

Resource IRI resolution service

Introduction

Vocabularies represented in RDF define *resources*, each of which is assigned an IRI ([Internationalized Resource Identifier](#)). Please refer to these related pages for more information on representing resources in RDF:

- [Good practice in vocabulary creation](#)
- [Working with resource identifiers](#)

As noted in the various "best practice" guides referenced at those pages, resources made available as linked data use IRIs which are valid URLs.

As a provider of vocabulary data that is visible in RVA, you may wish to resolve the resource URLs to the appropriate Linked Data API (SISSVoc) page hosted on RVA.

To support this, RVA hosts an IRI resolution service, that maps IRIs for certain types of SKOS resource to the URL for the SISSVoc /resource endpoint of the containing vocabulary. (See [Linked Data API](#) for more details about the Linked Data API.)

What the IRI resolution service makes available

The IRI resolution service is designed to resolve resources that meet both of these criteria:

1. resources only in your hosted vocabularies
 - "hosted" means vocabularies published through RVA according to the instructions in the section "Setting up your vocabulary data" below
 - "your" means only resources for which the resource IRI's hostname is registered for your organisation in RVA per the section "Preparing for publication" below
2. resources *either*:
 - explicitly defined to be of one of a small number of specific types: SKOS ConceptSchemes, Collections, and Concepts
 - marked as deprecated using the `owl:deprecated` property, which are not also explicitly defined to be of a type *other* than SKOS ConceptScheme, Collection, or Concept

For example, suppose your organisation is "Frobnitz Research Organisation", and all of your concept IRIs begin with "http://vocab.frobnitz.org/", which is a hostname that you "own".

Specifically, given this vocabulary fragment represented in RDF (using Turtle syntax):

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
<http://vocab.frobnitz.org/def/vocab1/1> a skos:ConceptScheme .
<http://vocab.frobnitz.org/def/vocab1/2> a skos:Collection .
<http://vocab.frobnitz.org/def/vocab1/3> a skos:Concept .
<http://vocab.frobnitz.org/def/vocab1/4> owl:deprecated true .
<http://vocab.frobnitz.org/def/vocab1/5> a skos:Concept ;
    owl:deprecated true .
<http://vocab.frobnitz.org/def/vocab1/6> a owl:Class ;
    owl:deprecated true .
<http://some.other.org/concept/1> a skos:Concept .
```

The resolution service will resolve the resources `http://vocab.frobnitz.org/def/vocab1/1`, `http://vocab.frobnitz.org/def/vocab1/2`, `http://vocab.frobnitz.org/def/vocab1/3`, `http://vocab.frobnitz.org/def/vocab1/4`, and `http://vocab.frobnitz.org/def/vocab1/5`, but not `http://vocab.frobnitz.org/def/vocab1/6` or `http://some.other.org/concept/1`.

Note: as per the second criterion mentioned at the beginning of this section, the determination of whether a resource is one of the specific types of interest is *only* on the basis of the presence of triples specifying resource type and deprecated status. However, a triple of the form:

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>

<http://vocab.frobnitz.org/def/vocab1/1> skos:exactMatch <http://vocab.frobnitz.org/def/vocab2/2> .
```

would *not*, by itself, enter either the subject or object resource of that triple into the resolution service, *even though* by the rules of entailment, this triple is enough to *infer* that both resources are SKOS Concepts. (See Example 57 of the SKOS Reference at <http://www.w3.org/TR/skos-reference/#example-57>.) This enables you to partition your concepts into separate vocabularies, while ensuring correct resolution of resources.

The IRI resolution service endpoint

The IRI resolution service has an endpoint hosted on RVA. The endpoint is:

```
https://vocabs.ardc.edu.au/registry/api/services/resolve/lookupIRI
```

The following query parameters are available:

Query parameter	Required/optional	Meaning	Example
iri	Required	The IRI to be resolved.	<code>http://vocab.frobnitz.org/def/vocab1/1</code>
mode	Optional; defaults to "current"	The resolution mode. At present, only "current" is supported, and is the default value.	<code>current</code>
suffix	Optional	A suffix to be appended to the resolved URL. See the documentation of the Elda library for more details. (For example, see the Elda cribsheet .) Typically, this will be URL encoded, and begin with %26, which is the URL encoding of the & character.	<code>%26_format%3Djson</code>

Here are some example lookups:

- `https://vocabs.ardc.edu.au/registry/api/services/resolve/lookupIRI?iri=http://vocab.frobnitz.org/def/vocab1/1`
- The simplest form of resolution. Resolve an IRI, using the "current" mode. No suffix is appended.
`https://vocabs.ardc.edu.au/registry/api/services/resolve/lookupIRI?iri=http://vocab.frobnitz.org/def/vocab1/1&mode=current`
- The same as the previous example.
`https://vocabs.ardc.edu.au/registry/api/services/resolve/lookupIRI?iri=http://vocab.frobnitz.org/def/vocab1/1&suffix=%26_format%3Djson`

The suffix `&_format=json` will be appended to the URL returned by the resolver.

Resolution modes

Mode: current

This is at present the only supported mode. If you require support for different behaviour, please contact us at services@ardc.edu.au to discuss your needs.

In `current` mode, the behaviour of the resolver is as follows:

Resource IRI	Resolution behaviour
Resource not defined in any version of any vocabulary	Not resolved; a resolution attempt yields a "404" page

Resource defined only in <i>superseded</i> version(s) (i.e., not in any <i>current</i> version of any vocabulary)	Not resolved; a resolution attempt yields a "404" page
Resource defined in the <i>current</i> version of multiple vocabularies	Not resolved; a resolution attempt yields a "404" page
Resource defined in the <i>current</i> version of one vocabulary	Redirected (HTTP Status code 307) to the Elda (SISSVoc) resource page for the resource

Please note that resources will continue to resolve if the *vocabulary* itself is subsequently deprecated. In order to stop resources from being resolved, it is necessary either to mark the current *version* as superseded, or to delete it completely.

How to use the IRI resolution service

Preparing for publication

Because RVA's IRI resolution service supports many vocabulary publishers, it is necessary for RVA to keep track of which hosts correspond to each vocabulary owner.

For example, suppose your organisation is "Frobnitz Research Organisation", and all of your resource IRIs begin with "http://vocab.frobnitz.org/". A configuration entry must be added to RVA that allows members of your organisation to add your resources beginning with that host to the resolution service. Note: your organisation may control multiple hostnames, and you may use more than one of them in your resource IRIs; RVA will need to know about all of those hostnames.

At present, there is no user interface to manage this mapping. Therefore, please get in contact with us to set up the mapping between your organisation and the hosts you control. To contact us, please send an email to services@ardc.edu.au.

Setting up your vocabulary data

1. Create your vocabulary record in RVA.
2. Add a version to your vocabulary, with its status as "current", and import RDF data into it, either from a PoolParty project, or by uploading SKOS RDF data. Ensure that both of the two checkboxes "Import vocabulary into the ARDC Vocabulary Repository" and "Publish vocabulary via the ARDC Linked Data API" are checked.
3. Publish the vocabulary.
4. As part of the publication process, the vocabulary data will be loaded into a Sesame repository. That repository will then be queried to find all SKOS ConceptScheme, Collection, and Concept resources in the repository whose IRIs match against the hostnames recorded against your organisation. (See the previous section for details about how to prepare for this.)

Setting up rewriting in your web server

There are numerous types of web server, and numerous ways of arranging URL rewriting, so it is not possible to provide instructions that will suit all possible situations. The following section provides instructions for one specific situation that may be adaptable to your setup. If you require further assistance, please contact services@ardc.edu.au.

Adding URL rewriting to Apache HTTP Server

The following instructions are for Apache HTTP Server. If you're using a different web server, you'll need to make adjustments.

Please see the Apache HTTP Server documentation for URL rewriting. For version 2.4, that page is http://httpd.apache.org/docs/2.4/mod/mod_rewrite.html. For version 2.2, that page is http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html.

You will first need to ensure that the `mod_rewrite` module is available and enabled. That might mean confirming that the `httpd.conf` file contains a line such as this:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

If `httpd.conf` contains a line like this, but it is commented out (i.e., the line begins with a hash, "#"), then you will need to uncomment the line (by removing the hash character).

Most modern packaged versions of Apache HTTP Server support splitting up configurations into multiple files, which are included by the main configuration file using the configuration file `Include` directive. For example, on many Linux distributions, the file `httpd.conf` contains this line:

```
Include conf.d/*.conf
```

so that all files in the directory `conf.d` with a filename that ends in `.conf` will be included. If your setup is like this, it is easiest to create a new configuration file, say, `vocab_rewriting.conf`, just to contain the rewriting. Otherwise, you will need to edit the main configuration file `httpd.conf` to add the rewriting.

In the new configuration file (or in `httpd.conf`, if that's what you're using), insert the following:

```
RewriteEngine On
RewriteRule "^(/def/.*)" "https://vocabs.ardc.edu.au/registry/api/services
/resolve/lookupIRI?iri=http://vocab.frobnitz.org$1" [B,L,NE,QSA,R=307]
```

Please note the following points:

- The pattern for rewriting will match only those URLs that begin with `http://vocab.frobnitz.org/def/`. If your web server serves other content, you will need to have devised a URL pattern for your resources that enables them to be identified in such a way.
- The parentheses in the pattern enclose the entire path; this is then substituted where `"$1"` occurs in the substitution.
- In this example, only URLs with the HTTP scheme are considered. If your resource IRIs use the HTTPS scheme, modify the part of the substitution after `"?iri="` accordingly. The substitution must use the scheme that is used in your resource IRIs. For example, if your resource IRIs begin with `http://...`, your web server needs to support rewriting of accesses on port 80. But you *may* also choose to support rewriting of incoming requests on port 443 (i.e., incoming requests that begin `https` instead.)
- The rewrite flags may be modified to suit your needs. For, example, you may wish to use a status code other than 307. (But note that the resolver will itself send back a redirect with status code 307.) You almost certainly want the `L` flag for efficiency, and the `NE` and `QSA` flags to ensure correct rewriting.

If you use virtual hosts in your configuration, you can *either* include the rewriting given above directly in *each* virtual host for which you wish rewriting to work, or, you can include it at the top level, and use rewrite option inheritance. In this case, within each virtual host for which you want rewriting to work, insert these lines:

```
RewriteEngine On
RewriteOptions Inherit
```