CURIE Syntax 1.0

A syntax for expressing Compact URIs

W3C Editor's Draft 22 January 2008

This version:
http://www.w3.org/MarkUp/2007/ED-curie-20080122
Latest version:
http://www.w3.org/TR/curie
Previous version:
http://www.w3.org/MarkUp/2008/ED-curie-20071205
Diff from previous version:
curie-diff.html
Editors:
Mark Birbeck, x-port.net Ltd.
Shane McCarron, Applied Testing and Technology, Inc.

This document is also available in these non-normative formats: PostScript version, PDF version.

The English version of this specification is the only normative version. Non-normative translations may also be available.

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

Abstract

The aim of this document is to outline a syntax for expressing URIs in a generic, abbreviated syntax. While it has been produced in conjunction with the XHTML 2 Working Group, it is not specifically targeted at use by XHTML Family Markup Languages. Note that the target audience for this document is Language designers, not the users of those Languages.

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 document is an updated working draft based upon comments received since the last draft. Originally this document was based upon work done in the definition of [XHTML2 [p.13]], and work done by the RDF-in-HTML task force RDF-in-HTML Task Force, a joint task force of the Semantic Web Best Practices and Deployment Working Group and XHTML 2 Working Group. It is not yet stable, but has had extensive review and some use in other W3C documents. It is being released in a separate, stand-alone specification in order to speed its adoption and facilitiate its use in various specifications.

This document has been produced by the W3C XHTML 2 Working Group as part of the HTML Activity. The goals of the XHTML 2 Working Group are discussed in the XHTML 2 Working Group charter.

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.

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

Please report errors in this specification to www-html-editor@w3.org (archive). It is inappropriate to send discussion email to this address. Public discussion may take place on www-html@w3.org (archive).

Table of Contents

1. Introduction 3
1.1. Existing Uses of CURIEs 4
1.1.1. Wikis 4
2. Conformance Requirements 5
2.1. User Agent Conformance 5
3. Syntax 7
4. Incorporating CURIEs into Markup Languages 9
4.1. SPARQL 9
4.2. HTML 4.01 9
4.3. XHTML 2 9
5. Usage 11
5.1. Examples 11
5.2. Ambiguities Between CURIEs and URIs 11
A. References 13

1. Introduction

This section is informative.

Ed. note: I have added more introductory text to better explain the problem we are trying to solve -Shane.

More and more languages need a mechanism to permit the use of *extensible* value collections. These are primarily found in XML attribute values, but also found in other, similar spaces in non-XML languages (e.g., [SPARQL [p.13]]). Typically such extension mechanisms utilize the concept of *scoping*, where values are created within a unique *scope*, and that value space is managed by whomever defines it. Using such a mechanism allows independent organizations to define values without the risk of collision.

At the same time, language designers are trying to ensure that their languages mesh smoothly into the *semantic web*. Since the basis of the *semantic web* is the notion that meaning can be derived through the relationship among resources, these extension mechanisms need a ready way of mapping their scoped values to resources (via URIs).

In many cases, language designers are attempting to use QNames for this extension mechanism [XML-SCHEMA-QNAME [p.13]]. QNames do permit independent management of the value space, and *can* map the values to a resource. Unfortunately, QNames are unsuitable in most cases because 1) they are NOT intended for use in attribute values, and 2) the syntax of QNames is overly-restrictive and does not allow all possible URIs to be expressed.

A specific example of the problem this causes comes from attempting to define the value space for books. In a QName, the part after the colon must be a valid element name, making an example such as the following *invalid*: `isbn:0321154991`

This is not a valid QName simply because ’0321154991’ is not a valid element name. Yet, in the example given, we don’t really want to define a valid element name anyway. The whole reason for using a QName was to reference an item in a private value space - that of ISBNs. Moreover, in this example, we want that value to map to a URI that will determine the meaning of that ISBN. As you can see, the definition of QNames and this (relatively common) use case are in conflict with one another.

This specification addresses the problem by creating a new data type whose purpose is specifically to allow for the definition of scoped values that map to URIs in exactly this way. This type is called a "CURIE" or a "Compact URI", and values that are valid QNames are a subset of this.

Note that this specification is targeted at markup language designers, not document authors. Any language designer considering the use of QNames in attribute values should consider instead using CURIEs, since CURIEs are designed for this purpose, while QNames are not.

1.1. Existing Uses of CURIEs

Although they are not currently called CURIEs, the technique described here is in widespread usage. However, taken literally, QNames would not support many of the examples that we would find 'in the wild' — the fact that they do is mainly because systems and authors take a very lax approach to QNames.

In other words, the *principle* used in QNames — that of combining a *namespace name* with a *local part* to generate a URI — is widely used, but little checking is done on the *local part* to ensure that the string is a valid element name. However, this does mean that CURIEs can be easily used in a number of places, since there is already a large amount of 'mind-share'. Current uses include:

1.1.1. Wikis

Many Wikis support a feature where a prefix like `isbn` can be substituted for something like:

```
http://www.amazon.com/?isbn=
```

or:

```
http://www.barnesandnoble.com/?q=
```

When a Wiki author wants to make use of this, they can simply enter:

```
Go and buy T. V. Raman's [[isbn:0321154991][book on XForms]].
```

and the Wiki software will automatically generate:

```
Go and buy T. V. Raman's <a href="http://www.amazon.com/?isbn=0321154991">book on XForms</a>
```

2. Conformance Requirements

This section is *normative*.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119 [p.13]].

2.1. User Agent Conformance

A conforming user agent must support all of the features required in this specification.

3. Syntax

This section is normative.

A CURIE is by definition a syntactic superset of a QName. It is comprised of two components, a *prefix* and a *reference*. The prefix is separated from the reference by a colon (`:`). It is possible to omit both the prefix and the colon, or to omit just the prefix and leave the colon. To disambiguate a CURIE when it appears in a context where a normal [URI [p.13]] may also be used, the entire CURIE is permitted to be enclosed in brackets (`[`, `]`).

```
safe_curie  :=  '[' curie ']'

curie       :=  [ [ prefix ] ':' ] reference

prefix      :=  NCName

reference   :=  irelative-ref (as defined in IRI)
```

When CURIES are used in an XML-based host language, prefix values MUST be able to be defined using the 'xmlns:' syntax specified in [XMLNAMES [p.13]]. Such host languages MAY also provide additional prefix mapping definition mechanisms.

When CURIES are used in a non-XML host language, the host language MUST provide a mechanism for defining the mapping from the `prefix` to an IRI.

A host language MAY provide a mechanism for defining a default prefix value. In such a host language, if the `prefix` is omitted from a CURIE, the default prefix value is used.

The concatenation of the prefix value associated with a CURIE and its `reference` MUST be an IRI [IRI [p.13]]. Note that while the set of IRIs represents the *lexical space* of a CURIE, the *value space* is the set of URIs (IRIs after canonicalization - see [IRI [p.13]]).

The CURIE prefix '_' is reserved. For this reason, the prefix '_' SHOULD be avoided by authors.

Host languages MAY define additional constraints on these syntax rules when CURIES are used in the context of those host languages. Host languages MUST NOT relax the constraints defined this specification.

The `safe_curie` production is for use in attribute values where it would be otherwise impossible to disambiguate between a CURIE and a URI.

Language designers SHOULD only use CURIEs (or safe_curies) as the datatype of new attributes in their markup language, since using them in values where historically an attribute has taken a URI as its datatype could break backward compatibility.

RTL languages and CURIE values

Should there be any advice given about using CURIEs in XML documents that are authored in an RTL language such as Arabic? Or does document order dictate what is the prefix and what is

the reference?

4. Incorporating CURIEs into Markup Languages

This section is informative.

Each host language that incorporates CURIEs supplies a mechanism for defining prefix mappings. In the case of XML-based host languages, one such mechanism is required to be `xmlns`. This section illustrates some possible alternative mapping mechanisms available in various existing languages.

NOTE: There are a number of different situations where QNames are currently used. There will be more illustrations in a future draft. Possible candidates include RDF/XML and RDF N3/Turtle.

4.1. SPARQL

The [SPARQL [p.13]] language provides a `PREFIX` keyword for defining the prefix used in their CURIE-like identifiers.

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?x ?name
WHERE  { ?x foaf:name ?name }
```

4.2. HTML 4.01

HTML 4.01 does not currently employ CURIEs. An extension to HTML 4.01 to support RDFa, however, has been discussed. Such an extension would need to define a prefix mapping mechanism in order to support the use of CURIEs in the RDFa attributes. For example:

```
<html>
    <head>
        <title>An HTML document using RDFa</title>
        <meta scheme="prefix" name="myPrefix" content="http://www.example.com/myPrefix/" >
    </head>
    <body>
        <p about="http://www.example.com/something" rel="myPrefix:reference">
            some content
        </p>
    </body>
</html>
```

4.3. XHTML 2

XHTML 2 incorporates RDFa. Since XHTML 2 is an XML-based markup language, documents annotated with RDFa could use the xmlns mechanism to define prefixes. However XHTML 2 also defines a special "prefix" value for the `property` attribute. So, in XHTML 2 the following would work:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:dc="http://purl.org/dc/elements/1.1/">
    <head>
        <title>An HTML document using RDFa</title>
        <link property="prefix" content="myPrefix" href="http://www.example.com/myPrefix/" >
    </head>
    <body>
        <p about="http://www.example.com/something" rel="myPrefix:reference">
            some content was written by <span property="dc:creator">some author</span>
        </p>
    </body>
</html>
```

5. Usage

This section is informative.

CURIEs can be used in exactly the same way that QNames have been used in attribute values, with the modification that the format of the strings after the colon are looser. In all cases a parsed CURIE will produce an IRI. However, the process of parsing involves substituting the value represented by the prefix for the prefix itself, and then simply appending the part after the colon.

Note that if CURIEs are to be used in the context of scripting, accessing a CURIE via standard mechanisms such as the XML DOM will return the raw CURIE, not its lexical value. In order to develop portable applications that evaluate CURIEs, a script author must transform CURIEs into their lexical value before evaluating them (e.g., comparing two CURIEs).

5.1. Examples

All of the following are valid CURIEs — even though they are not valid QNames — and they take advantage of the fact that the part after the colon no longer needs to conform to the rules for element names:

```
home:#start
joseki:
google:xforms+or+'xml+forms'
```

5.2. Ambiguities Between CURIEs and URIs

There will be situations in the design of a language where it is desirable for an attribute that can take a URI to also be able to contain a CURIE. For example, in XHTML the `href` attribute allows a URI to be specified that will be navigated on user action, but it would also be useful to be able to abbreviate this URI, using the compact syntax. However, the problem is that it is not possible for the language parser to be completely sure whether it has located a CURIE or a URI. For example, a link to an email address can be expressed like this:

```
<span rel="foaf:homePage" resource="http://www.example.org/home.html">home</span>
```

There is no way to be sure that this is a normal URI, or a CURIE. Therefore the syntax for carrying a CURIE when there is any possibility of ambiguity is to enclose the CURIE in square brackets, as in the following example:

```
<html xmlns:wp="http://en.wikipedia.org/wiki/">
    <head>...</head>
    <body>
        <p>
            Find out more about <span resource="[wp:Thales]">Thales</span>.
        </p>
    </body>
</html>
```

Note:

Not only does this abbreviate the URI, but it also makes it possible to change a whole group of URIs to point to some other source, simply by changing the prefix definition. For example, consider the following mark-up:

```
<html xmlns:wp="http://en.wikipedia.org/wiki/">
    <head>...</head>
    <body>
        <p>
            Thales had a profound influence on other Greek thinkers and
            therefore on Western history.  Some believe
            <span resource="[wp:Anaximander]">Anaximander</span>
            was a pupil of Thales. Early sources report that one of
            Anaximander's more famous pupils,
            <span resource="[wp:Pythagoras]">Pythagoras</span>, visited
            Thales as a young man, and that Thales advised him to travel
            to Egypt to further his philosophical and mathematical studies.
        </p>
    </body>
</html>
```

Given that all references to Wikipedia entries in this example are based on the prefix defined in `xmlns:wp`, then simply changing this prefix changes the base for all Wikipedia references within the document. It is not difficult to see how, by extending this principle a user can begin to get control of their own browsing experience. For example, a document might contain a reference to a company, with links to news about the company, financial information and details on key directors. By using CURIEs to express those links it is possible to use different sources for the information, even to the extent that they could be overridden the user:

```
<html xmlns:finance="..."
      xmlns:news="..."
      xmlns:people="...">
    <head>...</head>
    <body>
        <p>We hear from people in the know that the great thinker
        Bullwinkle is being recruited by <b>Google</b>
        (nasdaq: <span resource="[finance:GOOG]" class="maintkrlink">GOOG</span>
        - <span resource="[news:GOOG]">news</span>
        - <span resource="[people:GOOG]">people</span>)
        was an "unconfirmed rumor", but that the search engine behemoth is
        indeed keen to expand its cartoon presence.</p>
    </body>
</html>
```

A. References

This appendix is informative.

IRI
"*Internationalized Resource Identifiers (IRI)*", RFC 3987, M.Duerst, M. Suignard January 2005.
Available at: http://www.ietf.org/rfc/rfc3987.txt

RFC2119
"*Key words for use in RFCs to indicate requirement levels*", RFC 2119, S. Bradner, March 1997.
Available at: http://www.ietf.org/rfc/rfc2119.txt

SPARQL Query Language for RDF
SPARQL Query Language for RDF (See http://www.w3.org/TR/2007/PR-rdf-sparql-query-20071112/.)

URI
"*Uniform Resource Identifiers (URI): Generic Syntax*", RFC 3986, T. Berners-Lee *et al.*, January 2005.
Available at: http://www.rfc-editor.org/rfc/rfc3986.txt. This RFC updates RFC 1738 [URI [p.13]] and obsoletes RFC 2732, 2396 and 1808.

XHTML2
"*XHTML™ 2.0*". J. Axelsson *et al.*, 26 July 2006.
Available at: http://www.w3.org/TR/2006/WD-xhtml2-20060726
The latest version is available at: http://www.w3.org/TR/xhtml2

XML
"*Extensible Markup Language (XML) 1.0 (Third Edition)*", W3C Recommendation, T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, *eds.*, 2 February 2004.
Available at: http://www.w3.org/TR/2004/REC-xml-20040204

XMLNAMES
"*Namespaces in XML*", W3C Recommendation, T. Bray, D. Hollander, A. Layman, *eds.*, 14 January 1999.
Available at: http://www.w3.org/TR/1999/REC-xml-names-19990114

XML-SCHEMA-QNAME
XML Schema Part 2: Datatypes Second Edition: Section 3.2.18 QName (See http://www.w3.org/TR/xmlschema-2/#QName.)