2008**

Document published as Interest Group Note. Feedback to previous version and changes are [listed here](https://gnowsis.opendfki.de/repos/gnowsis/papers/2006_11_concepturi/feedback/index.htm#feedback_v20080321).