



Gleaning Resource Descriptions from Dialects of Languages (GRDDL)

W3C Recommendation 11 September 2007

This Version:

<http://www.w3.org/TR/2007/REC-grddl-20070911/>

Latest Version:

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

Previous Version:

<http://www.w3.org/TR/2007/PR-grddl-20070716/>

Editor:

[Dan Connolly](#)

Authors:

see [Acknowledgments](#)

Please refer to the [errata](#) for this document, which may include some normative corrections.

See also [translations](#).

Copyright © 2006-2007 W3C® ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

GRDDL is a mechanism for **G**leaning **R**esource **D**escriptions from **D**ialects of **L**anguages. This GRDDL specification introduces markup based on existing standards for declaring that an XML document includes data compatible with the Resource Description Framework (RDF) and for linking to algorithms (typically represented in XSLT), for extracting this data from the document.

The markup includes a namespace-qualified attribute for use in general-purpose XML documents and a profile-qualified link relationship for use in valid XHTML documents. The GRDDL mechanism also allows an XML namespace document (or XHTML profile document) to declare that every document associated with that namespace (or profile) includes gleanable data and for linking to an algorithm for gleanable data.

A corresponding [GRDDL Use Case Working Draft](#) provides motivating examples. A [GRDDL Primer](#) demonstrates the mechanism on XHTML documents which include widely-deployed dialects known as microformats. A [GRDDL Test Cases](#) document illustrates specific issues in this design and provides materials to aid in test-driven development of GRDDL-aware agents.

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](#) at <http://www.w3.org/TR/>.

This is a [W3C Recommendation](#).

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

Comments on this document should be sent to public-grddl-comments@w3.org, a mailing list with a [public archive](#).

This document was produced by [GRDDL Working Group](#), which is part of the [W3C Semantic Web Activity](#). The first release of this document as a Working Draft was 24 Oct 2006 and the Working Group has addressed a number of [comments received](#) and [issues](#) since then. Normative assertions are marked up in this way.

The Working Group's [implementation report](#) demonstrates that the goals for interoperable implementations, set in the [May 2007 Candidate Recommendation draft of this document](#), were achieved.

GRDDL is intended to contribute to addressing Web Architecture issues such as [RDFinXHTML-35](#), [namespaceDocument-8](#), and [xmlFunctions-34](#) as well as issues postponed by the RDF Core working group such as [rdfms-validating-embedded-rdf](#) and [faq-html-compliance](#). In particular, the GRDDL Working Group has postponed [issue-faithful-infoset](#), and anticipates that the resolution of TAG issue [xmlFunctions-34](#) will provide further clarification and guidance.

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). W3C maintains a [public list of any patent disclosures](#) 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\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

The issues appendix that used to be part of this draft has been moved to a [Working Group issues list](#); specifically: [issue-whichlangs](#), [issue-output-formats](#), [issue-base-param](#), [issue-tx-element](#), [issue-html-nsdoc](#), [issue-faithful-infoset](#), [issue-mt-ns](#), [issue-conformance-labels](#), [issue-http-header-links](#)

Table of Contents

1. [Introduction](#)
2. [Adding GRDDL to well-formed XML](#)
3. [GRDDL for XML Namespaces](#)
4. [Using GRDDL with valid XHTML](#)
5. [GRDDL for HTML Profiles](#)
6. [GRDDL Transformations](#)
7. [GRDDL-Aware Agents](#)
8. [Security Considerations](#)
9. [The GRDDL Vocabulary](#)
10. [References](#)

- Appendix: [Transformations for Styling versus data extraction](#)
- Appendix: [Base IRI considerations](#)
- Appendix: [Acknowledgements and Change History](#)

Linked documents:

- Appendix: [About the Mechanical Rules](#)

1. Introduction: Data and Documents

There are many domain-specific languages ("dialects") used in practice among the many XML documents on the web. There are dialects of XHTML, XML and RDF that are used to represent everything from poetry to prose, purchase orders to invoices, spreadsheets to databases, schemas to scripts, and linked lists to ontologies.

While this breadth of expression is quite liberating, inspiring new dialects to represent information, it can be a barrier to understanding across different domains or fields. How, for example, does software discover the author of a poem, a spreadsheet and an ontology? And how can software determine whether authors of each are in fact the same?

The following are examples of how the same musical work might be described in different XML dialects:

iTunes Music Library

```
<key>Artist</key>
  <string>The Jimi Hendrix Experience</string>
<key>Album</key>
  <string>Are You Experienced?</string>
```

Audioscrobbler

```
<album>
  <artist mbid="">The Jimi Hendrix Experience</artist>
  <name>Are You Experienced?</name>
  ...
</album>
```

Atom

```
<entry ... >
  <title>Are You Experienced?</title>
  <author>
    <name>The Jimi Hendrix Experience</name>
  </author>
  ...
</entry>
```

Open Office

```
<office:document-meta ... >
<office:meta>
<dc:title>Are You Experienced?</dc:title>
  <meta:initial-creator>
    The Jimi Hendrix Experience
  </meta:initial-creator>
<dc:creator>The Jimi Hendrix Experience</dc:creator>
</office:meta>
</office:document-meta>
```

Although the examples above are obviously encodings of the same information, there remains no clear mechanism through which computer software might be able to determine this connection.

Resource Descriptions

The Resource Description Framework[RDFC04] provides a standard for making statements about resources in the form of a subject-predicate-object expression. One way to represent the fact "*Are You Experienced?*'s artist is The Jimi Hendrix Experience" in RDF would be as a triple whose subject is *Are You Experienced*, whose predicate is "has artist," and whose object is The Jimi Hendrix Experience. The predicate, "has artist" expresses a relationship between the subject (*Are You Experienced?*) and the object (The Jimi Hendrix

Experience). Using URIs to uniquely identify the album, the artist and even the relationship would facilitate software design because not everyone knows The Jimi Hendrix Experience or even spells its name consistently.

Here's the information contained in the XML fragments above, this time expressed as RDF:

```

<rdf:RDF
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdf:Description rdf:about=
    "http://musicbrainz.org/mm-2.1/album/6b050dcf-7ab1-456d-9e1b-c3c41c18eed2">
    <dc:title>Are You Experienced?</dc:title>
    <foaf:maker>
      <foaf:Agent rdf:about=
        "http://musicbrainz.org/mm-2.1/artist/33b3c323-77c2-417c-a5b4-af7e6a111cc9">
        <foaf:name>The Jimi Hendrix Experience</foaf:name>
      </foaf:Agent>
    </foaf:maker>

  </rdf:Description>
</rdf:RDF>

```

Both the entities (subject and object resources) and relationships (predicates) are identified using unambiguous URIs.

Note that GRDDL follows HTML 4, RDF, and XML Schema in using Internationalized Resource Identifiers, i.e. IRIs [RFC3987]. While in informal usage, this specification uses the more familiar term "URI" interchangeably with the recently standardized term "IRI", the formal rules use the relevant terms precisely.

The publishers of the XML above could also provide the same data in RDF using RDF/XML or one of the other RDF syntaxes. GRDDL provides a relatively inexpensive mechanism for bootstrapping RDF content from uniform XML dialects, shifting the burden from formulating RDF to creating transformation algorithms specifically for each dialect.

GRDDL works by associating transformations for an individual document, either through direct inclusion of references or indirectly through profile and namespace documents. Content authors can nominate the transformations for producing RDF from their content and use GRDDL to refer to them.

Faithful Renditions

By specifying a GRDDL transformation, the author of a document states that the transformation will provide a faithful rendition in RDF of information (or some portion of the information) expressed through the XML dialect used in the source document.

Likewise, by specifying a GRDDL namespace transformation or profile transformation, the creator of that namespace or profile states that the transformation will provide a faithful RDF rendition of a class of source documents which relate to that namespace or profile. A namespace document or a profile document also provide a means for their authors to explain in prose the purpose of the transformation or any policy statements.

Preface and Companion Documents

This GRDDL specification is a concise technical specification of the GRDDL mechanism and its XML syntax. It specifies the GRDDL syntax to use in valid XHTML and well-formed XML documents, as well as how to encode GRDDL into namespaces and HTML profiles. Discussions of the GRDDL transformation link and security issues are also covered. Appendices provide links to extended examples and existing software and services that employ GRDDL.

GRDDL Primer

The GRDDL Primer [primer] is a step-by-step tutorial on the GRDDL mechanism. It develops a number of examples from the GRDDL Use Cases document to illustrate GRDDL techniques for associating documents with transformations for extracting RDF.

GRDDL Use Cases

The use cases document [usecases] collects a number of use cases with their goals and requirements for GRDDL. These use cases also illustrate how XML and XHTML documents can be decorated with microformats, Embedded RDF or RDFa statements to support GRDDL transformations in charge of extracting valuable data that can then be used to automate a variety of tasks.

GRDDL Test Cases

The GRDDL Test Cases [GRDDL-TESTS] provides a collection of tests illustrating this specification. Some of the tests may help clarify the intended reading of the normative text.

2. Adding GRDDL to well-formed XML

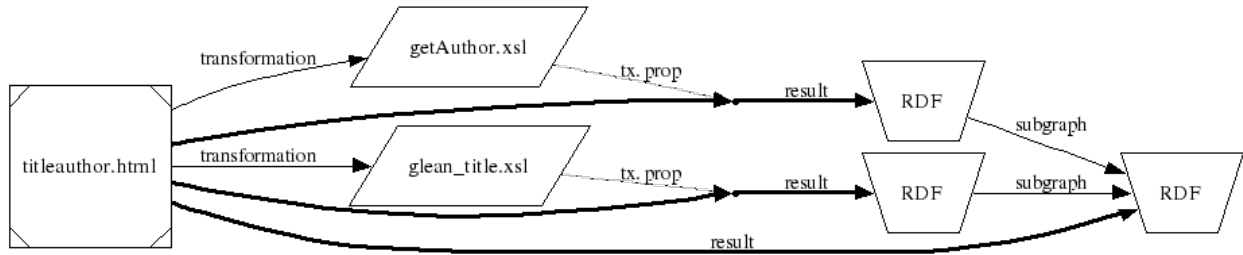
The general form of associating a GRDDL transformation link with a well-formed XML document is adding to the root element a `grddl` namespace declaration and a `grddl:transformation` attribute whose value is an IRI reference, or list of IRI references, that refer to executable scripts or programs which are expected to transform the source document into RDF. This method is suitable for use with any XML dialects that can accommodate an extra namespace-qualified attribute on the root element.

For example, this XML document, located at <http://www.w3.org/2001/sw/grddl-wg/td/titleauthor.html>, is linked to two GRDDL

transformations:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:grddl="http://www.w3.org/2003/g/data-view#"
      grddl:transformation="glean_title.xml
                        http://www.w3.org/2001/sw/grddl-wg/td/getAuthor.xml"
>
<head>
<title>Are You Experienced?</title>
[...]
```

1. It is linked to the transformation identified by `http://www.w3.org/2001/sw/grddl-wg/td/getAuthor.xml`.
2. To resolve the relative URI reference `glean_title.xml` to absolute form, we use the base URI of this XML element, `http://www.w3.org/2001/sw/grddl-wg/td/titleauthor.html`. Then this document is also linked to the GRDDL transformation identified by the absolute form, `http://www.w3.org/2001/sw/grddl-wg/td/glean_title.xml`.



extracting title and author information

(svg)

As you will see in later sections, there are other ways to add GRDDL to HTML documents, especially designed to leverage HTML's existing capabilities and thereby overcome constraints imposed by the XML DTDs for some dialects of HTML. See [Using GRDDL with valid XHTML](#) and [GRDDL for HTML Profiles](#).

The formal specification of this markup is given below. *An informative mechanical version of each rule is given with the premise and the conclusion written as SPARQL graph patterns* [\[SPARQL\]](#). See the [Mechanical Rules](#) appendix for namespace prefix bindings and further explanation. These are included for those readers who find them helpful. Other readers are encouraged to ignore them.

Normative Statement	Mechanical Rule (Informative)
<p>Given an XPath [XPATH] root node <i>N</i> with root element <i>E</i>, if the expression</p> <pre>/*/*[local-name()="transformation" and namespace-uri()= "http://www.w3.org/2003/g/data-view#"]</pre> <p>matches an attribute of an element <i>E</i>, then for each space-separated token <i>REF</i> in the value of that attribute, the resource identified [WEBARCH] by the absolute form (see section 5.2 Relative Resolution in [RFC3986]) of <i>REF</i> with respect to the base IRI [RFC3987], [XMLBASE] of <i>E</i> is a GRDDL transformation of <i>N</i>.</p> <p>Space-separated tokens are the maximal non-empty subsequences not containing the whitespace characters #x9, #xA, #xD or #x20.</p>	<pre>(?N "/"*) gspec:xpath ?E. (?N "/*/*[local-name()="transformation" and namespace-uri()= "http://www.w3.org/2003/g/data-view#"]"*) gspec:xpath [fn:string ?V]. ?V fn:normalize-space ?Vnorm. (?Vnorm "[\t\r\n]+") fn:tokenize [list:member ?REF]. ?E fn:base-uri ?BASE. (?REF ?BASE) fn:resolve-uri ?TXURI. ?TX log:uri ?TXURI.</pre> <hr/> <pre>?N grddl:transformation ?TX.</pre>

The `glean_title.xml` transformation computes the following RDF/XML document, given the XML document above as input:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description rdf:about="">
<dc:title>Are You Experienced?</dc:title>
</rdf:Description>
</rdf:RDF>
```

The graph serialized by that document is a **GRDDL result** of the resource identified by `http://www.w3.org/2001/sw/grddl-wg/td/titleauthor.html`. Note that this serialization of the graph contains a relative URI reference (in the value of the `rdf:about` attribute). The base IRI for interpreting relative IRI references in a serialization of a graph produced by a GRDDL transformation is the base IRI of the source document.

The `glean_title.xml` resource specifies a function from XPath document nodes to RDF/XML documents, and hence to RDF graphs; this function is called the **transformation property** of the XSLT document. See the [GRDDL Transformations section](#) for more details.

The general rule for using GRDDL with well-formed XML is:

If an information resource([WEBARCH] , section 2.2) <i>IR</i> is represented	<pre>?IR log:uri [fn:doc ?R].</pre>
---	---------------------------------------

by an XML document with an XPath root node <i>R</i> , and <i>R</i> has a GRDDL transformation with a transformation property <i>TP</i> , and <i>TP</i> applied to <i>R</i> gives an RDF Graph [RDFC04] <i>G</i> , then <i>G</i> is a GRDDL result of <i>IR</i> .	<pre> ?R grddl:transformation [grddl:transformationProperty ?TP]. ?R ?TP ?G. ?IR grddl:result ?G . </pre>
--	--

The `titleauthor.html` resource has another GRDDL result via the `getAuthor.xml` transformation. These results can be merged together into another result, by this rule:

If <i>F</i> and <i>G</i> are GRDDL results of <i>IR</i> , then the merge [RDF-MT] of <i>F</i> and <i>G</i> is also a GRDDL result of <i>IR</i> .	<pre> ?IR grddl:result ?F, ?G. (?F ?G) log:conjunction ?H. ?IR grddl:result ?H. </pre>
--	---

3. Using GRDDL with XML Namespace Documents

Transformations can be associated not only with individual documents but also with whole dialects that share an XML namespace. Any resource available for retrieval from a namespace URI is a **namespace document** (cf. section [4.5.4. Namespace documents](#) in [\[WEBARCH\]](#)). For example, a namespace document may have an XML Schema representation or an RDF Schema representation, or perhaps both, using [content negotiation](#).

To associate a GRDDL transformation with a whole dialect, include a `grddl:namespaceTransformation` property in a GRDDL result of the namespace document.

For example, consider this privacy policy written in P3Q, a contrived analog to P3P[\[P3P\]](#):

```

<POLICIES xmlns="http://www.w3.org/2004/01/rdxh/p3q-ns-example">
  <EXPIRY max-age="604800"/>
  ...

```

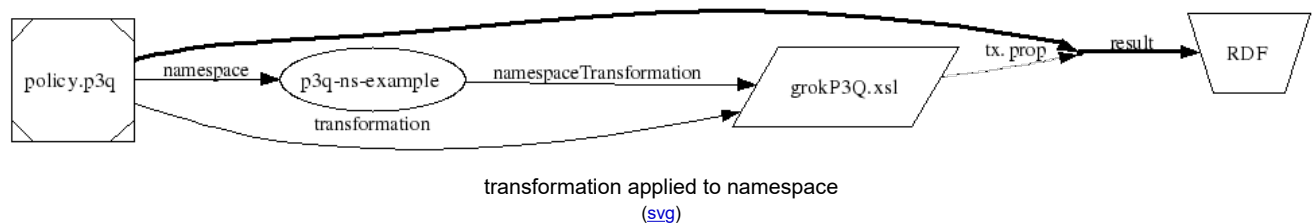
The namespace document for P3Q relates the `grokP3Q.xml` transformation to all P3Q documents:

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dataview="http://www.w3.org/2003/g/data-view#">
  <rdf:Description rdf:about="http://www.w3.org/2004/01/rdxh/p3q-ns-example">
    <dataview:namespaceTransformation
      rdf:resource="http://www.w3.org/2004/01/rdxh/grokP3Q.xml"/>
  </rdf:Description>
</rdf:RDF>

```

That is: every document whose root namespace name is `...p3q-ns-example` has `grokP3Q.xml` as a **GRDDL transformation** implicitly, as illustrated in this figure:



Some namespace documents, such as the XHTML namespace document <http://www.w3.org/1999/xhtml> have very many references to them. If GRDDL-aware agents were to retrieve these documents every time they processed a document referring to them, the origin servers of those documents could become overloaded. GRDDL-aware agents therefore should not retrieve such documents on every reference and should retain some cache or local memory of the transformations those documents indicate should be applied. To avoid misrepresentation of published information, GRDDL-aware agents should ensure that this local memory is up to date and should support user options to configure or disable the cache. See also section [3.1. Using a URI to Access a Resource](#) of [\[WEBARCH\]](#).

The general case of namespace transformations is:

Normative Statement	Mechanical Rule (Informative)
<p>If</p> <ul style="list-style-type: none"> an information resource <i>NSDOC</i>, identified by an IRI <i>NS</i> has a GRDDL result that includes a triple whose <ul style="list-style-type: none"> subject is <i>NSDOC</i>, whose 	<pre> ?NSDOC log:uri ?NS; grddl:result [log:includes [rdf:subject ?NSDOC; rdf:predicate grddl:namespaceTransformation; rdf:object ?TX]]. </pre>

<ul style="list-style-type: none"> ◦ predicate is the property <code><http://www.w3.org/2003/g/data-view#namespaceTransformation></code>, and whose ◦ object is <i>TX</i>, • and an information resource <i>IR</i> has an XML representation with root node <i>NODE</i> and with a root element with a namespace name <i>NS</i>, <p>then <i>TX</i> is a GRDDL transformation of <i>NODE</i>.</p>	<pre>?IR log:uri [fn:doc ?NODE]. (?NODE "/"*) gspec:xpath ?E. ?E fn:namespace-uri ?NS.</pre> <hr/> <pre>?NODE grddl:transformation ?TX.</pre>
--	--

Note that as a base case, the result of parsing an RDF/XML document is a GRDDL result of that document:

Normative Statement	Mechanical Rule (Informative)
If an information resource <i>IR</i> is represented by a conforming RDF/XML document [RDFX] , then the RDF graph represented by that document is a GRDDL result of <i>IR</i> .	<pre>?IR log:uri [fn:doc [gspec:rdfParse ?G]].</pre> <hr/> <pre>?IR grddl:result ?G.</pre>

Note that while an `application/rdf+xml` media type is one indication that a document is RDF/XML, section [7.2.1 Grammar start](#) of [\[RDFX\]](#) leaves open "other means" by which an RDF/XML document may be identified. For the purposes of the rule above, a root element whose local name is `rdf` and whose namespace URI is `http://www.w3.org/1999/02/22-rdf-syntax-ns#` is such a means. For a case in point, see the [grddl-on-rdf-xml-media-type](#) test case.

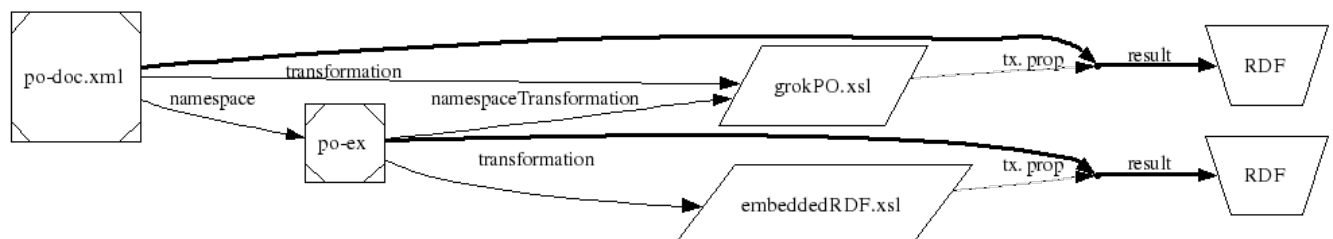
Example: Using GRDDL with an XML Schema namespace document

A namespace transformation link may be discoverable by transforming the namespace document itself. Note that this means that namespace documents need not be written in RDF/XML directly.

Consider a purchase order that has a namespace document represented in XML Schema, where the XML Schema bears a `data-view:transformation` attribute licensing extraction of statements that include `namespaceTransformation` statements:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://.../Order-1.0"
  targetNamespace="http://.../Order-1.0"
  version="1.0"
  ...
  xmlns:data-view="http://www.w3.org/2003/g/data-view#"
  data-view:transformation="http://www.w3.org/2003/g/embeddedRDF.xsl" >
  <xsd:element name="Order" type="OrderType">
    <xsd:annotation
      <xsd:documentation>This element is the root element.</xsd:documentation>
    </xsd:annotation>
    ...
  </xsd:element>
  <xsd:annotation>
    <xsd:appinfo>
      <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        <rdf:Description rdf:about="http://www.w3.org/2003/g/po-ex">
          <data-view:namespaceTransformation
            rdf:resource="grokPO.xsl" />
        </rdf:Description>
      </rdf:RDF>
    </xsd:appinfo>
  </xsd:annotation>
  ...
```

Every purchase order using that schema as a namespace document is linked to the `grokPO.xsl` transformation, as illustrated below:



using GRDDL with an XML Schema

[\(svg\)](#)

4. Using GRDDL with valid XHTML

To accommodate the DTD-based syntax of XHTML[\[XHTML\]](#), which precludes using attributes from foreign namespaces, we use <http://www.w3.org/2003/g/data-view> as a metadata profile (cf. section [7.4.4.3 Meta data profiles](#) of [\[HTML4\]](#)).

The general form of adding a GRDDL assertion to a valid XHTML document is by specifying the GRDDL profile in the `profile` attribute of the `head` element, and `transformation` as the value of the `rel` attribute of a `link` or a `element` whose `href` attribute value is an IRI reference that refers to an executable script or program which is expected to transform the source document into RDF. This method is suitable for use with valid XHTML documents which are constrained by an XML DTD.

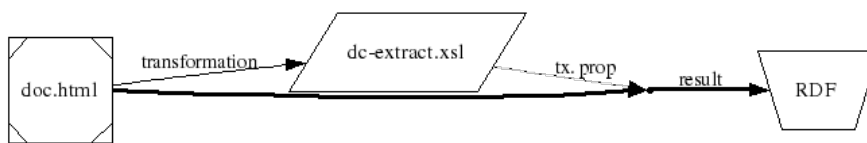
An example Dublin Core META transformation

For example, this document follows the conventions of [RFC2731](#), and it explicitly uses the GRDDL profile and links to an XSLT transformation to RDF/XML to signal that the transformation is a faithful rendition:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head profile="http://www.w3.org/2003/g/data-view">
    <title>Some Document</title>

    <link rel="transformation"
      href="http://www.w3.org/2000/06/dc-extract/dc-extract.xsl" />
    <meta name="DC.Subject"
      content="ADAM; Simple Search; Index+; prototype" />
    ...
  </head>
  ...
</html>
```

The figure below shows the source document, the `dc-extract.xsl` transformation, and the GRDDL result:



Decoding HTML meta-data to RDF

(svg)

This is what the data looks like in RDF/XML:

```
<rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:about=""
    <dc:subject>ADAM; Simple Search; Index+; prototype</dc:subject>
  </rdf:Description>
</rdf:RDF>
```

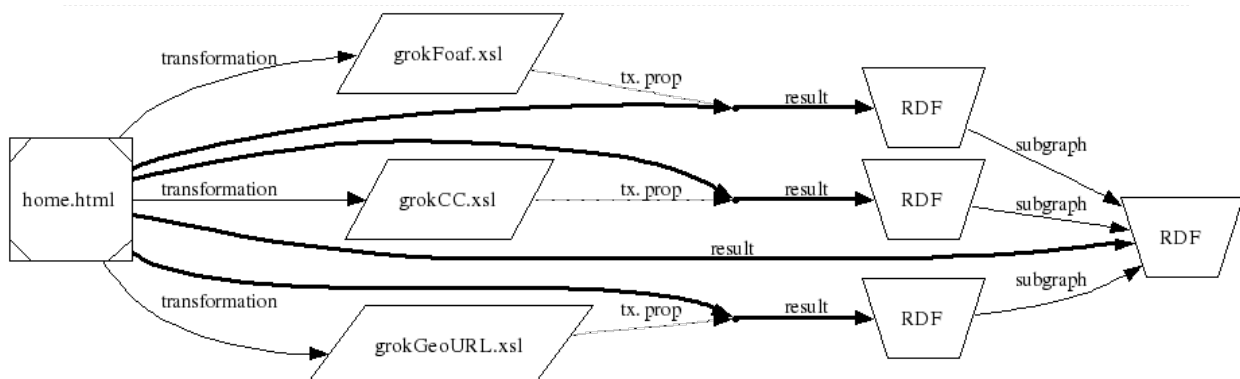
Multiple transformations in XHTML

An XHTML document may conform to a number of dialects simultaneously and link to more than one GRDDL transformation. However, since the `href` attribute of the `link` and `a` elements accept only a single IRI reference, multiple instances of these elements must be used to assert multiple links:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head profile="http://www.w3.org/2003/g/data-view">
  <title>Joe Lambda's Home page [an example of RDF in XHTML]</title>

  <link rel="transformation" href="http://www.w3.org/2003/12/rdf-in-xhtml-xslts/grokFOAF.xsl" />
  <link rel="transformation" href="http://www.w3.org/2003/12/rdf-in-xhtml-xslts/grokCC.xsl" />
  <link rel="transformation" href="http://www.w3.org/2003/12/rdf-in-xhtml-xslts/grokGeoURL.xsl" />
  ...

```



multiple transformations

(svg)

Rules for GRDDL with valid XHTML

The general rule is:

<p>Given XPath root node <i>N</i>, if <i>N</i> has metadata profile name <code>http://www.w3.org/2003/g/data-view</code>, then for each <i>a</i> and link descendant element <i>E</i> whose rel attribute[HTML4] has transformation as one of its space separated values the resource identified by the absolute form of the <code>href</code> attribute with respect to the base IRI of <i>E</i> is a GRDDL transformation of <i>N</i>.</p>	<pre>?N gspec:profileName "http://www.w3.org/2003/g/data-view". (?N """/**[namespace-uri()="http://www.w3.org/1999/xhtml" and (local-name() = "a" or local-name() = "link")"") gspec:xpath ?E. (?E "@rel") gspec:xpath [fn:string [fn:normalize-space ?E_REL]]. (?E_REL "[\t\r\n]+") fn:tokenize [list:member "transformation"]. (?E "@href") gspec:xpath [fn:string ?T_REF]. ?E gspec:htmlBase ?BASE. (?T_REF ?BASE) fn:resolve-uri ?TURI. ?T log:uri ?TURI. ?N grddl:transformation ?T.</pre>
---	--

Note that the base IRI of an element node in an XHTML document may be influenced by factors such as a [base element](#)[\[HTML4\]](#) [Retrieval URIRFC3986](#), etc. See the [Base IRI considerations](#) appendix and test cases such as [htmlbase1](#) for further clarification.

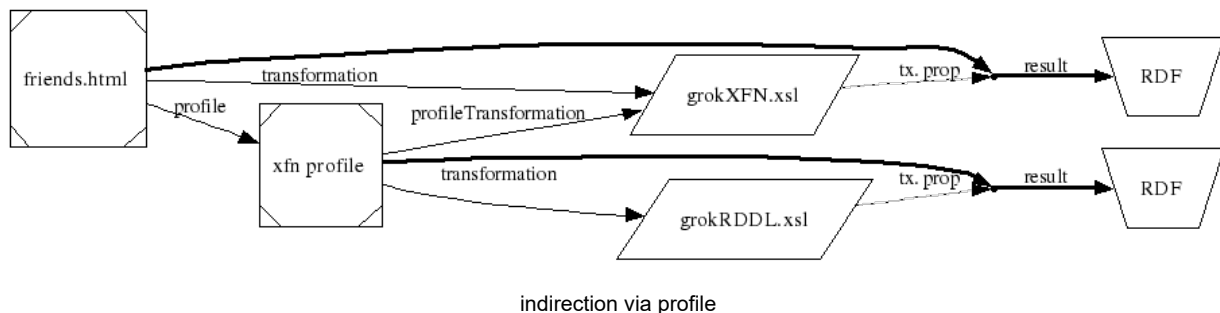
The rule above depends on the following formalization of metadata profiles in XHTML:

<p>Given an XPath root node <i>N</i> of an XHTML document (that is, an XML document whose root element has a local name of <code>html</code> and a namespace name of <code>http://www.w3.org/1999/xhtml</code>) for each space-separated token <i>REF</i> in the value of the <code>profile</code> attribute[HTML4] of the <code>head</code> element <i>E</i>, the absolute form of <i>REF</i> with respect to the base IRI of <i>E</i> is a metadata profile name of <i>N</i>.</p>	<pre>(?N "" *[local-name()="html" and namespace-uri()="http://www.w3.org/1999/xhtml"] / *[local-name()="head" and namespace-uri()="http://www.w3.org/1999/xhtml"]""") gspec:xpath ?E. (?E "@profile") gspec:xpath [fn:string ?V]. ?E fn:base-uri ?BASE. ?V fn:normalize-space ?Vnorm. (?Vnorm "[\t\r\n]+") fn:tokenize [list:member ?P_REF]. (?P_REF ?BASE) fn:resolve-uri ?PROFID. ?N gspec:profileName ?PROFID.</pre>
--	--

5. GRDDL for HTML Profiles

XHTML provides the profile mechanism to link to the meaning of properties and the set of legal values for those properties. As with namespace documents, a profile document can effectively be written using XHTML with embedded RDF statements and a GRDDL transformation to extract the definition of terms that are applicable. Those terms can then be used in an XHTML document to convey profile-dependent meaning. As discussed in [Using GRDDL with valid XHTML](#), the GRDDL profile can be used with XHTML documents to apply GRDDL semantics over link elements where the value of `rel` attribute is transformation. This very powerful and flexible mechanism integrates well with [microformat profiles](#)[\[MF-RDF-FAQ\]](#) which overlay the normally semantically-poor HTML markup.

The following diagram illustrates an XFN document[\[XFN\]](#), `friends.html` associated with the `grokXFN.xsl` transformation indirectly via an XFN profile.



(svg)

Adding a GRDDL `profileTransformation` assertion to a profile document is much like [adding a namespaceTransformation assertion to a namespace document](#). For a dialect defined by a valid XHTML profile documents, add `profile="http://www.w3.org/2003/g/data-view"` to the `head` element and make a link of type `profileTransformation` to the transformation of the dialect.

The general rule is:

If	<pre>?PDOC log:uri ?PNAME; grddl:result [</pre>
----	---

<ul style="list-style-type: none"> • an information resource <i>PDOC</i>, identified by an IRI <i>PNAME</i> has a GRDDL result that includes a triple whose <ul style="list-style-type: none"> ◦ subject is <i>PDOC</i>, whose ◦ predicate is the property <code><http://www.w3.org/2003/g/data-view#profileTransformation></code>, and whose ◦ object is <i>TX</i>, • and an information resource <i>IR</i> has an XML representation with XPath root node <i>NODE</i> that has a metadata profile name <i>PNAME</i>, <p>then <i>TX</i> is a GRDDL transformation of <i>NODE</i>.</p>	<pre>log:includes [rdf:subject ?PDOC; rdf:predicate grddl:profileTransformation; rdf:object ?TX]]. ?IR log:uri [fn:doc ?NODE]. ?NODE gspec:profileName ?PNAME.</pre> <hr/> <pre>?NODE grddl:transformation ?TX.</pre>
--	---

6. GRDDL Transformations

As noted above, each GRDDL transformation specifies a **transformation property**, a function from XPath document nodes to RDF graphs. This function need not be total; it may have a domain smaller than all XML document nodes. For example, use of `xsl:message with terminate="yes"` may be used to signal that the input is outside the domain of the transformation.

Developers of transformations should make available representations in widely-supported formats. XSLT version 1 [[XSLT1](#)] is the format most widely supported by GRDDL-aware agents as of this writing, though though XSLT2 [[XSLT2](#)] deployment is increasing. While technically Javascript, C, or virtually any other programming language may be used to express transformations for GRDDL, XSLT is specifically designed to express XML to XML transformations and has some good safety characteristics; XQuery has similar characteristics to XSLT, though use of XQuery in GRDDL implementation is less widely deployed at the time of this writing.

<p>If</p> <ul style="list-style-type: none"> • <i>RDFXML</i> is the root XPath node of a conforming RDF/XML document [RDFX] that represents an RDF Graph <i>G</i>, and • <i>R</i> is the root node of some XML document and <i>TXNODE</i> is the root node of an XSLT transformation [XSLT1], and • <i>RDFXML</i> is the root node of the XSLT result tree when <i>TXNODE</i> is applied to <i>R</i>, and • <i>TXDOC</i> is an information resource with <i>transformation property</i> <i>TP</i> represented by an XML document with root node <i>TXNODE</i> <p>then <i>TP</i> relates <i>R</i> to <i>G</i>.</p>	<pre>?RDFXML gspec:rdfParse ?G. ({TXNODE ?R}) gspec:resultTree ?RDFXML. ?TXDOC grddl:transformationProperty ?TP; log:uri [fn:doc ?TXNODE].</pre> <hr/> <pre>?R ?TP ?G</pre>
---	---

The rule above covers the case of a *transformation property* that relates an XPath document node to an RDF graph via an RDF/XML document. Transformations may use other, unspecified, mechanisms. For example, see [test #atomttl1](#), in which the the `media-type` attribute of the `xsl:output` element bears a "text/rdf+n3" value to indicate a media type other than "application/rdf+xml". GRDDL agents that can process such a media type can then produce an RDF graph in accordance with the media type. Non-XSLT transforms may indicate the RDF graph in some other, unspecified, fashion.

At present, when an information resource is represented by an XML document, the corresponding XPath data model may not be fully determined, depending on, for example, whether an agent elaborates inclusions, parameter entities, fixed and default attributes, or checks digital signatures. Put another way, if an author takes responsibility for the information in an XML document, for what information exactly is the author taking responsibility? And how can the author ensure that a GRDDL transformation is able to meet GRDDL's [Faithful Rendition assurance](#)?

This specification is silent on the question of which XML processors are employed by or for GRDDL-aware agents. Whether or not processing of XInclude, XML Validity, XML Schema Validity, XML Signatures or XML Decryption take place is currently unspecified. However, this specification anticipates that the resolution of TAG issue [xmlFunctions-34](#) and the definition, by the [XML Processing Model Working Group](#), of a default processing model will provide further clarification and guidance, and GRDDL-aware agents are expected to comply with such guidance if it is issued. There is no universal expectation that an XSLT processor will call on such processing before executing a GRDDL transformation. Therefore, it is suggested that GRDDL transformations be written so that they perform all expected pre-processing, including processing of related DTDs, Schemas and namespaces. Such measure can be avoided for documents which do not require such pre-processing to yield an infoset that is faithful. That is, for documents which do not reference XInclude, DTDs, XML Schemas and so on.

Document authors, particularly XHTML document authors, who wish their documents to be unambiguous when used with GRDDL should avoid dependencies on an external [DTD subset](#); specifically:

- Explicitly include the XHTML namespace declaration in an XHTML document, or an appropriate namespace in an XML document.
- Avoid use of entity references, except those listed in [section 4.6 of the XML specification](#).
- And, more generally, follow the rules listed for [the standalone document](#) validity constraint.

XProc: An XML Pipeline Language [[XPROC](#)], a language for describing operations to be performed on XML documents, has recently been published as a W3C Working Draft. It merits consideration for expressing more complex or sophisticated transformations which require control over the flow of processing through a variety of XML processing tools. Using XProc, one could apply a sequence of operations such XInclude, validation, and transformation to a document, aborting if the result of an intermediate stage is not valid, for example.

7. GRDDL-Aware Agents

A **GRDDL-aware agent** is a software module that computes **GRDDL results** of information resources.

For example, a SPARQL query service might use a GRDDL-aware agent for collecting RDF data. Or a Web browser might serve as a GRDDL-aware agent for the purpose of collecting calendar and contact data. The appropriate policy, for which results to compute and when, is likely to involve waiting for a signal from user more in the Web browser case than in the query service case.

Subject to [security considerations](#) below and local policy as expressed in its configuration, given an information resource *IR*, and an XPath node *N* for a representation of *IR*, a GRDDL-aware agent **should**:

1. Find each transformation associated with *N*, i.e.
 1. each transformation associated with *N* via the `grddl:transformation` attribute as in the [Adding GRDDL to well-formed XML](#) section
 2. each transformation associated with *N* via HTML links of type `transformation`, provided the document bears the `http://www.w3.org/2003/g/data-view` profile, as in the [Using GRDDL with valid XHTML](#) section.
 3. each transformation indicated by any available namespace document, as in the [GRDDL for XML Namespaces](#) section.
 4. each transformation indicated by any XHTML profiles, as in the [GRDDL for HTML Profiles](#) section.
2. Selectively apply any or all discovered transformations to obtain GRDDL results. Note selection may be guided by the agent's capabilities, local security policies and possibly user/client intervention.
3. Merge those GRDDL results.

Note that discovery by namespace or profile document is recursive; Loops in the profile/namespace structure should be detected in order to avoid infinite recursion.

Example: A GRDDL-aware Agent protocol trace

While this declarative specification of GRDDL allows a variety of implementation strategies, in this example we trace the behavior common to a number of typical implementations.

Consider a GRDDL-aware agent that is asked for results from `http://www.w3.org/2003/g/po-doc.xml`. It starts by dereferencing that URI, noting that RDF/XML, HTML, and XML are acceptable representations:

```
[00:00.000 - client connection from 127.0.0.1:39645]
GET http://www.w3.org/2003/g/po-doc.xml HTTP/1.1
Host: www.w3.org
Accept: application/rdf+xml,application/xml,text/xml,application/xhtml+xml,text/html

[00:00.055 - server connected]
HTTP/1.1 200 OK
Last-Modified: Tue, 07 Dec 2004 22:59:02 GMT
Content-Length: 1302
Content-Type: application/xml; qs=0.9

<purchaseOrder orderDate="1999-10-20"
  xmlns="http://www.w3.org/2003/g/po-ex">
  <shipTo country="US">
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
  ...
```

The XML document that comes back has no explicit transformation markup, but the rules in [the XML Namespaces section](#) suggest looking up results from the namespace document:

```
[00:00.000 - client connection from 127.0.0.1:39647]
GET http://www.w3.org/2003/g/po-ex HTTP/1.1
Host: www.w3.org
Accept: application/rdf+xml,application/xml,text/xml,application/xhtml+xml,text/html

[00:00.051 - server connected]
HTTP/1.1 200 OK
Content-Location: po-ex.xsd
Last-Modified: Tue, 07 Dec 2004 23:18:25 GMT
Content-Length: 2624
Content-Type: application/xml; qs=0.9

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:po="http://www.w3.org/2003/g/po-ex"
  targetNamespace="http://www.w3.org/2003/g/po-ex"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"

  xmlns:data-view="http://www.w3.org/2003/g/data-view#"
  data-view:transformation="http://www.w3.org/2003/g/embeddedRDF.xsl"
>

<xs:annotation>
  <xs:appinfo>
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
      <rdf:Description rdf:about="http://www.w3.org/2003/g/po-ex">
        <data-view:namespaceTransformation
          rdf:resource="grokPO.xsl" />
        </rdf:Description>
      </rdf:RDF>
    </xs:appinfo>
  </xs:annotation>
  ...
```

We don't yet have a result in the form of an RDF/XML document, but this time we find an explicit `transformation` attribute in the GRDDL namespace, so we follow that link, noting that we accept XML representations:

```

00:00.000 - client connection from 127.0.0.1:39649]
GET http://www.w3.org/2003/g/embeddedRDF.xml HTTP/1.1
Host: www.w3.org
Accept: application/xml

[00:00.054 - server connected]
HTTP/1.1 200 OK
Last-Modified: Wed, 23 Mar 2005 18:49:12 GMT
Content-Length: 797
Content-Type: application/xml; qs=0.9

<xsl:transform
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  ...

```

Applying that transformation yields...

```

<rdf:RDF
  xmlns:data-view="http://www.w3.org/2003/g/data-view#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <rdf:Description rdf:about="http://www.w3.org/2003/g/po-ex">
    <data-view:namespaceTransformation rdf:resource="http://www.w3.org/2003/g/grokPO.xml"/>
  </rdf:Description>
</rdf:RDF>

```

... which tells us that `.../grokPO.xml` is a transformation for all documents in the `.../po-ex` namespace.

Continuing recursively, we examine the namespace document for `po-ex.xsd`. As this is a well-known namespace document, following the [Security considerations section](#), we note the last modified date of our cached copy in the request, and the origin server lets us know that our copy is current:

```

[00:00.000 - client connection from 127.0.0.1:39651]
GET http://www.w3.org/2001/XMLSchema HTTP/1.1
Host: www.w3.org
Accept: application/rdf+xml,application/xml,text/xml,application/xhtml+xml,text/html
If-modified-since: Fri, 16 Dec 2005 14:19:38 GMT

[00:00.047 - server connected]
HTTP/1.1 304 Not Modified
Content-Location: XMLSchema.html
Expires: Wed, 07 Feb 2007 15:09:29 GMT
Cache-Control: max-age=21600
Vary: negotiate, accept, accept-charset

```

Since our cached copy of the XML Schema namespace document shows no associated GRDDL transformation, we return to the namespace transformation from `po-ex`, i.e. `grokPO.xml`:

```

[00:00.000 - client connection from 127.0.0.1:39653]
GET http://www.w3.org/2003/g/grokPO.xml HTTP/1.1
Host: www.w3.org
Accept: application/xml

[00:00.048 - server connected]
HTTP/1.1 200 OK
Last-Modified: Tue, 07 Dec 2004 23:33:28 GMT
Content-Length: 1739
Content-Type: application/xml; qs=0.9

<xsl:transform
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:po="http://www.w3.org/2003/g/po-ex"
  xmlns:poF="http://www.w3.org/2003/g/po-ex#"
>

<xsl:output method="xml" indent="yes" />

<div xmlns="http://www.w3.org/1999/xhtml">
<h1>grokPO.xml -- interpret purchase order format as RDF</h1>
...

```

Applying this transformation to `po-doc.xml` yields RDF/XML; we parse this to an RDF graph (using the URI of the source document, `http://www.w3.org/2003/g/po-doc.xml`, as the base URI) and return the graph as a GRDDL result of `po-doc.xml`:

```

<rdf:RDF
  xmlns:poF="http://www.w3.org/2003/g/po-ex#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <rdf:Description rdf:nodeID="hOhqYGHx9">
    <poF:city>Mill Valley</poF:city>
    <poF:state>CA</poF:state>
    <poF:zip>90952</poF:zip>
    <poF:street>123 Maple Street</poF:street>
    <poF:name>Alice Smith</poF:name>
  </rdf:Description>
  ...

```

HTTP trace data was collected via [TCPWatch](#) by Shane Hathaway. For more details, see [HTTP tracing in the GRDDL test materials](#).

8. Security considerations

The execution of general-purpose programming languages as interpreters for transformations exposes serious security risks. Designers of GRDDL-aware agents are advised to guard against simply sending GRDDL transformations to "off-the-shelf" interpreters. While it is usually safe to pass documents from trusted sources through a GRDDL transformation, implementors should consider all of the following before adding the ability to execute arbitrary GRDDL transformations linked from arbitrary Web documents.

GRDDL, like many Web technologies, fundamentally relies on the dereferencing of URIs. Writers of GRDDL transformations are advised against employing URL operations which are potentially dangerous, because these operations are more likely to be unavailable in secure GRDDL implementations. Software executing GRDDL transformations are advised to either completely disable all potentially dangerous URL operations or take special care not to delegate any special authority to their operation. In particular, operations to read or write URLs are more safely executed with the privileges associated with an untrusted party, rather than the current user. Such disabling and/or checking should be done completely outside of the reach of the transformation language itself; care should be taken to insure that no method exists for re-enabling full-function versions of these operators.

The remainder of this section outlines some, though probably not all, of the possible problems with the execution of GRDDL transformations, with particular reference to transformations in XSLT.

1. With unconstrained use of GRDDL, untrusted transformations may access URLs which the end-user has read or write permission, while the author of the transformation does not. This is particularly pertinent for URLs from the file: scheme; but many other schemes are also impacted. The untrusted code may, having read documents which the author did not have permission to access, transmit the content of the documents, to arbitrary Web servers by encoding the contents within a URL, that may be passed to the server.
2. Dangerous operations in the XSLT language include, but may not be limited to, the operations involving getting a URL: `document()`, `doc()`, `unparsed-text()` and `unparsed-text-available()`, and `xsl:result-document` which involves writing to a URL. `xsl:include` and `xsl:import` present fewer risks if they are processed before execution of the transformation, rather than during it.
3. Some transformation language implementations may provide facilities for loading and executing other programming language code. For example, an XSLT implementation may provide a method for executing Java code. Such facilities are obviously open to abuse. Designers of GRDDL transformations are advised against making use of such features. Besides being implementation-specific, they are more likely to be unavailable in secure implementations of the transformation language. The use of such operators in software executing GRDDL transformations should protect against such operators in case they are encountered.
4. XSLT implementations often provide their own extensions. Designers of GRDDL transformations are advised not make use of extensions because they are not guaranteed to be present in all implementations. Software executing GRDDL transformations should make sure that extensions are secure and do not present any kind of threat.
5. Since it is possible to write transformations that inordinately consume system resources or that loop indefinitely. Both types of transformations have the potential to cause damage if sent to unsuspecting recipients. Designers of GRDDL transformations are advised to avoid the construction and dissemination of such transformations. Software executing GRDDL transformations should provide appropriate mechanisms to abort processing after a reasonable amount of time has elapsed. In addition, GRDDL software should be limited to the consumption of only a reasonable amount of any given system resource.
6. Finally, bugs may exist in some interpreters of a transformation language which might be exploited to gain unauthorized access to a recipient's system. Apart from noting this possibility, no specific action is advised to take to prevent this aside from timely correction of such bugs as they are discovered.

9. The GRDDL Vocabulary

The following is excerpted from the GRDDL profile/namespace document:

This document, <http://www.w3.org/2003/g/data-view>, is a metadata profile in the sense of the HTML specification, in section [7.4.4.3 Meta data profiles](#).

The following term is introduced here as an XHTML link relationship name and RDF property name:

- `transformation`: relates a source document to a transformation, usually represented in [XSLT](#), that relates the source document syntax to the RDF graph syntax. domain: [RootNode](#); range: [Transformation](#)

The following terms are introduced here as RDF properties:

- `namespaceTransformation`: relates a namespace to a transformation for all documents in that namespace. range: [Transformation](#)
- `profileTransformation`: relates a profile document to a transformation for all documents bearing that profile. range: [Transformation](#)
- `result`: an RDF graph obtained from an information resource by directly parsing a representation in the standard RDF/XML syntax or indirectly by parsing some other dialect using a transformation nominated by the document. domain: [InformationResource](#); range: [RDFGraph](#)
- `transformationProperty` relates a transformation to the algorithm specified by the property that computes an RDF graph from an XML document node domain: [Transformation](#) range: [TransformationProperty](#)
- `Transformation` an [InformationResource](#) that specifies a transformation from a set of XML documents to RDF graphs. Each Transformation has at least one [transformationProperty](#) that is a [TransformationProperty](#).
- `TransformationProperty` a [FunctionalProperty](#) that relates [XML document root nodes](#) to [RDF graphs](#)

The following terms are bound to concepts from existing standards:

- `RootNode` the root of the tree in the XPath data model, per [section 5.1 Root Node in XML Path Language \(XPath\) Version 1.0](#)
- `RDFGraph` a set of RDF triples, per [definition in Resource Description Framework \(RDF\): Concepts and Abstract Syntax](#)
- `InformationResource` A resource which has the property that all of its essential characteristics can be conveyed in a message, per [definition in Architecture of the World Wide Web, Volume One](#)

The namespace document includes RDF data about the terms in the GRDDL Vocabulary, but these RDF data do not include any triples whose predicate is `grddl:profileTransformation`.

In the section on [Using GRDDL with XML Namespace Documents](#), only explicit `grddl:namespaceTransformation` triples satisfy the premise of the rule. Likewise, `grddl:profileTransformation` triples must be explicit in the GRDDL result of a profile document in order to satisfy the premise of the rule in the section on and on [GRDDL for HTML Profiles](#). Authors of GRDDL source documents are advised against using RDFS or OWL expressions which imply such triples but do not explicitly state them.

10. References

Normative References

RFC3987

[Internationalized Resource Identifiers \(IRIs\)](#), Internet RFC 3987 January 2005. Duerst, Suignard

RFC3986

[Uniform Resource Identifier \(URI\): Generic Syntax](#), Internet RFC3986 January 2005. Berners-Lee, Fielding, Masinter

WEBARCH

[Architecture of the World Wide Web, Volume One](#), N. Walsh, I. Jacobs, Editors, W3C Recommendation, 15 December 2004, <http://www.w3.org/TR/2004/REC-webarch-20041215/> . [Latest version](#) available at <http://www.w3.org/TR/webarch/> .

RDFC04

[Resource Description Framework \(RDF\): Concepts and Abstract Syntax](#), G. Klyne, J. J. Carroll, Editors, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/> . [Latest version](#) available at <http://www.w3.org/TR/rdf-concepts/> .

RDF-MT

[RDF Semantics](#), P. Hayes, Editor, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/> . [Latest version](#) available at <http://www.w3.org/TR/rdf-mt/> .

RDFX

[RDF/XML Syntax Specification \(Revised\)](#), D. Beckett, Editor, W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/> . [Latest version](#) available at <http://www.w3.org/TR/rdf-syntax-grammar/> .

XMLBASE

[XML Base](#), J. Marsh, Editor, W3C Recommendation, 27 June 2001, <http://www.w3.org/TR/2001/REC-xmlbase-20010627/> . [Latest version](#) available at <http://www.w3.org/TR/xmlbase/> .

XHTML

[Modularization of XHTML™](#), S. Schnitzenbaumer, F. Boumphrey, T. Wugofski, S. McCarron, M. Altheim, S. Dooley, Editors, W3C Recommendation, 10 April 2001, <http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/> . [Latest version](#) available at <http://www.w3.org/TR/xhtml-modularization/> .

HTML4

[HTML 4.01 Specification](#), D. Raggett, A. Le Hors, I. Jacobs, Editors, W3C Recommendation, 24 December 1999, <http://www.w3.org/TR/1999/REC-html401-19991224/> . [Latest version](#) available at <http://www.w3.org/TR/html401/> .

XPATH

[XML Path Language \(XPath\) Version 1.0](#), J. Clark, S. J. DeRose, Editors, W3C Recommendation, 16 November 1999, <http://www.w3.org/TR/1999/REC-xpath-19991116/> . [Latest version](#) available at <http://www.w3.org/TR/xpath/> .

XSLT1

[XSL Transformations \(XSLT\) Version 1.0](#), J. Clark, Editor, W3C Recommendation, 16 November 1999, <http://www.w3.org/TR/1999/REC-xslt-19991116/> . [Latest version](#) available at <http://www.w3.org/TR/xslt/> .

Informative references

The following documents provide additional background but are not part of this specification.

primer

[GRDDL Primer](#), I. Davis, Editor, W3C Working Draft (work in progress), 2 October 2006, <http://www.w3.org/TR/2006/WD-grddl-primer-20061002/> . [Latest version](#) available at <http://www.w3.org/TR/grddl-primer/> .

usecases

[GRDDL Use Cases: Scenarios of extracting RDF data from XML documents](#), F. Gandon, Editor, W3C Working Group Note, 6 April 2007, <http://www.w3.org/TR/2007/NOTE-grddl-scenarios-20070406/> . [Latest version](#) available at <http://www.w3.org/TR/grddl-scenarios/> .

GRDDL-TESTS

[GRDDL Test Cases](#), C. Ogbuji, Editor, W3C Recommendation, 11 September 2007, <http://www.w3.org/TR/2007/REC-grddl-tests-20070911/> . [Latest version](#) available at <http://www.w3.org/TR/grddl-tests/> .

SPARQL

[SPARQL Query Language for RDF](#), E. Prud'hommeaux, A. Seaborne, Editors, W3C Working Draft (work in progress), 26 March 2007, <http://www.w3.org/TR/2007/WD-rdf-sparql-query-20070326/> . [Latest version](#) available at <http://www.w3.org/TR/rdf-sparql-query/> .

XSLT2

[XSL Transformations \(XSLT\) Version 2.0](#), M. Kay, Editor, W3C Recommendation, 23 January 2007, <http://www.w3.org/TR/2007/REC-xslt20-20070123/> . [Latest version](#) available at <http://www.w3.org/TR/xslt20/> .

RFC2731

J. Kunze [Encoding Dublin Core Metadata in HTML](#) in 1999

XFN

[XFN: Introduction and Examples](#) copyright GMPG 2003-2007. Eric, Tantek, and Matt

DCRDF

[Expressing Simple Dublin Core in RDF/XML](#) Beckett, Miller, Brickley 2002-07-31

P3P

[The Platform for Privacy Preferences 1.0 \(P3P1.0\) Specification](#), M. Marchiori, Editor, W3C Recommendation, 16 April 2002,

<http://www.w3.org/TR/2002/REC-P3P-20020416/> . [Latest version](#) available at <http://www.w3.org/TR/P3P/> .

STYPI

[Associating Style Sheets with XML documents](#) , J. Clark, Editor, W3C Recommendation, 29 June 1999, <http://www.w3.org/1999/06/REC-xml-stylesheet-19990629/> . [Latest version](#) available at <http://www.w3.org/TR/xml-stylesheet/> .

XPROC

[XProc: An XML Pipeline Language](#) , N. Walsh, Editor, W3C Working Draft (work in progress), 28 September 2006, <http://www.w3.org/TR/2006/WD-xproc-20060928/> . [Latest version](#) available at <http://www.w3.org/TR/xproc/> .

MF-RDF-FAQ

[Microformat FAQs for RDF Fans](#), last modified 17:57, 30 May 2006

Appendix: Transformations for Styling versus data extraction (Informative)

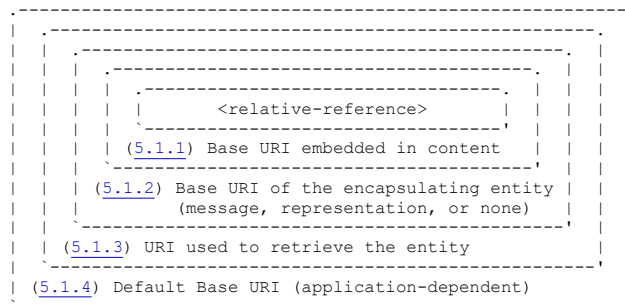
The xml-stylesheet processing instruction[STYPI] is generally deployed for automated presentation processing. This type of link is different from links to GRDDL transformation algorithms, which are intended to facilitate extracting data. Also, parsing the content of processing instructions is not supported by XML tools such as XSLT processors, and grounding processing instructions in URI space is not as straightforward as using namespaces with attributes.

Appendix: Base IRI considerations

In the [Adding GRDDL to well-formed XML](#) section, we have:

The base IRI for interpreting relative IRI references in a serialization of a graph produced by a GRDDL transformation is the base IRI of the source document.

This corresponds to RFC 3986, particularly [section 5.1](#), which illustrates the identification of a base URI, with the following picture:



During typical GRDDL processing, an intermediate RDF/XML serialization is produced as the output of a transform. To convert this serialization into an RDF graph, any relative references in the serialization are resolved to IRIs. To identify the appropriate base IRI for resolving a given relative reference, first check for a base URI embedded within this RDF/XML, following XML Base, as permitted by RDF Syntax. If there is no base URI embedded within this RDF/XML, then section 5.1.2 of RFC 3986 may apply, because the *encapsulating entity* of this serialization is the root element of the input document. If this element does not define a base URI, then its encapsulating entity, the input document, may define a base IRI.

The original document may be an XHTML family document, or it may be some other XML document.

The Base IRI of an XHTML Family document

For an XHTML family document, the base IRI of the input document may be specified as the value of the `href` attribute of the `<base>` element (if any). This is in accordance with section 5.1.1 of RFC 3986.

In many other cases, section 5.1.2 does not apply, and section 5.1.3 does apply. Section 5.1.3 specifies the use of the retrieval IRI as the base IRI. Furthermore, [section 5.1.3](#) of RFC 3986 specifies that:

if the retrieval was the result of a redirected request, the last URI used (i.e., the URI that resulted in the actual retrieval of the representation) is the base URI.

The resulting IRI is used as the base IRI parameter for processing the intermediate RDF/XML serialization.

The base IRI of other XML documents

Other XML documents may use XML Base. This is only recommended when the specific document format permits the use of XML Base.

When an `xml:base` attribute is present on the root element of an XML document, this specifies the base IRI for that document, following section 5.1.1 of RFC 3986.

When there is no `xml:base` attribute on the root element, even if there is such an attribute on a descendent element, then section 5.1.1 of RFC 3986 does not apply.

As in the XHTML case, we then have to consider sections 5.1.2, 5.1.3 and 5.1.4 of RFC 3986.

Of these, sections 5.1.3 is the most common case, and the note about redirected retrieval also applies.

The base IRI in a processing pipeline

A GRDDL aware agent computes GRDDL results when

given a URI *I* of an information resource *IR*, and an XPath node *N* for a representation of *IR*

To use a GRDDL aware agent in a processing pipeline, as well as the XPath node *N*, it is also necessary to specify a corresponding IRI *I*. This is used as the base IRI when the other mechanisms do not apply. This corresponds to section 5.1.4 of RFC 3986. It is even possible for the default IRI used to bear no relationship with the XPath node *N*, but in such a case, we [read](#):

As this definition is necessarily application-dependent, failing to define a base URI by using one of the other methods may result in the same content being interpreted differently by different types of applications.

A sender of a representation containing relative references is responsible for ensuring that a base URI for those references can be established.

Responsibilities for correct processing of base IRIs

Document authors, including profile and namespace documents

Document authors should, in general, include a base URI if the document is retrievable from some other URI.

For an XHTML family document[\[XHTML\]](#), this is done using the `base` element.

For other XML documents, if the format supports `xml:base` then this should be used. In general, experience suggests that there is least confusion when this is done on the root element. Document authors may also use `xml:base` attributes elsewhere in their documents, as permitted by the document format, with semantics as defined by XML Base[\[XMLBASE\]](#).

For XML documents in formats that do not support `xml:base`, and are not XHTML family documents, there is no support in GRDDL for specifying an in-line base URI.

When a profile or namespace document can be accessed via multiple URIs, for instance by a redirect, document authors should, in general, provide a GRDDL result that specifies profile transformations or namespace transformations for each of these URIs.

GRDDL aware agents

When a GRDDL result represented in RDF/XML using the [rule for RDF/XML](#), a base URI may be needed for this representation, in order to convert it into a RDF Graph, following the rules in the RDF/XML Syntax Specification[\[RDFX\]](#).

GRDDL results represented in other ways may also need a base URI.

Following the analysis above, a base URI for resolving a relative reference is defined by following section 5.1 of RFC 3986.

In many applications, it is highly undesirable that GRDDL results may depend on an application default URI, from section 5.1.4 of RFC 3986, ; some GRDDL aware agents may treat this possibility as an error.

GRDDL transformation authors

In general, when writing a GRDDL transformation for an XHTML family document to RDF/XML the best advice is to ignore issues to do with the base URI. The easiest approach is to produce relative URIs in the output, corresponding to any relative URIs in the input, and absolute URIs corresponding to any concepts built into the transform. Such relative URIs will be resolved, during the processing performed by a GRDDL aware agent, against the correct base URI.

When writing a GRDDL transformation for an XML document format that does not support `xml:base`, and has no means to represent an in-line base URI, there is little choice but to ignore issues of the correct base.

When writing a GRDDL transformation for an XML document format, other than an XHTML family document, that does not support `xml:base`, but has some other means to represent an in-line base URI, then a GRDDL aware agent will be ignorant of this means, and a well-written GRDDL transformation will attempt to correct for this. When a base URI is specified in such a way, one approach is to insert the base URI into the RDF/XML output as the value of an `xml:base` attribute, so that the RDF/XML parser will resolve relative URIs against that base, and ignore the base URI passed by the GRDDL aware agent, which will have been computed ignoring the conventions specific to this format.

When writing a GRDDL transformation for an XML document format that does support `xml:base`, then it must be remembered that a GRDDL aware agent has responsibility to handle an `xml:base` on the root element. If there is such an `xml:base` attribute, then the simplest behaviour for a GRDDL transformation, is to ignore it.

However, other `xml:base` attributes, not on the root element, are the responsibility of the transform, since the GRDDL aware agent ignores these. Thus, these lower level `xml:base` attributes should be honored, most easily by copying them into the output graph in the appropriate place. However, in general, `xml:base` attributes on ancestor nodes also have to be taken into account, unless there is an intervening `xml:base` attribute with an absolute URI as its value. This is clearly non-trivial to get right: to assist, the GRDDL library provides a module to be imported into your stylesheet, see below.

In all cases, while often unnecessary, if a transform is aware of an absolute base URI, specified in its input, for the whole document, it is never incorrect to use this base URI as the base URI for the output, for example, by adding an appropriate `xml:base` attribute to the `rdf:RDF` element.

Transforms that do this need to guard against the possibly incorrect similar treatment of relative base URIs. For example a `xml:base=".."` on the root element might, in the interaction between a correct GRDDL aware agent and a poorly written transform, be applied twice,

resulting in relative references being resolved at the wrong level in the directory hierarchy.

Acknowledgements and Change History

A companion [GRDDL design history and rationale](#) discusses this design in the context of HTML, PICS, and RDF since about 1997. The editor gratefully acknowledges the many contributions of community members in the development of GRDDL:

- In Dec 2000, Ann Navarro raised the [faq-html-compliance](#) issue: “The suggested way of including RDF meta data in HTML is not compliant with HTML 4.01 or XHTML”; in Apr 2001, Lee Jonas raised issue [rdfms-validating-embedded-rdf](#): “RDF embedded in XHTML and other XML documents is hard to validate”.
- In May 2003, Joseph Reagle convened a task force with a [Kickoff of public-rdf-in-xhtml-tf@w3.org](#) message. Dan Connolly sent a [relational data views of XHTML via XSLT](#) design sketch.
- In Nov 2003, [Dominique Hazaël-Massieux](#) wrote [An RDF-in-XHTML Proposal](#), a predecessor of this spec.
- In Jan 2004, Dan Connolly integrated that draft into this one and sent [a message calling for review](#). Discussion with Tim Berners-Lee led to generalizing from XHTML to all of XML and to indirection via namespace/profile document.
- In February 2004, the RDF Core specifications became W3C Recommendations; the issues [rdfms-validating-embedded-rdf](#) and [faq-html-compliance](#) were postponed.
- A [13 April 2004 snapshot](#) was published as a W3C Coordination Group Note to facilitate exchange between the Semantic Web Best Practices and Deployment Working Group and the HTML Working Group.
- Also in February 2004, Connolly presented to the TAG a [GRDDL design history and rationale](#) which discusses contribution of this design to Web Architecture issues such as [RDFinXHTML-35](#) and [namespaceDocument-8](#). Feedback from Norm Walsh has been valuable, and Noah Mendelsohn noted a connection to the *Cambridge Communiqué* in a [message of 22 March](#).
- Ben Adida started contributing use cases from Creative Commons in a [March 2004 meeting of the Semantic Web Best Practices & Deployment Working Group](#).
- A [16 May 2005 snapshot](#) was published as a W3C Team Submission by Dom and Dan.
- In a [March 2006 Semantic Web Interest Group meeting](#), Murray Maloney took an interest in the connection with XML Schemas and the readability of the specification, Brian McBride demonstrated some related implementation experience with transforming documents to RDF, and Ian Davis contributed the eRDF use case and profile.

The GRDDL Working Group convened August 2006 with Harry Halpin as chair and several of the contributors and implementors above participating, plus Chimezie Ogbuji, Fabien Gandon, Brian Suda, and Rachel Yager.

Jeremy Carroll provided detailed security considerations based on [RFC 2046](#) and implemented the HTTP header linking as proposed by Ian Davis.

The Working Group published a [24 October 2006 draft](#). The [issues list](#) shows the major design decisions since then.

The only changes since the 16 July 2007 release, outside the status section, are:

- a typo in the section [An example Dublin Core META transformation](#) and
- the addition of XQuery among the languages mentioned in section [6. GRDDL Transformations](#).