# EutilsRDF Web Service: an RDF interface to NCBI Entrez Utilities

Chris Maloney, 2013-08-13

# Introduction

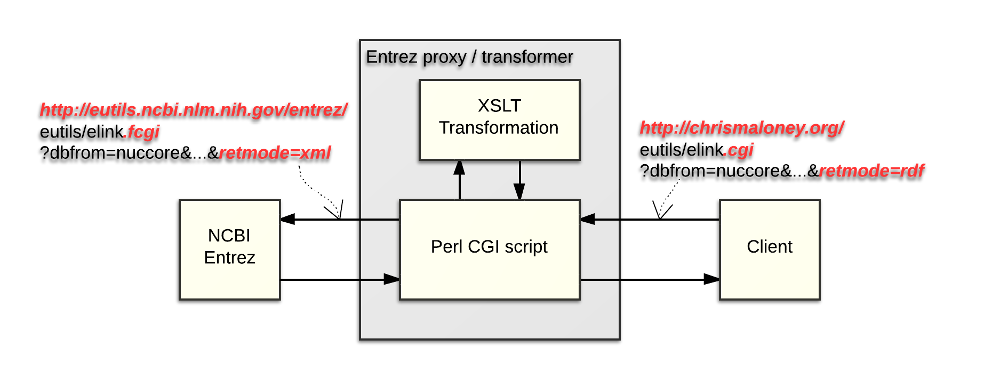
The NCBI E-utilities are a set of APIs that provide programmatic search-and-retrieval access to a large set of NCBI databases, including nucleotide and protein sequence databases (including GenBank), scientific journal article abstract, citation, and full-text databases (PubMed and PMC), and many others (52 at the time of this writing). Currently, E-utilities provides output in a few different formats, including text and XML, but not in RDF (NCBI 2010 [1]).

In this project, I have created a web service and a framework for the enhancement of NCBI E-utilities to provide an RDF response format. In this format, the data from the NCBI databases will be presented as RDF triples, using a mixture of a standardized NCBI vocabulary (new URI patterns) in a newly developed ontology, and existing, third-party, ontologies.

This service is currently deployed to http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/. The home page provides links to other project resources, documentation, and examples.

## The web service

The service is a set of simple Perl CGI scripts that proxy HTTP requests that mimic the calling convention (CGI parameters) of E-utilities, send them to NCBI E-utilities to retrieve XML results, and then pass those results through an XSLT transformation, to produce RDF. This is illustrated in the following figure.



*Figure 1. Web service block diagram.*

To get an RDF representation of an E-utilities response, you first need to identify the URL of the NCBI E-utilities request, and then translate that into the form expected by this service. For example, consider the URL

<http://eutils.ncbi.nlm.nih.gov/entrez/eutils/einfo.fcgi>

To translate this to an EutilsRDF service URL:

* Change the base part from http://eutils.ncbi.nlm.nih.gov/entrez/eutils/ to http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/,
* Change .fcgi to .cgi, and
* Add ?retmode=rdf.

Resulting in

<http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/einfo.cgi?retmode=rdf>

If the NCBI E-utilities URL already has a retmode parameter, then change its value to rdf. Another example:

<http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esummary.fcgi?retmode=xml&db=pmc&id=14900&version=2.0>

would become

<http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/esummary.cgi?retmode=rdf&db=pmc&id=14900&version=2.0>

## The project

The work was done in the open, on GitHub [here](https://github.com/Klortho/eutilsrdf), and with some engagement and feedback from staff at NCBI and the wider Semantic Web / Bioinformatics community.

This project has included the development of:

* A standard for NCBI RDF URIs;
* A web server framework for transforming the E-utililities results into RDF, including:
  + A driver Perl CGI script,
  + XLST transformations
* A design for the representation of a sample of E-utilities responses in RDF. This design includes the use of existing ontologies, as well as the development of a custom ontology;
* Documentation and examples.

The following are the various web pages associated with this project:

* + [GitHub repository](https://github.com/Klortho/EutilsRDF)
  + [Documentation](https://github.com/Klortho/EutilsRDF/wiki) (GitHub wiki)
  + [Issues](https://github.com/Klortho/EutilsRDF/issues) - tasks and bug reports

In addition, the [eutilsrdf Google group](https://groups.google.com/forum/#!forum/eutilsrdf) has been set up to provide a mailing list for discussions.

NCBO BioPortal (NCBO [2]) is a web site / service for registering Bioinformatics / Semantic Web projects and ontologies. This project has been registered with BioPortal [here](http://bioportal.bioontology.org/projects/257).

## Potential benefit

The network of linked open data in the bioinformatics realm is growing steadily, and it is hoped that this project could act as a catalyst to NCBI’s taking steps in the direction of providing more data in RDF format, and thereby more tightly integrate their resources with the rest of the Semantic Web. That would, in turn, make them easier to use for the discovery of new and interesting relationships, and perhaps quicken the pace of scientific discovery.

NCBI provides a massive amount of free, openly available data related to biology and medicine. The NCBI E-utilities are the primary API for accessing this data. However, the data provided by this API are not truly part of the network of linked open data as envisioned by Tim Berners-Lee, because:

* The format is not standard and not tightly controlled. The E-utilities addressed in this document are provided in a custom XML format, with ad-hoc element and attribute names.
* Standard forms for identifiers are not global and not clearly defined. For example, what is the standard, canonical, globally-unique, and stable identifier for a given PubMed abstract? They appear in various places in many different forms, and this ambiguity provides a barrier to reliably linking data sets.
* They do not provide links to other datasets in an unambiguous fashion.

Among the goals of this project are to conform to the principles outlined by Tim Berners-Lee for linked open data (Berners-Lee 2006 [3]), which I will quote here in full. See the reference for a discussion of the benefits of providing data in this way.

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF\*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

# NCBI RDF URIs

The initial phase of this project involved working with people inside and outside of NCBI to develop a document to standardize RDF URIs for NCBI. One of the goals of this project was to develop tools that might be incorporated into the real, production NCBI APIs, and so considerable up-front effort was expended on designing URIs for resources according to the widely-recognized current best practices. For example, those described in [Cool URIs for the Semantic Web](http://www.w3.org/TR/cooluris/) (W3C 2008 [4]) and [Linked Data: Evolving the Web into a Global Data Space](http://linkeddatabook.com/editions/1.0/) (Heath and Bizer 2011 [5]).

The result of this phase was the NCBI RDF URI standards document, which is included as Appendix A. (At the time of this writing, this document is under review, and has not been officially approved in any way.)

This standard was used in the minting of URIs for the ontologies (described below) for NCBI database entities (which will be used as RDF subjects and objects) and Entrez filters and links (which will be used as RDF predicates).

Among the challenges during this phase of the project were engaging the appropriate people (who are all very busy) and convincing them of the value of this effort. Another problem was in reaching consensus about certain recommendations, such as the use of slash- vs. hash-namespace URIs. In that particular issue, it was decided to allow for either, but to recommend the use of slash-namespace URIs (see the standards document for rationale.)

# RDF URI resolution

Point three of Tim Berners-Lee's recommendations for linked open data is to allow users to dereference RDF URIs, and provide useful information to them when they do so. To achieve this, I implemented another, closely related project, which is the NCBI [RDF URI Resolver](https://github.com/Klortho/rdf-uri-resolver) (also on GitHub).

This tool provides a simple mechanism to specify, in a distributed way, the way any URI within the "http://rdf.ncbi.nlm.nih.gov" namespace is handled. The [config.xml](https://github.com/Klortho/rdf-uri-resolver/blob/master/config.xml) specifies, on a project-by-project basis, what should occur when someone accesses on of these URIs in their browser.

All of the URIs discussed in this paper are within the "entrez" project (that is, they begin with "http://rdf.ncbi.nlm.nih.gov/entrez/"). The configuration file for the resolver tool specifies three rules:

1. If the URI is for an entrez:Db (for example, http://rdf.ncbi.nlm.nih.gov/entrez/db/pmc) then redirect the client to the RDF format of the detailed EInfo response for that database
2. If the URI is for an entrez:Link (for example, http://rdf.ncbi.nlm.nih.gov/entrez/link/pmc\_nuccore) then also redirect to the EInfo detailed response (because it contains information about that link)
3. Otherwise, redirect to [the ontology file](http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/entrez-ontology.xml).

# Perl CGI

The backbone of the web service is a set of Perl CGI scripts that mimics the E-utilities API, and provides valid RDF data for many of the E-utilities requests.

There is one CGI script for each of the E-utilities programs that we implement: [einfo.cgi](https://github.com/Klortho/eutilsrdf/blob/master/einfo.cgi), [elink.cgi](https://github.com/Klortho/eutilsrdf/blob/master/elink.cgi), and [esummary.cgi](https://github.com/Klortho/eutilsrdf/blob/master/esummary.cgi). ([efetch.cgi](https://github.com/Klortho/eutilsrdf/blob/master/efetch.cgi) also exists, but none of the responses from this has been implemented yet). These all have exactly the same content, and are just thin wrappers around the module [EutilsRDF.pm](https://github.com/Klortho/eutilsrdf/blob/master/lib/EutilsRDF.pm).

That module has a run() method, which is executed for every HTTP request. It examines the URL, the query-string parameters, and the environment, and dispatches the request. Normally, if there are no errors, then NCBI E-utilities is invoked to request XML output. That XML is then passed through an XSLT stylesheet with the same name as the utility.

# XSLT transformations

There is one XSLT stylesheet for each of the implemented E-utilities:

* + [einfo.xsl](https://github.com/Klortho/eutilsrdf/blob/master/einfo.xsl)
  + [elink.xsl](https://github.com/Klortho/eutilsrdf/blob/master/elink.xsl)
  + [esummary.xsl](https://github.com/Klortho/eutilsrdf/blob/master/esummary.xsl)

Transformations were written in XSLT 1.0 (since I used Perl's XML::LibXSLT module to run them, which only supports 1.0).

The input to each of these is the XML output from the respective NCBI E-utility tool. The output from each of these is either a simple error response, or a valid RDF/XML document.

Doing this work on GitHub, and providing these XSLTs as examples/templates, facilitates others who might want to contribute other transformations of E-utilities responses that were not covered by this initial effort. Others could also, of course, enhance or enrich the outputs already provided. GitHub provides a permanent public repository, that makes it very easy for other groups or individuals to contribute.

# Service responses

The EutilsRDF service produces various types of responses, depending on the request.

The [home page of the service](http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/) lists several sample invocations, including a few error cases. (Note that the table of samples is generated from a data file, [tests.xml](https://github.com/Klortho/EutilsRDF/blob/master/tests.xml) that could also be used for automated testing.

This utility will *always* return a well-formed XML document. If no error is encountered, then it will return a valid RDF/XML document.

The following is a detailed list of all of the possible responses.

* 200 OK, with a valid RDF/XML document.
* 404 Not found - for cases where it doesn't understand the request, or if the request is for RDF (retmode is rdf) but (for example) the requested service doesn't have a transformation implemented.
* 303 See Other redirect to the NCBI Eutilities URL - if the request is not for RDF (retmode is not given or is not rdf)
* 502 Bad gateway - if this service received a bad response from the NCBI E-utilities.

## Validating RDF responses

All of the RDF responses, for the samples listed on the home page, have been validated both with the [W3C RDF Validation service](http://www.w3.org/RDF/Validator/) (W3C 2006 [6]) and with the [Virtuoso Universal Server](http://virtuoso.openlinksw.com/) (OpenLink Software [7]), in order to ensure that the provided output is valid and useful.

## EInfo

The sample response types that have been converted into RDF include the responses from the [EInfo](http://www.ncbi.nlm.nih.gov/books/NBK25499/#chapter4.EInfo) (NCBI 2010 [1]) tool.

There are two types of responses that this tool provides:

1. List of all databases.
2. Database details.

### EInfo - list of all databases

When invoked without any query-string parameters, it provides a simple listing of all of the Entrez databases. To see the live NCBI EInfo response, click [here](http://eutils.ncbi.nlm.nih.gov/entrez/eutils/einfo.fcgi?retmode=xml). If the EutilsRDF service is up and running, then the converted RDF response should be available from [here](http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/einfo.cgi?retmode=rdf).

The following is a sample of the original E-utilities XML.

<eInfoResult>  
 <DbList>  
 <DbName>pubmed</DbName>  
 <DbName>protein</DbName>  
 <DbName>nuccore</DbName>  
 ...  
 </DbList>  
</eInfoResult>

Transformed into RDF/XML, this looks like the following.

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
 xmlns:entrez="http://rdf.ncbi.nlm.nih.gov/entrez/" >  
 <entrez:Db rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/db/pubmed">  
 <entrez:dbName>pubmed</entrez:dbName>  
 </entrez:Db>  
 <entrez:Db rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/db/protein">  
 <entrez:dbName>protein</entrez:dbName>  
 </entrez:Db>  
 <entrez:Db rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/db/nuccore">  
 <entrez:dbName>nuccore</entrez:dbName>  
 </entrez:Db>  
 ...  
</rdf:RDF>

The [ontology](https://github.com/Klortho/eutilsrdf/blob/master/entrez-ontology.xml) underpinning this particular graph is very simple. Each Entrez database is assigned a URI, and is an individual of the class entrez:Db. Furthermore, each database has an entrez:dbName property, which is defined to be an owl:DatatypeProperty that is a subclass of rdfs:label. See the "Custom ontology" section below for more details.

### EInfo - database details

When invoked with the db parameter having a value of one of the databases, EInfo provides detailed information and current statistics about that database. For example, for the PMC database, details can be retrieved from the NCBI service [here](http://eutils.ncbi.nlm.nih.gov/entrez/eutils/einfo.fcgi?retmode=xml&db=pmc). From the EutilsRDF servce, the converted response can be seen [here](http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/einfo.cgi?retmode=rdf&db=pmc),.

Here is a snippet of the original XML from E-utilities:

<eInfoResult>  
 <DbInfo>  
 <DbName>pmc</DbName>  
 <MenuName>PMC</MenuName>  
 <Description>PubMed Central</Description>  
 <DbBuild>Build130816-1514.1</DbBuild>  
 <Count>2895721</Count>  
 <LastUpdate>2013/08/16 22:12</LastUpdate>  
 <FieldList>  
 <Field>  
 <Name>ALL</Name>  
 <FullName>All Fields</FullName>  
 <Description>All terms from all searchable fields</Description>  
 <TermCount>216665370</TermCount>  
 <IsDate>N</IsDate>  
 <IsNumerical>N</IsNumerical>  
 <SingleToken>N</SingleToken>  
 <Hierarchy>N</Hierarchy>  
 <IsHidden>N</IsHidden>  
 </Field>  
 ...  
 </FieldList>  
 <LinkList>  
 <Link>  
 <Name>pmc\_bioproject</Name>  
 <Menu>BioProject Links</Menu>  
 <Description>BioProject links</Description>  
 <DbTo>bioproject</DbTo>  
 </Link>  
 ...  
 </LinkList>  
 </DbInfo>  
</eInfoResult>

This provides some information about an Entrez database. Every database has some basic metadata, that includes its name, and a count of records that it currently holds. Also, every database has a set of fields that are indexed and can be used to search it. The values of each of these fields can be one of a few different types, including date, numerical, single-token words, etc. Finally, each database also has a set of associated links, that can be used to find related records in other NCBI databases.

The RDF that results from converting the above result is:

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
 xmlns:entrez="http://rdf.ncbi.nlm.nih.gov/entrez/" >  
 <entrez:Db rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/db/pmc">  
 <entrez:dbName>pmc</entrez:dbName>  
 <entrez:menuName>PMC</entrez:menuName>  
 <entrez:description>PubMed Central</entrez:description>  
 <entrez:dbBuild>Build130816-1514.1</entrez:dbBuild>  
 <entrez:count>2895721</entrez:count>  
 <entrez:lastUpdated>2013/08/16 22:12</entrez:lastUpdated>  
 <entrez:hasField>  
 <entrez:Field rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/db/pmc/fields/ALL">  
 <entrez:fullName>All Fields</entrez:fullName>  
 <entrez:description>All terms from all searchable fields</entrez:description>  
 <entrez:termCount>216665370</entrez:termCount>  
 <entrez:isDate>N</entrez:isDate>  
 <entrez:isNumerical>N</entrez:isNumerical>  
 <entrez:singleToken>N</entrez:singleToken>  
 <entrez:isHierarchy>N</entrez:isHierarchy>  
 <entrez:isHidden>N</entrez:isHidden>  
 </entrez:Field>  
 </entrez:hasField>  
 ...  
 <entrez:hasLink>  
 <entrez:Link rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/link/pmc\_bioproject">  
 <entrez:description>BioProject links</entrez:description>  
 </entrez:Link>  
 </entrez:hasLink>  
 ...  
 </entrez:Db>  
</rdf:RDF>

Each of the RDF resources used here is defined in the [ontology file](https://github.com/Klortho/eutilsrdf/blob/master/entrez-ontology.xml). entrez:Db is a class, and there are several owl:DatatypeProperties defined that give the metadata about the database.

The property entrez:hasField is defined to have a range entrez:Field, which is a class. So every field defined for any of the entrez databases has its own persistent URI identifier. The properties of the field can be used to determine the type of values and other information.

Finally, each database also has any number of links associated with it, and the entrez:hasLink property is used to define those. The class entrez:Link is defined as a subclass of rdf:Property, which will make sense when we look at the output of the ELink utility.

## ELink

Another E-utilities response type that has a conversion into RDF is the [“by id” variant of the ELink utility](http://www.ncbi.nlm.nih.gov/books/NBK25499/#chapter4.ELink) (NCBI 2010 [1]).

Note that the ["batch mode" response](http://www.ncbi.nlm.nih.gov/books/NBK25500/#_chapter1_Finding_Related_Data_Through_En_) will not be handled, because it does not lend itself to conversion to RDF. This is because the response indicates a list of "subjects" and "objects" (in the RDF sense), and the type of link between them, but there is no indication of which object is associated with which of the subjects.

This illustrates an important, general limitation of this project: that not every E-utility request/response is amenable to conversion into RDF.

For an example of a response that can be converted, we construct a query that finds the genes associated with two particular nucleotide sequences. To specify the two nucleotide sources, we use "dbfrom=nuccore" and two id parameter values, 312836839 and 24475906. To specify that we are interested in genes associated with these, we use "db=gene" to identify the target database. The original Eutils XML can be retrieved from [here](http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?retmode=xml&dbfrom=nuccore&db=gene&id=312836839&id=24475906). The converted RDF, from [here](http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/elink.cgi?retmode=rdf&dbfrom=nuccore&db=gene&id=312836839&id=24475906).

The original XML is:

<eLinkResult>  
 <LinkSet>  
 <DbFrom>nuccore</DbFrom>  
 <IdList>  
 <Id>312836839</Id>  
 </IdList>  
 <LinkSetDb>  
 <DbTo>gene</DbTo>  
 <LinkName>nuccore\_gene</LinkName>  
 <Link>  
 <Id>159</Id>  
 </Link>  
 </LinkSetDb>  
 </LinkSet>  
 <LinkSet>  
 <DbFrom>nuccore</DbFrom>  
 <IdList>  
 <Id>24475906</Id>  
 </IdList>  
 <LinkSetDb>  
 <DbTo>gene</DbTo>  
 <LinkName>nuccore\_gene</LinkName>  
 <Link>  
 <Id>22018</Id>  
 </Link>  
 </LinkSetDb>  
 </LinkSet>  
</eLinkResult>

Translated into RDF, this becomes:

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
 xmlns:entrez="http://rdf.ncbi.nlm.nih.gov/entrez/" >  
 <entrez:DbRecord rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/312836839">  
 <nuccore\_gene xmlns="http://rdf.ncbi.nlm.nih.gov/entrez/link/"  
 rdf:resource="http://rdf.ncbi.nlm.nih.gov/entrez/gene/159"/>  
 </entrez:DbRecord>  
 <entrez:DbRecord rdf:about="http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/24475906">  
 <nuccore\_gene xmlns="http://rdf.ncbi.nlm.nih.gov/entrez/link/"  
 rdf:resource="http://rdf.ncbi.nlm.nih.gov/entrez/gene/22018"/>  
 </entrez:DbRecord>  
</rdf:RDF>

This encodes a very simple graph. Each nuccore DbRecord is linked to its corresponding gene DbRecord by a predicate entrez:nuccore\_gene. That predicate is defined in the EInfo response that gives details about the nuccore database (see [here](http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/einfo.cgi?retmode=rdf&db=nuccore)). See the description of the custom ontology below for some discussion of the inferences afforded for this response.

## ESummary

The final sample presented in this paper is of a PMC ESummary result. ESummary is a tool that provides summary metadata for a set of Entrez database records (sometimes called "docsums"). The type of data fields provided depends on the database. See the [ESummary documentation](http://www.ncbi.nlm.nih.gov/books/NBK25499/#chapter4.ESummary) (NCBI 2010 [1]) for more information about this service.

PubMed Central (PMC) is NCBI's archive of full text biomedical literature. An ESummary result for a PMC record provides a collection of bibliographic and other metadata about the article, including publication date, publisher, title, and identifiers that allow cross-referencing to other systems (for example, DOIs).

ESummary outputs are provided in two versions: 1.0 and 2.0. Since version 1.0 is deprecated, the EutilsRDF service only transforms version 2.0. Also, the ESummary result for every database is different, and so each would require custom mapping into RDF. For this project, I have only written the transformation for the *pmc* database. If the request is for an ESummary result from any database that isn't yet implemented, an error would result.

For example, for the article GenBank (2012), the original ESummary XML can be retrieved from [here](http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esummary.fcgi?db=pmc&id=3531190&version=2.0). The converted RDF, from [here](http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/esummary.cgi?retmode=rdf&db=pmc&id=3531190&version=2.0).

Unlike previous transformations, with the PMC docsum, we made use primarily of third-party, well-established, standard ontologies. Much of the design behind this transformation was derived from the work described in [Biotea: RDFizing PubMed Central in Support for the Paper as an Interface to the Web of Data](http://www.jbiomedsem.com/content/4/S1/S5) (Castro et. al. 2012 [8]). The transformation uses [BIBO](http://bibliontology.com/specification) (D'Arcus and Giasson 2009 [9]) and [DCMI Terms](http://dublincore.org/documents/dcmi-terms/) (DCMI Usage Board 2012 [10]) to model bibliographic metadata, and [FOAF](http://www.foaf-project.org/) ([11]) to identify authors.

Given the following original XML from E-utilities,

<eSummaryResult>  
 <DocumentSummarySet status="OK">  
 <DbBuild>Build130816-1514.1</DbBuild>  
 <DocumentSummary uid="3531190">  
 <PubDate>2013 Jan</PubDate>  
 ...  
 <Source>Nucleic Acids Res</Source>  
 <Authors>  
 <Author>  
 <Name>Benson DA</Name>  
 <AuthType>Author</AuthType>  
 </Author>  
 <Author>  
 <Name>Cavanaugh M</Name>  
 <AuthType>Author</AuthType>  
 </Author>  
 ...  
 </Authors>  
 <Title>GenBank</Title>  
 <Volume>41</Volume>  
 <Issue>D1</Issue>  
 <Pages>D36-D42</Pages>  
 <ArticleIds>  
 <ArticleId>  
 <IdType>pmid</IdType>  
 <Value>23193287</Value>  
 </ArticleId>  
 <ArticleId>  
 <IdType>doi</IdType>  
 <Value>10.1093/nar/gks1195</Value>  
 </ArticleId>  
 <ArticleId>  
 <IdType>pmcid</IdType>  
 <Value>PMC3531190</Value>  
 </ArticleId>  
 </ArticleIds>  
 <FullJournalName>Nucleic Acids Research</FullJournalName>  
 ...  
 </DocumentSummary>  
 </DocumentSummarySet>  
</eSummaryResult>

The following RDF is produced

<?xml version="1.0"?>  
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"   
 xmlns:owl="http://www.w3.org/2002/07/owl#"  
 xmlns:dct="http://purl.org/dc/terms/"  
 xmlns:bibo="http://purl.org/ontology/bibo/"  
 xmlns:foaf="http://xmlns.com/foaf/0.1/" >  
 <rdf:Description rdf:about="http://rdf.ncbi.nlm.nih.gov/pmc/articles/PMC3531190">  
 <rdf:type rdf:resource="http://purl.org/ontology/bibo/Document"/>  
 <rdf:type rdf:resource="http://purl.org/ontology/bibo/AcademicArticle"/>  
 <dct:identifier>  
 http://rdf.ncbi.nlm.nih.gov/pmc/articles/PMC3531190</dct:identifier>  
 <owl:sameAs rdf:resource="http://biotea.idiginfo.org/pubmedOpenAccess/rdf/PMC3531190"/>  
 <rdfs:seeAlso>  
 http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3531190/</rdfs:seeAlso>  
 <dct:date>2013 Jan</dct:date>  
 <bibo:authorList rdf:parseType="Collection">  
 <rdf:Description>  
 <foaf:name>Benson DA</foaf:name>  
 </rdf:Description>  
 <rdf:Description>  
 <foaf:name>Cavanaugh M</foaf:name>  
 </rdf:Description>  
 ...  
 </bibo:authorList>  
 <dct:title>GenBank</dct:title>  
 <bibo:volume>41</bibo:volume>  
 <bibo:issue>D1</bibo:issue>  
 <bibo:pageStart>D36</bibo:pageStart>  
 <bibo:pageEnd>D42</bibo:pageEnd>  
 <bibo:pmid>23193287</bibo:pmid>  
 <dct:identifier>  
 http://rdf.ncbi.nlm.nih.gov/pubmed/23193287</dct:identifier>  
 <rdfs:seeAlso>http://www.ncbi.nlm.nih.gov/pubmed/23193287</rdfs:seeAlso>  
 <bibo:doi>10.1093/nar/gks1195</bibo:doi>  
 <dct:identifier>doi:10.1093/nar/gks1195</dct:identifier>  
 <owl:sameAs rdf:resource="http://dx.doi.org/10.1093/nar/gks1195"/>  
 <dct:isPartOf>  
 <bibo:Journal>  
 <dct:title>Nucleic Acids Research</dct:title>  
 </bibo:Journal>  
 </dct:isPartOf>  
 </rdf:Description>  
</rdf:RDF>

The PMC article in question is assigned a stable HTTP URI according to the RDF URI standards document (see the Appendix). That URI is "http://rdf.ncbi.nlm.nih.gov/pmc/articles/PMC3531190". Next, the type of resource is identified unambiguously by specifying that it belongs to the classes bibo:Document and bibo:AcademicArticle.

The article is linked to internal and external identification schemes using the both the dct:identifier, owl:sameAs, and other predicates. dct:identifier is used to specify text strings that are used as identifiers in the PMC system itself, as well as in PubMed and the DOI system. The article is also cross-referenced, with an owl:sameAs relationship, to both the formal Biotea and the DOI RDF resources. The bibo:pmid predicate is also used to specify the PubMed ID, and the bibo:doi to specify the DOI.

It is also cross-referenced to various web documents, using the rdfs:seeAlso link. These include the HTML version of the PMC article on the NCBI site, as well as the PubMed abstract.

The publication date is indicated with the dct:date predicate.

The author list is given with the RDF collection data structure. This is significant because it preserves the ordering of the authors, whereas multiple hasAuthor triples, for example, would not. (As any academic knows, the order of the authors on a published work is quite significant!)

Other BIBO predicates are used to provide extra bibliographic data, such as the journal, issue, volume, and page numbers. Note that the journal ISSN is not available from the ESummary output, so a blank bibo:Journal node is used, to which the journal's title is attached.

# Ontologies

## Custom ontology

The custom ontology developed for this project has a canonical URI of http://rdf.ncbi.nlm.nih.gov/entrez/, and can be found in the GitHub repository [here](https://github.com/Klortho/EutilsRDF/blob/master/entrez-ontology.xml).

The ontology comprises several terms for classes and properties. Names were assigned following the best practices recommendations of "Semantic Web for the Working Ontologist" (Allemang & Hendler 2011 [12]), viz:

* Camelcase was used for all vocabulary terms,
* Property names start with a lowercase letter, and
* Class names start with an uppercase letter.

In addition to the controlled vocabulary of terms, the "namespace" given by the ontology URI (specifically, "http://rdf.ncbi.nlm.nih.gov/entrez/") also contains some dynamically generated URIs. For example:

* Individual Entrez databases. For example, http://rdf.ncbi.nlm.nih.gov/entrez/db/pmc
* Links. For example, http://rdf.ncbi.nlm.nih.gov/entrez/link/pmc\_gene

The classes defined in the ontology are:

* entrez:Db - an NCBI Entrez database
* entrez:Field - a search field
* entrez:Link - a link, that corresponds to an RDF property between two Entrez database records
* entrez:DbRecord - a record within any of the Entrez databases

Some of the key properties are:

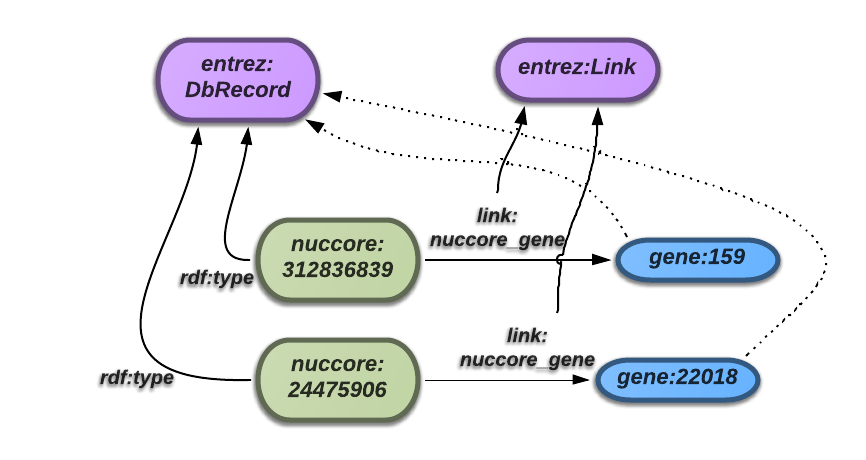
* entrez:dbName - name of a database. Defined to be a subclass of rdfs:label.
* entrez:description - human-readable description, used in many places. Also defined to be a subclass of rdfs:label.
* entrez:hasField - links an entrez:Db to an entrez:Field.
* entrez:hasLink - links an entrez:Db to an entrez:Link.

There are many other datatype properties defined which are used to record information about an Entrez database, as provided by the EInfo responses.

As mentioned above, individual Entrez links (such as, for example, *gene\_nuccore*) map very cleanly to RDF predicates (maps to link:gene\_nuccore). Information about these links is not in the ontology file, since these are dynamically generated from the Entrez system. They can be discovered, however, through the EInfo calls that provide details about an individual database. In the RDF versions of those responses (see above), each link is declared to be a subclass of entrez:Link, which is a subclass of rdf:Property, and also fixes its domain and range to be an entrez:DbRecord.

One of the common relationships among links in Entrez are so-called "reciprocal" links, which just owl:inverseOf relationships between link. Unfortunately, those relationships are not exposed through E-utilities, and so can't be automatically discovered through this service. As a sample, I encoded one such relationship, between link:gene\_nuccore and link:nuccore\_gene, in the ontology file.

A class diagram for a very simple ELink result is given below.



*Figure 2. ELink result graph.*

This graph defines two resources that represent *nuccore* database records, and specifies each to be of type entrez:DbRecord. Then, it gives the links to the *gene* database records as RDF triples. Because the link:nuccore\_gene property is defined to be a subclass of entrez:Link, its domain and range are known, and so the *gene* objects can be inferred (dotted lines) to be of type entrez:DbRecord as well.

This ontology file has been validated both with the [W3C RDF Validation service](http://www.w3.org/RDF/Validator/) and with Protege.

## Third-party ontologies

In this project, third-party ontologies were used in the construction of the RDF representation of the PMC docsums (ESummary results). There were two reasons for doing this:

* These controlled vocabularies already encode the semantic relationships we need, and are already established and in common use -- why reinvent the wheel?
* It allows the data to conform to Tim Berners-Lee's fourth principle for linked open data (Berners-Lee 2009): that it link to other, external URIs, so that users can get more information about the resources of interest.

Note that the ontologies that were selected for use for the PMC docsums might not be appropriate for other Entrez database records. With the modular architecture of the transformations, each database could be transformed using whatever ontologies best fit their data.

In addition to the fundamental standards of RDFS and OWL, the ontologies used were:

* [DCMI Metadata Terms](http://dublincore.org/documents/dcmi-terms/) (DCMI 2012 [10])
* [Bibliographic Ontology Specification](http://bibliontology.com/specification) (BIBO) (D'Arcus & Giasson 2009 [9])
* Biotea (Castro et al. 2010 [8])
* [Friend of a Friend](http://www.foaf-project.org/) (FOAF [11])

# Practical applications

As a simple demonstration of the value of providing this data in RDF format, I used the service to load some data into an installation of the [Virtuoso Universal Server](http://virtuoso.openlinksw.com/) (OpenLink Software [7]), and executed some SPARQL queries on it.

I loaded the following results into a single graph:

* http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/einfo.cgi?retmode=rdf - the list of all Entrez databases
* http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/einfo.cgi?retmode=rdf&db=pmc - detailed information about the PMC database
* http://www.ncbi.nlm.nih.gov/staff/maloneyc/eutilsrdf/elink.cgi?retmode=rdf&dbfrom=pmc&db=nuccore&id=3531190 - links from the article GenBank to nucleotide sequences.

Using that graph, the following SPARQL query finds all of the Entrez links defined for the PMC database:

select ?link  
where {   
 ?pmc a <http://rdf.ncbi.nlm.nih.gov/entrez/Db> .  
 ?pmc <http://rdf.ncbi.nlm.nih.gov/entrez/dbName> "pmc" .  
 ?pmc <http://rdf.ncbi.nlm.nih.gov/entrez/hasLink> ?link  
}

The results give 26 links, including pmc\_bioproject, pmc\_biosystems\_pubmed, pmc\_books\_refs, etc.

The following finds all of the nucleotide sequences associated with the *GenBank* article:

select ?seq  
where {   
 <http://rdf.ncbi.nlm.nih.gov/entrez/pmc/3531190>  
 <http://rdf.ncbi.nlm.nih.gov/entrez/link/pmc\_nuccore> ?seq .  
}

Results:

http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/387756559  
http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/387756539  
http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/224384768  
http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/54362548  
http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/20043254  
http://rdf.ncbi.nlm.nih.gov/entrez/nuccore/7274584

Of course, this approach has the drawback that it would not typically be feasible to load all of the triples of interest, using this interface, into a single triple-store. Greater benefits would be realized if NCBI were to provide a SPARQL endpoint, as part of an enhanced Entrez API, that would allow federated queries.

Another benefit that could be realized with this interface now, however, is that it would be possible to take advantage of RDF libraries that are provided in many programming languages, in order to process the data. One of the barriers to writing scripts against E-utilities is that the data format and structure is difficult to parse.

The ontology provided also allows users to take advantage of inferencing that is afforded by models. A sophisticated inferencing engine could find some relationships that are not explicit in the E-utilities data. This would depend on further enhancement of the ontology file, which, currently, does not contain significant modeling constructs.

# Related work, sites, and services

[Bio2RDF](http://bio2rdf.org/) ([13]) is a project and a website that provides RDF data from many different data sets in the life sciences. The developers on this project have already done a fantastic amount of work to integrate NCBI’s (as well as other institutions’) data into the Semantic Web. Among the data sets that they have converted into RDF are NCBI’s Gene, GenBank, Homologene, OMIM, PubMed, RefSeq, and Taxonomy databases.

The [TogoWS web service](http://togows.org) ([14]) provides a wrapper for some eutils, in a manner very similar to this project.  
For example, [this document](http://togows.org/entry/pubmed/23398680.ttl) is an RDF representation of a PubMed record.

# Conclusions and future work

Limited time, unfortunately, prevented me from researching as fully as I would have liked some of the pre-existing work in the Semantic Web that relates to, or overlaps with, this project. In particular, it would be important to harmonize this work with what is already being done in the Bio2RDF project.  
The [Bio2RDF-scripts Wiki](https://github.com/bio2rdf/bio2rdf-scripts/wiki) (Bio2RDF [15]) provides extensive detailed information about how that project operates.

Likewise, it would be worthwhile to integrate the work done for the [TogoWS web service](http://togows.org) ([14]) to see how it could be integrated with this effort.

Implementing more of the E-utilities responses would be an obvious next step, starting with all of the other ESummary responses, and then tackling EFetch wherever practical. As always, researching work done by others, and providing interlinking that is harmonious with those schemes, would be important.

To enhance the discoverability of this service, it would be worthwhile to find other registries similar to the NCBO BioPortal, with which we could register this service.

Automated tests should be added, as described in the "Responses" section above.

Other "to do" items will be added and tracked in the [GitHub issues](https://github.com/Klortho/EutilsRDF/issues). Please feel free to add your own, or to offer other contributions.

# Acknowledgements

I would like to thank the following people for their valuable suggestions regarding the NCBI RDF URI standards document:

* + Jerven Bolleman (Uniprot),
  + Michel Dumontier (Bio2RDF),
  + Fu Gang (NCBI),
  + Evan Bolton (NCBI),
  + Mark Johnson (NCBI).

I'd also like to thank my professor, Dr. Richard Cost, for important feedback on this paper.

# References

1. [Entrez Programming Utilities Help](http://www.ncbi.nlm.nih.gov/books/NBK25501/). NCBI. 2010.
2. [National Center for Biomedical Ontology BioPortal](http://bioportal.bioontology.org/). NCBO.
3. [Linked Data - Design Issues](http://www.w3.org/DesignIssues/LinkedData.html). Tim Berners-Lee. 2006. ([WebCite](http://www.webcitation.org/6HFgSl1S7))
4. [Cool URIs for the Semantic Web](http://www.w3.org/TR/cooluris/). W3C. 2008.
5. [Linked Data: Evolving the Web into a Global Data Space](http://linkeddatabook.com/editions/1.0/). Tom Heath and Christian Bizer. 2011. Accessed: 2013-06-09. ([WebCite](http://www.webcitation.org/6HFldFSR6))
6. [W3C RDF Validation service](http://www.w3.org/RDF/Validator/). W3C. 2006.
7. [Virtuoso Universal Server](http://virtuoso.openlinksw.com/). OpenLink Software.
8. [Biotea: RDFizing PubMed Central in Support for the Paper as an Interface to the Web of Data](http://www.jbiomedsem.com/content/4/S1/S5), Garcia Castro, L J; McLaughlin, C; Garcia, A. 2012.
9. [Bibliographic Ontology Specification](http://bibliontology.com/specification) (BIBO). D'Arcus B, Giasson F. 2009.
10. [DCMI Metadata Terms](http://dublincore.org/documents/dcmi-terms/). DCMI Usage Board. 2012.
11. [The Friend of a Friend (FOAF) project](http://www.foaf-project.org/).
12. Semantic Web for the Working Ontologist. Dean Allemang and Jim Hendler. 2011.
13. [Bio2RDF](http://bio2rdf.org/)
14. [TogoWS web service](http://togows.org)
15. [Bio2RDF-scripts Wiki](https://github.com/bio2rdf/bio2rdf-scripts/wiki) Bio2RDF.

# Public domain dedication

CC0

To the extent possible under law, Christopher Maloney has waived all copyright and related or neighboring rights to NCBI E-utilities in RDF. This work is published from: United States.

# Appendix: NCBI RDF URI Standard

### Introduction

This section will describe standards for NCBI-minted persistent URIs that identify RDF resources.

The goals of this document include:

* Provide guidance to help projects get started publishing data on the Semantic Web,
* Provide standards and recommendations for the forms that the published RDF URIs should take,
* Provide distributed authority and control, that allows for groups/projects to mint their own identifiers, by providing namespaces,
* Specify a centralized system for handling HTTP requests to RDF URIs, in accordance with current best practices (e.g. HTTP 303 redirects)

RDF URIs are different from the URIs that we provide for our web pages. For background on this topic, please read the W3C document, [*Cool URIs for the Semantic Web*](http://www.w3.org/TR/cooluris/).  In a nutshell, best practices dictate that there be a clear distinction between URIs for real world resources, such as, for example, people, genes, and abstract concepts, and URIs for web documents, including the web documents that describe those real-world resources.  Furthermore, both should be identified by HTTP URIs.  (Keep in mind that even though these begin with "http://", they are first-and-foremost identifiers, rather than web addresses or locations.)

This, however, immediately leads to the question of what a server should do when an HTTP client requests the URI of a real world resource, as opposed to the URI of a web document.  There are two common approaches, both of which we should support (see [below](#NCBIRDFURIStandards-NCBIRDFURIStandards)).

* Use hash URIs for real world resources, such as, for example, **http://www.w3.org/2004/02/skos/core#Concept**.  In this case, when a client dereferences this, it sends the URI without the hash part to the server, i.e.  <http://www.w3.org/2004/02/skos/core>.  This URI, without the hash part, is then defined as the URI of a web document which includes information about the #Concept resource, and it is returned to the client.
* Use "303 See other" redirects.  In this case, when a web client dereferences an HTTP URI for a real world resource, the server responds with an HTTP status code, "303 See other", and redirects the client to the URI for the web document which is a representation of that real-world resource.  For example, the identifer **http://xmlns.com/foaf/0.1/Person** represents the abstract notion of a person.  When dereferenced, the server responds with a "303 See other", redirecting the client to a web document at <http://xmlns.com/foaf/spec/>, which includes information about that term.

Unlike URLs for web pages, which often can appear in various equivalent forms, it is very important for RDF URIs to be:

* Unambiguous. The thing that the URI refers to needs to be clearly defined
* Stable. Once defined, they should not change.  For the Semantic Web, stable URIs are even more important than they are for web pages (where the main concern would be breaking users' bookmarks, or links that might exist in other web pages).  In the world of linked open data, there might be many external applications that rely on URIs of a specific form, and changing them could break those applications in subtle ways.
* Consistent. The resources these URIs refer to are identified by character-by-character comparisons of these URIs, so minor inconsistencies such as those in case, trailing slashes, etc. should be avoided.
* Unique. It's recommended that, whenever possible, only one RDF URI be assigned for each NCBI resource. There can be exceptions to this rule, in which case the [owl:sameAs](http://owlsameAs) property can be used to identify two URIs as pointing to the same thing; but, in general, this should be avoided.

### General requirements

All NCBI URIs that are intended for use within RDF will be HTTP URIs with the base **http://rdf.ncbi.nlm.nih.gov/**.

Rationale

The domain prefix "rdf" was chosen over alternatives, because it clearly delimits the scope and intended use for these URIs. It is preferred to use a domain prefix, rather than a path segment under "www", because this facilitates centralized control over how these URIs are dereferenced, keeps the URIs shorter, and also allows them to correspond with "www" URIs more closely.

The URIs will be partitioned into "namespaces" by virtue of top-level path segments. Each group or project will have control over the URIs in their namespace, but should consult with the standards group to make sure that their URIs are in accord with the guidelines, as well as to make sure that the RDF URI resolver is up-to-date.

Example

The namespace "pubchem" is assigned to the PubChem group. All URIs with the base part **http://rdf.ncbi.nlm.nih.gov/pubchem/** are under their control. They have further divided this into subnamespaces, such as, for example,

* **http://rdf.ncbi.nlm.nih.gov/pubchem/compound/**
* **http://rdf.ncbi.nlm.nih.gov/pubchem/conformer/**
* etc.

For resources related to projects or databases that are in Entrez, it is recommended that the namespace name be the same as the Entrez database name, in all-lowercase, as returned by [einfo](http://eutils.ncbi.nlm.nih.gov/entrez/eutils/einfo.fcgi).

Example

All PMC URIs will start with **http://rdf.ncbi.nlm.nih.gov/pmc/**

Exception

PubChem has a few Entrez databases (including pccompound and pcsubstance) but their URIs are in the "pubchem" namespace. For example,  **http://rdf.ncbi.nlm.nih.gov/pubchem/compound/31703** .

Unlike the convention for URLs for our web pages, no trailing slashes should be used for RDF URIs

Rationale

With our web pages, the trailing-slash convention has not caught on: some groups implement it while others do not. Unlike web pages, for RDF URIs, exact string matches are important. The rest of the world seems to eschew trailing slashes, so there's no reason for us to impose them on ourselves for these new URIs. Furthermore, the reason for using trailing slashes -- that it makes it easier to refer to related resources using relative URIs -- does not apply to these RDF URIs.

Example

The URI for the PMC article PMC14900 is  **http://rdf.ncbi.nlm.nih.gov/pmc/articles/PMC14900** , with no trailing slash

Path segments in URIs should be NCNames, restricted to US-ASCII, with all letters in lowercase. In other words, path segments should start with a lower-case letter or an underscore ([a-z] | \_), and subsequent characters should be among ([a-z] | \_ | - | . | [0-9]).

Rationale

Establishing these conventions makes it less likely that typographical errors will arise.

RDF URIs should not use query strings.

RDF URIs for real-world resources should not use "filename extensions".

Rationale

Filename extensions will be used by the URI resolving system to mimic the content-negotiation features of HTTP, but they are technically not part of the RDF URI that identifies the real-world resources.

Bad example

The URI **http://rdf.ncbi.nlm.nih.gov/pmc/people/fred.html** is not good for the identification of a real-world resource. This would be interpreted by the URI resolving system as the URI **http://rdf.ncbi.nlm.nih.gov/pmc/people/fred** (which is a good RDF URI) with the ".html" part being used to mimic the HTTP accept header "text/html". In other words, dereferencing this URI would be equivalent to requesting the HTML representation of the resource identified by **http://rdf.ncbi.nlm.nih.gov/pmc/people/fred**.

### Slash-namespace and hash-namespace URIs

There are two types of URIs that will be used for RDF resources (see [Best Practice Recipes for Publishing RDF Vocabularies](http://www.w3.org/TR/swbp-vocab-pub/)):

* Slash-namespace URIs are formed by appending a local part to a URI that ends in a slash. For example, **http://xmlns.com/foaf/0.1/Person**.
* Hash-namespace URIs are formed by appending a local part to a URI that ends in a hash character. For example, **http://www.w3.org/2004/02/skos/core#Concept**.

These correspond with the two methods for distinguishing between real world URIs and web document URIs, described above.

At NCBI, it is strongly encouraged that slash-namespace URIs be used for all RDF URIs, including vocabulary and ontology terms.  However, that is at the discretion of the project members, and hash-namespace URIs could also be supported.

Rationale

Typically, these two types of URIs are used for different things.   Slash-namespace URIs are used to identify resources within large data sets, such as items in a database. Hash-namespace URIs are used to identify vocabulary terms, such as those that are manually designed, including classes and properties. The reason is, as described in [Making URIs dereferenceable](http://linkeddatabook.com/editions/1.0/#htoc11), when deferencing hash URIs, a web document can be retrieved that includes information about each of the resources associated with the set of possible hash values. In the case of vocabulary terms, this is typically an RDF Schema or OWL ontology file.

The reason to avoid hash-namespace URIs is that, every time any one of them gets dereferenced, the server must respond with the schema or ontology file that defines all of the terms within that namespace (since the server doesn't know which specific term is being requested). Invariably, these namespaces tend to grow, and it would be better not to have to worry about server performance or response times, when considering whether or not to add new terms to a vocabulary.

In practice, the only difference between these two types of URIs is that, when dereferencing hash-namespace URIs, the server (URI resolver) gets less information. When properly configured, there is nothing that the URI resolver couldn't do with slash-namespace URIs that it could do with hash-namespace URIs (such as returning on ontology file).

Example

An Entrez link could be identified with the URI **http://rdf.ncbi.nlm.nih.gov/entrez/vocabulary/nuccore\_gene**. When dereferenced, the URI resolver could be configured to do a 303 redirect to an OWL document at, for example, **http://rdf.ncbi.nlm.nih.gov/entrez/vocabulary.owl**.

However, while not recommended, an alternative would be to define the Entrez link with the URI  **http://rdf.ncbi.nlm.nih.gov/entrez/vocabulary#nuccore\_gene** . When dereferenced, the base part of this,  **http://rdf.ncbi.nlm.nih.gov/entrez/vocabulary**, could refer to an OWL document that would identify each of the Entrez links, specify that they are instances of rdf:Property, and their various attributes (such as, for example, symmetric and transitive properties).

### Entrez database URIs

For Entrez databases, the default way to form the URI for the real world object corresponding to the database entry is with the database name and the numeric UID, as **http://rdf.ncbi.nlm.nih.gov/{db}/{uid}**.

Example

The URI for the PubMed abstract with PMID 23804282 could be **http://rdf.ncbi.nlm.nih.gov/pubmed/23804282**

However, individual projects can override this default. In particular, it is encouraged that URIs for the real-world resources should, if possible, correspond to the clean "www" URL for the web page describing that resource.

Example

The URI for the journal described at the web page <http://www.ncbi.nlm.nih.gov/pmc/journals/559/> is  **http://rdf.ncbi.nlm.nih.gov/pmc/journals/559**.

Example

The URI for version 2 of the PMC article PMC1868567 is **http://rdf.ncbi.nlm.nih.gov/pmc/articles/PMC1868567.2**