

# **Dublin Core Application Profile Guidelines**

## **Draft CWA**

Contributor: CEN/ISSS MMI-DC Project Team on Metadata – Dublin Core - Application Profiles

Contributor: Thomas Baker

Contributor: Makx Dekkers

Contributor: Thomas Fischer

Contributor: Rachel Heery

Modified: 2003-06-03

Description: These guidelines specify the structure and content of Dublin Core Application Profiles, a form for documenting which terms a given application uses in its metadata, with what extensions or adaptations, and specifying how those terms relate both to formal standards such as Dublin Core as well as to less formally defined element sets and vocabularies. This draft will be submitted to the Metadata for Multimedia Information Dublin Core Workshop of the European Committee for Standardization (CEN) in Brussels for approval as a CEN Workshop Agreement in September 2003.



---

# Contents

Contents .....	2
Foreword .....	3
Introduction .....	4
1 Scope .....	5
2 Definitions .....	6
3 Identifying terms with appropriate precision .....	7
4 Attributes of Term Usages .....	8
4.1 Identifying attributes .....	8
4.2 Definitional attributes .....	8
4.3 Relational attributes .....	8
4.4 Constraints .....	8
5 Examples .....	10
5.2 Example 1 .....	10
6 Discussion .....	11
6.2 Documentational styles of DCAPs .....	11
6.2 Descriptive metadata about DCAPs .....	11
6.3 "Local" and "Source" attributes .....	11
6.4 Types of Comments .....	11
6.5 "Using" Encoding Schemes .....	12
6.6 "Using" Element Refinements .....	12
6.7 Declaring new elements .....	12
6.8 Documenting grouped or nested metadata elements .....	12
6.9 Documenting unorthodox practices .....	13
<b>Annex A: Metadata describing a DCAP .....</b>	<b>14</b>
<b>Annex B: Options for machine-interpretable DCAPs .....</b>	<b>15</b>
<b>Bibliography .....</b>	<b>16</b>

---

## Foreword

Normally the Foreword is drafted by the CEN/ISSS secretariat.

## Introduction

A Dublin Core Application Profile (DCAP) is a declaration specifying which metadata terms an organization, information provider, or user community uses in its metadata. By definition, a DCAP identifies the source of metadata terms used – whether they have been defined in formally maintained standards such as Dublin Core, in less formally defined element sets and vocabularies, or by the creator of the DCAP itself for local use in an application. Optionally, a DCAP may provide additional documentation on how the terms are constrained, encoded, or interpreted for application-specific purposes.

A DCAP is designed to promote interoperability within the constraints of the Dublin Core model and to encourage harmonization of usage and convergence on "emerging semantics" around its edges. Historically, application profiles have emerged out of a need to share local domain- or application-specific refinements of or extensions to Dublin Core within particular application communities without necessarily seeking an extension of the core standard maintained by the Dublin Core Metadata Initiative (DCMI). Application profiles document how implementers use elements from Dublin Core along with elements from other vocabularies, customizing standard definitions and usage guidelines for local requirements [HEERY].

In practice, application profiles are created for a wide range of purposes: to document the semantic structure and constraints used for a set of metadata records ("instance metadata"); to help communities of implementers harmonize metadata practice among themselves; to identify emerging semantics as possible candidates for formal standardization; as guides for semantic crosswalks and format conversions; as specifications for formal encoding structures such as Document Type Definitions (DTDs); for interpreting or presenting legacy or proprietary metadata in terms of widely-understood standards; or for documenting the rules and criteria according to which a set of metadata was created. Application profiles often represent "work in progress", providing foci for ongoing efforts to incrementally improve and clarify a body of shared metadata semantics within a particular user community.

In the absence of guidelines, creators of application profiles have hitherto invented a wide range of presentation formats. The present document distills the salient features of many existing profiles into a format that is as concise and simple as possible, yet as precise and detailed as is sometimes necessary to support the various uses identified above.

Semantic interoperability – the ultimate purpose of documents such as DCAPs – is a longer-term goal to be pursued as metadata vocabularies and related enabling technologies mature over time. In their current form, DCAPs are designed to document metadata usage in a normalized form that will lend itself to translation into common models, such as RDF, that can be processed by machines to automate such interoperability.

Machine-understandable representations will achieve this goal to the extent that metadata terms can be referenced using stable, well-documented identifiers. As discussed below, the practice of identifying metadata terms with Uniform Resource Identifiers (URIs) is currently gaining momentum. Maintaining a DCAP over time, then, may involve improving its precision incrementally by identifying its terms with URIs as the URIs become available.

In the meantime, these guidelines aim at the more modest aim of providing system developers and information specialists with a normalized and readable view of Dublin-Core-based metadata models. Creators of DCAPs should bear in mind that this normalized form of documentation cannot itself address the deeper problems of interoperability in a world with a diversity of underlying metadata models – problems which will continue to challenge the metadata community as a whole, and the Dublin Core Metadata Initiative in particular, for the foreseeable future.

---

# 1 Scope

The present document gives guidance on how information should be structured and presented in Dublin Core Application Profiles. Principles and concepts underlying DCAPs as declarative metadata constructs are defined and explained.

The guidelines do not mandate a particular document format for DCAPs. DCAPs may be presented as plain text files or as Web pages, word-processing files, PowerPoint, or indeed as ink on paper. By providing a consistent presentation structure for such documents, however, these guidelines aim at making it easier for people to understand what others are doing in their metadata. Moreover, the guidelines mandate enough structure to ensure that DCAPs will be convertible as straightforwardly as possible into expressions that use schema languages, such as RDF, for automatic processing by machines. In this sense, a normalized form for DCAPs is a first step towards the more ambitious and long-term goal of automating semantic interoperability across a diversity of information sources.

---

## 2 Definitions

**Dublin Core Application Profile (DCAP):** A DCAP is a declaration specifying which metadata terms an organization, information provider, or user community uses in its metadata and how those terms have been customized or adapted to a particular application. By definition, a DCAP is based in part on Dublin Core and follows DCMI Grammatical Principles [DCMI-PRINCIPLES]. A DCAP consists of one or more Term Usages.

**Term Usage:** A Term Usage is a description of a metadata term, which, at a minimum, identifies a metadata term with "appropriate precision" by using one or more identifying attributes – "Term URI", "Defined By", "Name", "Label" – as described in Section 3. Optionally, a Term Usage may also describe or annotate a term in more detail by providing additional definitional attributes, relational attributes, or constraints, as described in Section 4.

**DCMI Grammatical Principles:** As maintained by the Dublin Core Metadata Initiative, DCMI grammatical principles specify a typology of metadata terms – Elements, Element Refinements, Encoding Schemes, and Vocabulary Terms – along with their interrelationships and functions [DCMI-PRINCIPLES]. A DCAP is based on the simple model of a resource described with a flat set of properties. This is consistent with DCMI grammatical principles, which do not themselves specify more elaborate models.

### 3 Identifying terms with appropriate precision

Application profiles serve to clarify who is declaring and maintaining the metadata semantics that a group wants to share. This section describes how a metadata term used in a Term Usage can be identified with "appropriate precision".

At present, the preferred method for identifying a metadata term is to cite its Uniform Resource Identifier (URI) if such is available. A URI is "a compact string of characters for identifying an abstract or physical resource" constructed according to a generic and flexible syntax [URI]. The World Wide Web Consortium has promoted the notion that "All important resources should be identified by a URI" [WEBARCH] and has specifically promoted the use of URIs for identifying metadata elements. In the CORES Resolution of December 2002, the maintainers of seven leading metadata standards – Dublin Core, IEEE/LOM, DOI, CERIF, MARC21, ONIX, and GILS -- pledged to assign URIs to their elements and to articulate policies for the persistence of those URIs [CORES-RESOLUTION].

For metadata terms to which URIs have been officially assigned – for example, by signatories of the CORES Resolution such as DCMI – that URI should be cited in the field "Term URI". For example, the Dublin Core element "Audience" should be cited as "<http://purl.org/dc/terms/audience>". As this form of identification is precise and sufficient on its own, other identifying fields may be left blank:

Term URI	<a href="http://purl.org/dc/terms/audience">http://purl.org/dc/terms/audience</a>
Name	-
Label	-
Defined By	-

A term that has been declared or documented somewhere but not assigned a URI (as far as one knows) should be identified as precisely as possible by providing its name and pointing to a declarative document or schema in which it has been defined. The declarative document or schema should be cited with URI, Web address, or bibliographic reference in the field "Defined By". The term itself can be cited using either a string identifier or token (in the field "Name") or a natural-language label (in the field "Label"), or both, taken from the declarative document or schema:

Term URI	-
Name	AttendancePattern
Label	Attendance Pattern
Defined By	<a href="http://myproject.org/profile/">http://myproject.org/profile/</a>

For a term that has not already been defined in any other declarative document, the field "Defined By" should simply cite the URI of the DCAP itself (as assigned in with "Identifier" in the DCAP Descriptive Header). For example, in a DCAP with the URI "<http://myproject.org/profile/>", a new local term called "Star Ratings" could be defined as follows:

Term URI	-
Name	StarRatings
Label	Star Ratings
Defined By	<a href="http://myproject.org/profile/">http://myproject.org/profile/</a>

A creator of a DCAP wishing to declare locally coined terms in a way that makes them citable with precision, and thus re-usable by others, may undertake the additional step of assigning them URIs. At present, the technical conventions and "Web etiquette" for naming metadata terms with URIs have yet to establish themselves in common practice, though at a minimum it seems both polite and sensible not to promote new URIs unless it is expected they will be maintained. For the purposes of DCAPs, DCMI itself provides models of practice, and further options are likely to emerge as the CORES Resolution is implemented [NAMESPACE, DCMI-TERMS, DCMI-SCHEMAS].



## 4 Attributes of Term Usages

Attributes for describing the metadata terms "used" in a DCAP are listed below. Note that they are called "attributes" here simply to avoid confusingly recursive formulations such as "terms for describing terms".

In general, the principle of "appropriate precision" applies to the use of Identifying Attributes for Term Usages. At a minimum, a Term Usage should use one or more of the four Identifying Attributes to identify the term being used as precisely as appropriate – i.e., with a formally assigned URI if available, or otherwise by citing a name or label for the term along with a reference to a document, schema, or Web page in which that term is defined.

All of the other attributes of Term Usages are optional and should be used as local needs may dictate. As discussed in Section 5.3, these attributes may record "local" or "source" usage, as needed.

### 4.1 Identifying attributes

Term URI	A Uniform Resource Identifier used to identify the term.
Name	A unique token assigned to the term.
Label	A human-readable label assigned to the term.
Defined By	An identifier of a namespace, pointer to a schema, or bibliographic reference for a document within which the term is defined.

### 4.2 Definitional attributes

Definition	A statement that represents the concept and essential nature of the term.
Comments	Additional information about the term or its application.
Type of term	A grammatical category of the term (e.g., "Element", "Element Refinement", or "Encoding Scheme").

### 4.3 Relational attributes

Refines	The described term semantically refines the referenced term.
Refined By	The described term is semantically refined by the referenced term.
Encoding Scheme For	The described term, an Encoding Scheme, qualifies the referenced term.
Has Encoding Scheme	The described term is qualified by the referenced Encoding Scheme.
Similar To	The described term has a meaning the same as, or similar to, that of the referenced term.

### 4.4 Constraints

Obligation	Indicates whether the element is required to always or sometimes be present (i.e., contain a value). (Examples include "Mandatory", "Conditional", and "Optional".)
------------	---

Condition	Describes the condition or conditions according to which a value shall be present.
Datatype	Indicates the type of data that can be represented in the value of the element.
Occurrence	Indicates any limit to the repeatability of the element.

---

## 5 Examples

### 5.1

---

## 6 Discussion

### 6.1 Documentational styles of DCAPs

The use of different document formats or even formal schema languages for presenting DCAPs will impose a wide range of constraints on presentation. As examples of good presentation practice, several existing application profiles have been reformatted as Web pages in accordance with these guidelines. In general, each Term Usage should be depicted as a table with attributes on the left and attribute values on the right.

DCAPs intended for use by software developers will need to be explicit and detailed, but DCAPs intended primarily as informational documents for human consumption can (and often should) be much terser. Attributes not used can be omitted. URIs used for identifying metadata terms can be substituted by qualified names – for example, “dc:title” (entered in the “Name” field) instead of “http://purl.org/dc/elements/1.1/title” (entered in the “Term URI” field).

### 6.2 Descriptive metadata about DCAPs

When presented in a document or Web page, a DCAP should be described in a Dublin-Core-based header specifying, at a minimum, a Title, Contributor, Date, Identifier, and Description, as explained in more detail in Annex A.

### 6.3 “Local” and “Source” attributes

Ideally, application profiles would be dynamically up-dated with information on the terms they use directly from schemas on the Web and this information would be integrated with local annotations into a “one-stop” document for the convenience of users. The use of machine-understandable DCAPs may some day make this possible.

In the meantime, however, creators of DCAPs who wish to include definitions or other such information from original source documents in their Term Usages have no choice but to copy that information from the source. If not maintained, of course, such redundant documentation can go out of alignment with the official source.

If such redundant “source” documentation is supplied, it should ideally be labelled or otherwise visually distinguished from locally added information or annotations. As these guidelines do not specify a particular computer-file format for DCAPs, they cannot recommend a method for tagging these fields for machine recognition. Rather, a DCAP should establish its own document-internal convention, such as labelling certain fields as “Source Label” or “Source Definition” or presenting their contents in italics wherever it seems appropriate to specify such distinctions with precision.

### 6.4 Types of Comments

Past creators of application profiles for Dublin Core have invented many types of annotation, the most popular of which have been “Notes”, “Best Practice”, “Usage”, “Scope”, “Open Questions”, “Examples”, “Purpose”, “Guidelines”, and “Don’t Confuse With”. While the present guidelines lump all of the above into a generically named “Comments” field, creators of DCAPs may wish to repeat this field with different labels as needed. The needs of future machine processing do not now seem to dictate tighter uniformity in this area.

## 6.5 “Using” Encoding Schemes

In the Dublin Core model, Encoding Schemes are terms that provide an interpretive context for the values of an Element or Element Refinement. An Encoding Scheme names either a controlled vocabulary from which a value is taken or a specification for parsing a value string. (A Vocabulary Term is a member of a controlled vocabulary of values that is named by an Encoding Scheme.)

In general, it is not the role of application profiles to "declare" controlled vocabularies of values, either in the sense of creating lists of potential values or in the sense of giving that list (as a whole) a name and URI. As discussed above for metadata term declarations, doing so is the function of declarative documents or schemas.

Typically, creators of DCAPs need only to identify the encoding schemes that can be used with a particular element. In this case, a Term Usage may use the field "Has Encoding Scheme", either by citing an official URI or by providing enough information to identify a controlled vocabulary or syntax specifications.

Some creators of DCAPs may simply want to make shorthand statements along the lines that metadata "can use encoding schemes defined in Standard A, Standard B, and Standard C" or that "all values in Vocabulary D can be used as element refinements for Contributor". For now, such blanket statements should simply be recorded in "Comment" fields.

If the creator of a DCAP merely wishes to specify a few possible values (e.g., "Animal, Vegetable, or Mineral"), these can be simply listed in a "Comment" field.

Occasionally, creators of DCAPs may want to cite an Encoding Scheme in a stand-alone Term Usage in order to add annotations or comments. This is acceptable, though perhaps less readable than the style of listing Encoding Schemes in the Term Usage of an element. The field "Encoding Scheme For" can be used to point to the Element or Element Refinement qualified. Terms referenced with these attributes should be identified with appropriate precision, preferably with a URI if available.

## 6.6 “Using” Element Refinements

As acknowledged in the DCMI Grammatical Principles [DCMI-PRINCIPLES], element refinements are sometimes used in application environments together with the elements they refine (in the manner of adjectives) and sometimes stand-alone (like noun phrases). Whether one follows one or the other style may determine whether the DCAP documents Element Refinements under the Term Usages of Elements or as stand-alone Term Usages on their own. The attributes "Refines" and "Refined By" give creators of DCAPs the flexibility to do this either way. Terms referenced with these attributes should be identified with appropriate precision, preferably with a URI if available.

## 6.7 Declaring new elements

There is nothing to restrain the creator of a DCAP from creating new URIs as identifiers for locally coined metadata terms. For reasons discussed above in Section 3, one should perhaps pause for reflection before taking this step, and if URIs are declared, this step should perhaps be documented separately and not embedded "in passing" into a DCAP full of Term Usages. Any URIs declared for use in a DCAP might best be formed by following the DCMI algorithm and concatenating the URL of the DCAP (e.g., "http://myproject.org/profile/") and the Name of the term (e.g., "starRatings") into a single string (e.g., "http://myproject.org/profile/starRatings") [DCMI-NAMESPACE].

## 6.8 Documenting grouped or nested metadata elements

In order to be usable across a diversity of application environments, Dublin Core was designed as a flat set of attributes for describing a resource. In implementation practice, however, Dublin Core elements may be embedded in more elaborate models that group or nest the elements in locally specific ways.

In the absence of a clear and widely accepted data model beyond that of the flat set of attributes, however, applications for integrating metadata from many different sources may be able only to extract and interpret the metadata in terms of Simple Dublin Core, losing any application-specific modelling context. An

application designer wishing to document nesting or grouping constructs in a DCAP will need to extend the simple model described here in order to do so and should bear in mind that documenting such constructs will not in itself guarantee that they will be understood or correctly processed by other applications.

## 6.9 Documenting unorthodox practices

For reasons both of history and of expedience, a significant number of applications have metadata based on interpretations of the Dublin Core model that are unsound from the standpoint of today's grammatical principles. For example, an application may use "CreatorDateOfBirth" – an element representing the birth date of a creator of a resource that does not semantically "refine" dc:creator as its name may imply.

Rather than incorrectly asserting "CreatorDateOfBirth" to be an Element Refinement refining <http://purl.org/dc/elements/1.1/creator>, the Term Usage in the DCAP should simply record the local name of the element and identify the URI of the DCAP itself as its source. For example, if the DCAP itself is identified by "<http://myproject.org/profile/2003/03/17/>", the Term Usage should declare the following, leaving empty any fields (such as "Term URI" and "Refines") that would make incorrect assertions about the element:

Term URI	-
Local Name	CreatorDateOfBirth
Defined By	<a href="http://project.org/profile/2003/03/17/">http://project.org/profile/2003/03/17/</a>
Refines	-

Whether "errors" such as "CreatorDateOfBirth" will be of negative consequence for interoperability will depend on how they are interpreted and used in the context of particular applications. The analytical effort involved in creating a DCAP is in effect an important first step towards putting such applications onto a more interoperable foundation.

---

## Annex A: Metadata describing a DCAP

A DCAP should itself be described with Dublin Core metadata, either in a header or in a separate metadata record. At a minimum, this description should include:

Title	A name for the Application Profile.
Contributor	A creator or maintainer of the Profile.
Date	The date of last modification.
Identifier	An unambiguous reference to the Profile. Best practice is to provide a URL by which a copy of the document or schema can be retrieved over the Web.
Description	A concise description of the Profile. As appropriate, the description should elaborate on the context and purposes in which the DCAP is intended to be used; the organizations or individuals involved in its development; any arrangements, policies, or intentions regarding the future development and maintenance of the DCAP; or technical characteristics of the instance metadata or database described.

---

## Annex B: Options for machine-interpretable DCAPs

DCAPs can be expressed in machine-interpretable schema languages, and such machine-interpretable schemas can be manipulated by software applications. This CWA does not give detailed recommendations on how such schemas should be structured as a number of issues are still open for debate. The scope of this CWA is limited to recommending how application profiles can be expressed as text documents. Future options for machine-interpretable DCAPs are outlined below.

Currently, two schema languages specified by W3C might be considered: XML Schema [XML-SCHEMA] and RDF Schema [RDF-SCHEMA]. The choice of schema language will be influenced by the functionality that the schema is intended to support – for example, whether it is required as a predictable format for data exchange or intended to support inferences about existing metadata. Such different objectives imply different choices between the two schema languages. There has been some discussion on ways to combine XML Schema and RDF Schema to more fully express characteristics of application profiles [HUNTER]. More recently there has been an attempt within the W3C to differentiate RDF Schema as a vocabulary description language and XML Schema as a basis for providing structured data exchange.

An XML schema provides a structured expression that supports validation of instance metadata. In effect, an XML schema provides a document "template" which acts as an exchange format for metadata instances. An XML Schema serves the same function as an XML DTD with additional capability for extensibility and namespace handling.

An RDF schema expresses relationships between terms, providing a data model for expressing the semantics of terms – their properties, classes, and definitions. The underlying RDF data model combined with the use of unique identifiers allows software to infer relationships between terms and perform data aggregation.

RDF Schemas are effective for expressing the semantics of application profiles, whilst XML Schemas are more effective for expressing cardinality, data-typing, and constraints. Possible approaches to the expression of application profiles in RDF have been explored within projects such as SCHEMAS [BAKER] and MEG [MEG-REGISTRY].



---

## Bibliography

[BAKER] Thomas Baker, Makx Dekkers, Rachel Heery, Manjula Patel, Gauri Salokhe, What terms does your metadata use? Application profiles as machine-understandable narratives. *Journal of Digital Information* 2:2 (November 2001), <http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Baker>.

[CORES-RESOLUTION] CORES Resolution on Metadata Element Identifiers, <http://www.cores-eu.net/interoperability/>.

[DC-LIBRARY] Library Application Profile, <http://dublincore.org/documents/2002/09/24/library-application-profile/>.

[DCMI-NAMESPACE] Andy Powell, Harry Wagner, Stuart Weibel, Tom Baker, Tod Matola, Eric Miller, Namespace policy for the Dublin Core Metadata Initiative, <http://dublincore.org/documents/dcmi-namespaces/>.

[DCMI-PRINCIPLES] DCMi Grammatical Principles, <http://dublincore.org/usage/documents/principles/>.

[DCMI-SCHEMAS] DCMi Schemas, <http://dublincore.org/schemas/>.

[DCMI-TERMS] DCMi Metadata Terms, <http://dublincore.org/documents/dcmi-terms/>.

[HEERY] Rachel Heery, Manjula Patel, Application profiles: mixing and matching metadata schemas, *Ariadne* 25, September 2000, <http://www.ariadne.ac.uk/issue25/app-profiles/intro.html>.

[HUNTER] Jane Hunter, Carl Lagoze, Combining RDF and XML Schemas to enhance interoperability between metadata application profiles. *WWW10*, May 1-5, 2001, Hong Kong, <http://www10.org/cdrom/papers/572/index.html>.

[MEG-REGISTRY] Rachel Heery, Pete Johnston, Dave Beckett, Damian Steer, The MEG Registry and SCART: Complementary Tools for Creation, Discovery and Re-use of Metadata Schemas. In: *Proceedings of the International Conference on Dublin Core and Metadata for e-Communities*, 2002. Florence: Firenze University Press, 2002, pp. 125-132, <http://www.bncf.net/dc2002/program/ft/paper14.pdf>.

[RDF-SCHEMA] Brickley, Dan and Guha, R.V, editors. *RDF Vocabulary Description Language 1.0: RDF Schema* W3C Working Draft 23 January 2003, <http://www.w3.org/TR/rdf-schema/>.

[URI] T. Berners-Lee, R. Fielding, L. Masinter, Uniform Resource Identifiers (URI): Generic Syntax, August 1998, <http://www.ietf.org/rfc/rfc2396.txt>.

[WEBARCH] Ian Jacobs, ed., *Architecture of the World Wide Web*, <http://www.w3.org/TR/webarch/>.

[XML-SCHEMA] Thompson, Henry S. et al., editors. *XML Schema Part 1: Structure*. W3C Recommendation 2 May 2001, <http://www.w3.org/TR/xmlschema-1/>.