

Creating HTML Meta-tags Using the Dublin Core Element Set

Christopher Sean Cordes
Assistant Professor
Instructional Technology Librarian
Parks Library
Iowa State University

Abstract

The breadth and scope of information available online and through intranets is making the standardization and use of metadata to identify records imperative if not mandated. Some measure of searchability is provided by the html meta tag<meta>. But the tag has limited potential for describing complex documents. The Dublin Core Initiative group has developed metadata standards that support a broad range of purposes and business models," most notably the Dublin Core Element Set. The set is fairly easy to learn but must be combined with HTML if to provide metadata for referencing by a web crawler or search engine. This paper will outline the steps needed to create HTML metadata using the Dublin Core Element Set, and some of the benefits of doing so.

Keywords: Dublin Core Element Set, HTML, Metadata, Meta tag, Search engine

Creating HTML Meta-tags with the Dublin Core Element Set

Overview

The breadth and scope of information available online and through intranets is making the standardization and use of metadata to identify records imperative if not mandated. There is a small degree of metadata control built in to the HTML language in the meta <meta> element. However this tag has limited potential for describing complex documents. The Dublin Core Initiative a group working towards "the development of interoperable online metadata standards that support a broad range of purposes and business models" developed an element set for building flexible control records for

online documents. Yet while the set is fairly easy to learn and use to create a text based record, the DC elements must be combined with HTML if they are to provide metadata for referencing by a web crawler or search engine. This paper will outline the steps needed to create HTML metadata using the Dublin Core element set.

What HTML offers.

Some control over document metadata is built into the HTML language itself. The title tag <title> is often used a point of index by web spiders. In addition, some engines reference certain body elements, such as header tags like <h1>. Finally, there is an element set designed for identifying information about a document. The tags can be used to represent a number of elements including the author, page description, keywords and the date of creation (Sullivan). Two of the most well known uses are the meta-description tag and the meta keywords tag. These have 2 main functions. The meta description tag delivers a description of your site that is displayed when your site is viewed as the result of a search (figure1).

Figure 1

```
<meta name="description" content="This tutorial explains how to use HTML meta tags with the Dublin Core element Set.">
```

The meta keywords tag provides additional terms for the web crawler to index when passing through your site (figure 2). This method is now somewhat deprecated as most engines currently ignore this tag. In addition, for engines that do support the keyword function, the document must also contain the keyword within the body text if the meta-data is to be of any

benefit at all. However, you can get similar results by simply using the keywords within the page body alone, and for this reason as well, the use of the meta-keyword is waning.

Figure 2

```
<meta name="keywords" content="HTML, Dublin Core, Metadata, Search Engines, Web Crawler, Tutorials, Information Storage and Retrieval">
```

Despite this trend the meta tag is still commonly used for other purposes, such as to show ownership of an online work. For example, the meta data in figure 3 tells the world that I am the creator of the page.

Figure 3

```
<meta name="author" content="Sean Cordes">
```

The Gist of Dublin Core

The Dublin core element set is a "pidgin" metadata language that includes 15 elements (nouns). These elements are repeatable and can be clarified or enhanced through the application of qualifiers (adjectives) to give a consistent yet flexible description of objects (Figure 4).

Figure 4.

Dublin Core Element Set	
Element Name:	Title
Label:	Title
Definition:	A name given to the resource.
Comment:	Typically, Title will be a name by which the resource is formally known.
Element Name:	Creator
Label:	Creator
Definition:	An entity primarily responsible for making the content of the resource.

Element Name: Subject

Label: Subject and Keywords
Definition: A topic of the content of the resource.

Element Name: Description

Label: Description
Definition: An account of the content of the resource.

Element Name: Publisher

Label: Publisher
Definition: An entity responsible for making the resource available

Element Name: Contributor

Label: Contributor
Definition: An entity responsible for making contributions to the content of the resource.

Element Name: Date

Label: Date
Definition: A date of an event in the lifecycle of the resource.

Element Name: Type

Label: Resource Type
Definition: The nature or genre of the content of the resource.

Element Name: Format

Label: Format
Definition: The physical or digital manifestation of the resource.

Element Name: Identifier

Label: Resource Identifier
Definition: An unambiguous reference to the resource within a given context.

Element Name: Source

Label: Source
Definition: A Reference to a resource from which the present resource is derived.

Element Name: Language

Label: Language
Definition: A language of the intellectual content of the resource.

Element Name: Relation

Label: Relation
Definition: A reference to a related resource.

Element Name: Coverage

Label: Coverage
Definition: The extent or scope of the content of the resource.

Element Name: Rights

Label: Rights Management
Definition: Information about rights held in and over the resource.

The Dublin Core set can be applied to any type of object, but favors text documents. And though it is simple to use and provides a fairly thorough description, it is not well suited for expressing complex interactions or relationships (Hillman). To create a simple record, you just apply the elements to describe the object in question, just as you would with a traditional MARC record. For example, a simple record for this tutorial might look like this.

Title= “Creating HTML metadata using Dublin Core: A Tutorial”

Creator= “Sean Cordes”

Subject = “HTML Metadata”

Subject = “Dublin Core”

Type = “Text”

A simple enough task, but really powerful in a number of ways. First, note that my name is in forward order. I could have legitimately said “Cordes, Sean” as well. The point is that Dublin core can be very useful when dealing with fields containing alternate structures, such as names. Using Dublin Core, names of any language can be documented with ease. Another feature of Dublin Core shown here is the ability to repeat fields. In this case I have assigned 2 subject areas to the text that I think users might find relevant. At this point you may be wondering what the big deal is, after all, there are ways to address language issues, and subject notes in the MARC format. If we look at how DC can be applied to HTML, the relevance (and power) of Dublin Core become more clear.

Blending the two languages.

The html meta tag provides a logical container for blending Dublin Core elements into the html document. First, you apply the DC element to the meta "name" descriptor.

```
<meta name = "DC.Title"
```

Then apply the value of the DC element to the content descriptor of the meta tag like this:

```
content = "Creating HTML metadata using Dublin Core: A Tutorial">
```

If I continue the process we end up with these element tags:

```
<meta name = "DC.Title"
  content = "Creating HTML metadata using Dublin Core: A Tutorial">
<meta name = "DC.Creator"
  content = "Sean Cordes">
<meta name = "DC.Subject"
  content = "HTML Metadata">
<meta name = "DC.Subject"
  content = "Dublin Core">
<meta name = "DC.Type"
  content = "text">
```

Finally, we add a link to reference the definitions comprised by the element schema.

```
<link rel = "schema.DC"
  href = "http://purl.org/DC/elements/1.0/">
```

The final meta tag header result looks like this:

```
<head>
<title = Creating HTML metadata using Dublin Core: A Tutorial>
<meta name = "DC.Title"
  content = "Creating HTML metadata using Dublin Core: A Tutorial">
<meta name = "DC.Creator"
  content = "Sean Cordes">
<meta name = "DC.Subject"
  content = "HTML Metadata">
<meta name = "DC.Subject"
  content = "Dublin Core">
```

```
<meta name = "DC.Type"  
  content = "text">  
<link rel = "schema.DC"  
  href = "http://purl.org/DC/elements/1.0/">  
</head>
```

Summary

By creating the Dublin core meta tags, we provided a way for the document to be index using several distinct fields. Since the document can be searched by any of these fields, the document can be located more specifically than the “generic” fields provided by html. So people can more easily search for our static web page. Our Dublin Core tags also give our pages an eye toward to the future. By converting our DC information into a variable, we can store information about our page in a database, share it with other databases with documents using the DC scheme, and update it on the fly when we like by running a database query that writes new information to the stored tags. These are advanced ideas, but they all begin here. I hope this brief guide will help set aside any fears about DC html, and give some incentive (and hope) to those seeking a simple yet powerful meta-data solution.

Areas for further investigation

Once you feel comfortable using unqualified Dublin Core elements, you may want to increase the comprehensiveness of your tags using the Dublin Core qualifiers. In addition, you might want to check out other schema useful for creating metadata, such as the MARC XML format.

References

Hillmann, Diane. Using Dublin Core, 2001.

<http://dublincore.org/documents/usageguide/>

Sullivan, Danny. How to use HTML Metatags , 2002.

<http://www.searchenginewatch.com/webmasters/article.php/>