



**PRISM:
Publishing Requirements for Industry Standard Metadata**

Version 3.0

**PRISM Subset of the Dublin Core
Metadata Specification**

October 4, 2012



Copyright and Legal Notices

© 2001 – 2012 International Digital Enterprise Alliance, Inc. All Rights Reserved.

PRISM® and nextPub® are registered trademarks of the International Digital Enterprise Alliance, Inc. (IDEAlliance).

This document may be downloaded and copied provided that the above copyright notice and this Notice are included on all such copies. This document itself may not be modified in any way, except as needed for the purpose of developing International Digital Enterprise Alliance, Inc. ("IDEAlliance") specifications. Use of the specification or standard set forth in this document shall not create for the user any rights in or to such specification or standard or this document, which rights are exclusively reserved to IDEAlliance or its licensors or contributors.

Use of this document and any specification or standard contained herein is voluntary. By making use of this document or any specification or standard contained herein, the user assumes all risks and waives all claims against IDEAlliance, its licensors and contributors. By making this document available, IDEAlliance is not providing any professional services or advice to any person or entity. Any person or entity utilizing this document or any specification or standard contained herein should rely upon the advice of a competent professional before using any such information.

NO WARRANTY, EXPRESSED OR IMPLIED, IS MADE REGARDING THE ACCURACY, ADEQUACY, COMPLETENESS, LEGALITY, RELIABILITY OR USEFULNESS OF ANY INFORMATION CONTAINED IN THIS DOCUMENT OR IN ANY SPECIFICATION OR STANDARD OR OTHER PRODUCT MADE AVAILABLE BY IDEALLIANCE. THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN AND INCLUDED IN ANY SPECIFICATION OR STANDARD OR OTHER PRODUCT OR SERVICE OF IDEALLIANCE IS PROVIDED ON AN "AS IS" BASIS. IDEALLIANCE DISCLAIMS ALL WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY ACTUAL OR ASSERTED WARRANTY OF NON-INFRINGEMENT OF PROPRIETARY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL IDEALLIANCE, ITS LICENSEES, CONTRIBUTORS OR THEIR RESPECTIVE OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, REPRESENTATIVES, SUPPLIERS OR CONTENT OR SERVICE PROVIDERS BE LIABLE FOR DAMAGES OF ANY KIND, INCLUDING WITHOUT LIMITATION, DIRECT, INDIRECT, COMPENSATORY, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FROM DATA LOSS OR BUSINESS INTERRUPTION) EVEN IF MADE AWARE OF THE POSSIBILITY OF SUCH DAMAGES, WHETHER IN AN ACTION UNDER CONTRACT, TORT OR ANY OTHER THEORY, ARISING OUT OF OR IN CONNECTION WITH THE USE, INABILITY TO USE OR PERFORMANCE OF THIS DOCUMENT, THE SPECIFICATION OR STANDARD CONTAINED HEREIN, OR ANY OTHER DOCUMENT OR SPECIFICATION OR STANDARD MADE AVAILABLE BY IDEALLIANCE.

Some states do not allow the disclaimer or limitation of damages, so the disclaimers set forth above apply to the maximum extent permitted under applicable law.

IDEAlliance takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed or implicated with respect to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available. IDEAlliance does not represent that it has made any effort to identify any such rights. Information on IDEAlliance's procedures with respect to rights in IDEAlliance specifications can be found at the IDEAlliance website at www.idealliance.org. Copies of third-party claims of rights, assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the President of IDEAlliance at patent-disclosure@idealliance.org.

IDEAlliance requests interested parties to disclose any copyrights, trademarks, service marks, patents, patent applications, or other proprietary or intellectual property rights which may cover technology that may be required to implement this specification. Please address the information to the President of IDEAlliance at patent-disclosure@idealliance.org.

Table of Contents

- 1 Status..... 1**
 - 1.1 Document Status1
 - 1.2 Document Location1
 - 1.3 Version History.....1
- 2 PRISM Documentation Structure 3**
 - 2.1 Normative and Non-normative Sections3
 - 2.2 Requirement Wording Note3
 - 2.3 The PRISM Documentation Package3
 - 2.3.1 General Documents..... **Error! Bookmark not defined.**
 - 2.3.2 PRISM Metadata Specifications..... **Error! Bookmark not defined.**
 - 2.3.3 PRISM Aggregator Message Markup Specification..... **Error! Bookmark not defined.**
 - 2.3.4 PRISM Inline Markup Specification..... **Error! Bookmark not defined.**
 - 2.3.5 PRISM Controlled Vocabulary Specifications **Error! Bookmark not defined.**
 - 2.3.6 Additional PRISM Documentation **Error! Bookmark not defined.**
 - 2.3.7 Access to PRISM Documentation..... **Error! Bookmark not defined.**
 - 2.3.8 Access to PAM Schemas..... **Error! Bookmark not defined.**
 - 2.3.9 PRISM Source Vocabulary Documentation Set **Error! Bookmark not defined.**
 - 2.4 PSV Content Management Schema3
 - 2.5 Other PSV Schemas7
- 3 Introduction 9**
 - 3.1 Purpose and Scope9
 - 3.2 New in this Version.....9
 - 3.3 Dublin Core Namespaces.....9
 - 3.4 PRISM Subset of Dublin Core Element and Attribute Models9
 - 3.4.1 dc:contributor 10
 - 3.4.2 dc:creator..... 11
 - 3.4.3 dc:description 12
 - 3.4.4 dc:format 14
 - 3.4.5 dcterms:hasFormat..... 14
 - 3.4.6 dcterms:hasPart 15

PRISM Subset of the Dublin Core Metadata Version 3.0

- 3.4.7 dcterms:hasVersion 16
- 3.4.8 dc:identifier..... 17
- 3.4.9 dcterms:isFormatOf 18
- 3.4.10 dcterms:isPartOf..... 19
- 3.4.11 dcterms:isRequiredBy 20
- 3.4.12 dcterms:isVersionOf..... 21
- 3.4.13 dc:language 22
- 3.4.14 dc:publisher..... 22
- 3.4.15 dc:relation..... 23
- 3.4.16 dcterms:requires 24
- 3.4.17 dc:rights 25
- 3.4.18 dc:source..... 25
- 3.4.19 dcterms:source 26
- 3.4.20 dc:subject 27
- 3.4.21 dcterms:subject..... 28
- 3.4.22 dc:title 28
- 3.4.23 dc:type 29

1 STATUS

1.1 Document Status

The status of this document is:

✓	Draft	11/04/2011
✓	Released for Public Comment	12/15/2012
✓	Final Draft Released for Comment	06/12/2012
✓	Final Specification	10/04/2012

1.2 Document Location

The location of this document is:

http://www.prismstandard.org/specifications/3.0/PRISM_Dublin_Core_3.0.pdf

or

http://www.prismstandard.org/specifications/3.0/PRISM_Dublin_Core_3.0.xmp

1.3 Version History

Version Number	Release Date	Editor	Description
1.2	1/26/05	McConnell	Converted from unmodularized PRISM spec v 1.2
1.3	10/01/05	Kennedy	Resolve open industry comments
2.0	2/19/08	Kennedy	Final Release Version
2.1	05/15/09	Kennedy	Comments Resolved
2.1 Errata 2010	07/01/10	Kennedy	Errata Complete
3.0 Draft	12/15/2011	Kennedy	Public Draft
3.0 Final Draft	06/12/2012	Kennedy	Final Draft Spec, comments resolved
3.0 Final	10/04/2012	Kennedy	Final Specification

2 PRISM DOCUMENTATION STRUCTURE

PRISM is described in a set of formal, modularized documents that, taken together, represent “the PRISM Specification”. Together these documents comprise the PRISM Documentation Package.

2.1 Normative and Non-normative Sections

Documents in the PRISM Documentation Package may contain both normative and non-normative material; normative material describes element names, attributes, formats, and the contents of elements that is required in order for content or systems to comply with the PRISM Specification. Non-normative material explains, expands on, or clarifies the normative material, but it does not represent requirements for compliance. Normative material in the PRISM Documentation Package is explicitly identified as such; any material not identified as normative can be assumed to be non-normative.

2.2 Requirement Wording Note

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119]. The PRISM Specification also uses the normative term, “STRONGLY ENCOURAGES,” which should be understood as a requirement equivalent to “MUST” in all but the most extraordinary circumstances.

Capitalization is significant; lower-case uses of the key words are intended to be interpreted in their normal, informal, English language way.

2.3 The PRISM Documentation Package

The PRISM Documentation Package consists of:

2.3.1 General Documents

This is a set of general or overview documents that apply to PRISM.

Document	Description
PRISM Introduction [PRISMINT] http://www.prismstandard.org/specifications/3.0/PRISM_introduction_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_introduction_3.0.htm	Overview, background, purpose and scope of PRISM; examples; contains no normative material.
PRISM Compliance [PRISMCOMP] http://www.prismstandard.org/specifications/3.0/PRISM_compliance_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_compliance_3.0.htm	Describes three profiles of PRISM compliance for content and systems; includes normative material.

2.3.2 PRISM Metadata Specifications

This is the set of documents that outline the prism metadata fields and values by PRISM metadata category. PRISM has modularized its metadata specification by namespace so users may pick those modules that meet their unique business requirements without having to implement the entire PRISM specification.

PRISM Subset of the Dublin Core Metadata Version 3.0

Document	Description
The PRISM Basic Metadata Specification [PRISMBMS] http://www.prismstandard.org/specifications/3.0/PRISM_Basic_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_Basic_Metadata_3.0.htm	Describes the basic metadata elements contained in the PRISM namespace to describe article content; includes normative material.
PRISM Advertising Metadata Specification [PRISMADMS] http://www.prismstandard.org/specifications/3.0/PRISM_Advertising_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_Advertising_Metadata_3.0.htm	Describes advertising metadata elements including those drawn from AdsML, GWG and Ad-ID; includes normative material.
The PRISM Subset of Dublin Core Metadata Specification [PRISMDCMS] http://www.prismstandard.org/specifications/3.0/PRISM_Dublin_Core_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_Dublin_Core_Metadata_3.0.htm	Describes the metadata elements from the Dublin Core namespace that are included in PRISM; includes normative material.
The PRISM Image Metadata Specification [PRISMIMS] http://www.prismstandard.org/specifications/3.0/PRISM_Image_Metadata_Specification_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_Image_Metadata_Specification_3.0.htm	Describes the metadata elements contained in the PRISM Metadata for Images Namespace and other related image namespaces, includes normative material.
The PRISM Recipe Metadata Specification [PRISMRMS] http://www.prismstandard.org/specifications/3.0/PRISM_Recipe_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_Recipe_Metadata_3.0.htm	Describes the metadata elements contained in the PRISM Recipe Metadata Namespace, includes normative material
The PRISM Usage Rights Metadata Specification [PRISMURMS] http://www.prismstandard.org/specifications/3.0/PRISM_Usage_Rights_Metadata_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_Usage_Rights_Metadata_3.0.htm	Describes the metadata elements contained in the PRISM Usage Rights Namespace; includes normative material. This namespace will supersede elements in both the prism: and prl: namespaces in version 3.0 of the specification.

2.3.3 PRISM Aggregator Message Markup Specification

This module documents the PRISM Markup Elements and Attributes for use with the PRISM Aggregator Message. At the time of the publication of the Introduction to PRISM, the PAM Message remains at version 2.1. This set of documents includes:

Document	Description
The PRISM PAM Markup Specification [PRISMPAMMS] http://www.prismstandard.org/specifications/2.1/PRISM_PAM_Markup_2.1.pdf or http://www.prismstandard.org/specifications/2.1/PRISM_PAM_Markup_2.1.htm	Describes the XML elements and attributes used to encode the PRISM Aggregator Message from both the pam: and pim: namespaces; includes normative material.

PRISM Subset of the Dublin Core Metadata Version 3.0

2.3.4 PRISM Inline Markup Specification

This module documents the PRISM Inline Markup Elements and Attributes for use with the PRISM Aggregator Message. This set of documents includes:

Document	Description
The PRISM Inline Markup Specification [PRISMIMS] http://www.prismstandard.org/specifications/2.1/PRISM_PIM_Markup_Specification_3.0.pdf or http://www.prismstandard.org/specifications/2.1/PRISM_PIM_Markup_Specification_3.0.htm	Describes the XML elements used to encode the inline markup for the PRISM Aggregator Message. Includes normative material.

2.3.5 PRISM Controlled Vocabulary Specifications

These modules are new with PRISM 3.0. All controlled vocabularies and their terms are documented in this publication set.

Document	Description
The PRISM Controlled Vocabulary Markup Specification [PRISMCVMS] http://www.prismstandard.org/specifications/3.0/PRISM_Controlled_Vocabulary_Markup_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_Controlled_Vocabulary_Markup_3.0.htm	Describes the metadata fields in the PRISM Controlled Vocabulary Namespace that can be used to describe a controlled vocabulary. Actual PRISM controlled vocabularies are now placed in the PRISM Controlled Vocabularies Specification [PRISMCVS]
The PRISM Controlled Vocabularies Specification [PRISMCVS] http://www.prismstandard.org/specifications/3.0/PRISM_CV_Spec_3.0.pdf or http://www.prismstandard.org/specifications/3.0/PRISM_CV_Spec_3.0.htm	The PRISM Controlled Vocabularies are now documented in this document.

2.3.6 Additional PRISM Documentation

The [Guide to the PRISM Aggregator Message](#) [[PAMGUIDE](#)] documents the PRISM Aggregator Message (PAM), an XML-based application of PRISM.

The [PRISM Cookbook](#) [[PRISMCB](#)] documents implementation strategies for PRISM Profile 1 applications.

The [Guide to PRISM Usage Rights](#) [[RIGHTSGUIDE](#)] documents an XML-based PRISM application for the expression of PRISM Usage Rights. The Guide is accompanied by an XSD that can be used as the basis for developing a digital rights management system based on PRISM Usage Rights.

The [Guide to PRISM Metadata for Images](#) [[IMAGEGUIDE](#)] documents an XML-based PRISM Profile 1 application for the expression of the structure and use of PRISM Metadata for Images and can be used as the basis for developing an image management system based on PRISM Metadata for Images and for implementing PMI in XML.

The [Guide to PRISM Recipe Metadata and XML Encoding](#) [[RECIPEGUIDE](#)] documents the XML-based PRISM Profiles for the encoding of recipes for:

- Establish a Recipe Database

PRISM Subset of the Dublin Core Metadata Version 3.0

- Establish a tagging scheme to code a wide variety of recipes in XML
- Tag recipes within the PAM message
- Tag recipes in nextPub XML Content Source

2.3.7 Access to PRISM Documentation

The PRISM documentation package, the PAM guide (see above), the PAM DTD, the PAM XSD and a range of other information concerning PRISM are all publicly and freely available on the PRISM website, www.prismstandard.org.

2.3.8 Access to PAM Schemas

Standard URLs have been established to access PRISM/PAM XSDs and DTDs as well as the XSD for the new PRISM Usage Rights Model.

To access PAM XSDs and DTDs:

<http://www.prismstandard.org/schemas/pam/2.1/>

<http://www.prismstandard.org/schemas/pam/2.1/pam.xsd>

<http://www.prismstandard.org/schemas/pam/2.1/pam-dc.xsd>

<http://www.prismstandard.org/schemas/pam/2.1/pam-prism.xsd>

To access PRISM Rights Model XSD

<http://www.prismstandard.org/schemas/rights/3.0/rightsmodel.xsd>

To access PRISM Recipe Tagging and Recipe Database XSD

<http://www.prismstandard.org/schemas/recipe/3.0/recipe.xsd>

2.3.9 nextPub PRISM Source Vocabulary Documentation Set

nextPub has developed a series of specifications collectively known as the PRISM Source Vocabulary. The use case for PSV is to encode semantically rich content for transformation and delivery to any platform. This Specification is made up of a modular documentation package that builds on PRISM 3.0 and HTML5. Over time new modules may be added to the documentation package. The documentation package for the nextPub PRISM Source Vocabulary Specification Version 1.0 consists of:

Document	Description
PRISM Source Vocabulary Specification Overview [PSVSO] http://www.prismstandard.org/specifications/psv/1.0/PSV_overview.pdf or http://www.prismstandard.org/specifications/psv/1.0/PSV_overview.htm	The Introduction to the PRISM Source Vocabulary provides an introduction and a non-technical overview of the PRISM Source Vocabulary.
PRISM Source Vocabulary Specification [PSVS] http://www.prismstandard.org/specifications/psv/1.0/PSV.pdf or http://www.prismstandard.org/specifications/psv/1.0/PSV.htm	The PRISM Source Vocabulary Specification defines semantically rich for source metadata and content markup that can be transformed and served to a wide variety of output devices including eReaders, mobile tablet devices, smart phones and print.

PRISM Subset of the Dublin Core Metadata Version 3.0

Document	Description
PRISM Source Vocabulary Markup Specification [PSVMS] http://www.prismstandard.org/specifications/psv/1.0/PSV_markup.pdf or http://www.prismstandard.org/specifications/psv/1.0/PSV_markup.htm	The PSV Markup Specification documents the XML tags in the PSV namespace that are used to encode XML Source Content.
PAM to PSV_Guide [PAMPSVGUIDE] http://www.prismstandard.org/specifications/psv/1.0/PAM_PSV.pdf or http://www.prismstandard.org/specifications/psv/1.0/PAM_PSV.htm	This Guide documents mappings from PAM XML to PSV XML. It is normative only.

2.4 PSV Content Management Schema

In order to assist implementers develop a PSV-based federated content management solution, the nextPub Working Group is providing an XML Schema (XSD) that can serve as the basis for the design of a PSV content repository.

Note: The PSV CM schema is not designed for tagging content. It is provided simply to serve as a basis for the design of a content repository. Metadata building blocks from this schema can be combined with HTML5 by publishers who wish to develop a hybrid PSV metadata and content tagging schema.

2.5 Other PSV Schemas

Because PSV is a flexible framework, it supports many different use case scenarios. A different schema, using the PSV metadata fields and content encoding can be developed for each different use case. In order to assist PSV implementers, the nextPub Working Group is planning to provide a number of XML Schemas (XSDs) to support common use cases including tagging an article and transmitting articles to content aggregators. These PSV sample schemas will be available from the nextPub website (<http://www.nextpub.org>) and documented in the nextPub PSV Implementation Guide that will be published following the publication of this specification.

3 INTRODUCTION

3.1 Purpose and Scope

The purpose of this document is to describe the elements that PRISM includes from the Dublin Core namespace. For the Dublin Core specification, see [[DCMI-TERMS](#)]. All of section 4 of this document is normative.

All the element definitions appear in a uniform format. Each element definition begins with two fields – the Name and the Identifier of the element. The Name is a human-readable string that can be translated into different languages. Also, note that PRISM does NOT require that users be presented with the same labels. The Identifier is a protocol element. It is an XML element type and MUST be given as shown, modulo the normal allowance for variations in the namespace prefix used.

Note: This document describes element models and provides examples for all PRISM profiles. In addition, Profile 1 PRISM (well formed XML, with no requirement for RDF), is described in [Guide to the PRISM Aggregator Message V. 2.1](#) [[PAMGUIDE](#)].

3.2 New in this Version

See *PRISM Introduction 3.0* [[PRISMINT](#)] for all changes.

Changes in this document include:

- PRISM Dublin Core Namespaces Specification [[PRISMDCNS](#)] has been Renamed PRISM Dublin Core Metadata Specification [[PRISMDCMS](#)]

3.3 Dublin Core Namespaces

The normative definitions of the Dublin Core elements can be found in [[DCMI](#)] Dublin Core Metadata Element Set, Version 1.1 and [[DCMI-TERMS](#)] Dublin Core Metadata Terms, 2005-01-10. The use of some dc: elements are encouraged, others are discouraged, and others constrained. None of the Dublin Core elements are required to appear in a PRISM description -- except dc:identifier, under profile one compliance; see [[PRISMCOMP](#)] -- and all of them are repeatable any number of times.

The recommended PRISM namespace for Dublin Core is:

```
xmlns:dc="http://purl.org/dc/elements/1.1/"
```

and for Dublin Core Terms is:

```
xmlns:dcterms="http://purl.org/dc/terms/"
```

3.4 PRISM Subset of Dublin Core Element and Attribute Models

All three PRISM profiles are documented in this section. First Profile #1 is documented. The documentation for the XML only profile includes a field that indicates whether this element is included in the PRISM Aggregator Message. If the element is included in PAM, please refer to [Guide to the PRISM Aggregator Message](#) [[PAMGUIDE](#)] for more detailed information about the use of the element in the context of the XML PAM message.

PRISM Subset of the Dublin Core Metadata Version 3.0

PRISM Profile #2 (RDF/XML) is also documented in this section. In combining XML with RDF, there is far greater flexibility in tagging than we are used to when we define XML elements and attributes with an XML DTD. The remainder of this section contains the most likely element/attribute models for profile 2 PRISM. Other profile 2 models are possible based on the interaction between XML and RDF.

PRISM Profile #3 (XMP) is also documented in this section. The documentation concentrates on the property and container values for the XMP field to provide information required to develop an XMP schema to implement PRISM in the XMP environment. Note that XMP can be particularly useful in extending the capability of encoding multimedia objects with PRISM metadata.

3.4.1 dc:contributor

Name	Contributor
Identifier	dc:contributor
Definition	An entity responsible for making contributions to the content of a media resource.
Occurrence	Occurs 0 or more times
Comment	Dublin Core recommends that dc:contributor identify a person, an organization, or a service by name. PRISM recommends that magazine publishers use dc:contributor for people who do additional reporting, or individuals who would be called out for special acknowledgments, such as research assistants. Place and role attributes may be used in conjunction with the element to indicate the contributor's role and place of reporting.
Included in PAM?	Yes
Included in PSV?	Yes
Profile 1 (XML)	Recommended practice is to use a prism:role attribute inside dc:contributor and to specify role with values from the PRISM Controlled Vocabulary of Contributor Roles [PRISMCVS]
Model #1	
Element Content	String
Attributes	(Optional) xml:lang = designed for identifying the human language used (Optional) prism:role= value from Contributor Role Controlled Vocabulary [PRISMCVS] (Optional) prism:place = string indicating location for the contributor (Optional) prism:contactInfo = string indicating contact information for the contributor
Examples	<dc:contributor prism:role="writer" prism:place="New York City">Erin Clark</dc:contributor> <dc:contributor prism:role="editor">Ed Stevenson</dc:contributor> <dc:contributor prism:role="graphicDesigner">Lee Vetten</dc:contributor>
Profile 2 (RDF)	If there are multiple contributors for the resource PRISM recommends listing the multiple contributors inside one dc:contributor element using the RDF containers such as rdf:Bag, rdf:Seq or rdf:Alt to be XMP compatible.
Model #1	
Element Content	Plain Literal
Attributes	rdf:resource authority reference (rdf:resource) to indicate contributor type value from PRISM Contributor Role Controlled Vocabulary
Example	<dc:contributor rdf:resource="contributorrole.xml#writer">Stephanie Salmon</dc:contributor>
Model #2	

PRISM Subset of the Dublin Core Metadata Version 3.0

Element Content	Plain Literal (with multiple contributors) (Optional) prism:role= value from Contributor Role Controlled Vocabulary [PRISM CVS] prism:place = string indicating location for the contributor
Attributes	(Optional) xml:lang = designed for identifying the human language used (Optional) prism:role= value from Contributor Role Controlled Vocabulary [PRISM CVS] (Optional) prism:place = string indicating location for the contributor (Optional) prism:contactInfo = string indicating contact information for the contributor
Examples	Model #1 <pre><dc:contributor prism:role="writer" prism:location="New York City">Erin Clark</dc:contributor> <dc:contributor prism:role="editor">Ed Stevenson</dc:contributor> <dc:contributor prism:role="graphicDesigner">Lee Vetten</dc:contributor></pre> Model #2 <pre><dc:contributor> <rdf:Bag> <rdf:li rdf:resource="contributorrole.xml#writer">Erin Clark</rdf:li> <rdf:li rdf:resource="contributorrole.xml#editor">Ed Stevenson</rdf:li> <rdf:li rdf:resource="contributorrole.xml#graphicDesigner">Lee Vetten</rdf:li> </rdf:Bag> </dc:contributor></pre>
Profile 3 (XMP)	
Property Value	Model #1 Full interoperability with existing XMP Property Value dc:contributor bag ProperName (no support for role or place) Model #2 Parallel dc:terms element adding role and place information. Property Value dcterms:contributor bag dctermsContributor structure contributor.name Text contributor.a-role Text, closed choice prism:role controlled vocabulary contributor.a-place Text

3.4.2 dc:creator

Name	Creator
Identifier	dc:creator
Definition	An entity primarily responsible for creating the content of a media resource.
Comment	Dublin core recommends that dc:creator include a person, an organization, or a service. Place and role attributes may be used in conjunction with the element to indicate the contributor's role and place of reporting.
Occurrence	Occurs 0 or 1 time
Included in PAM?	Yes
Included in PSV?	Yes
Profile 1 (XML)	Recommended practice is to use the prism:role attribute inside dc:creator and to use a value from the PRISM Controlled Vocabulary of Creator Roles.
Model #1	
Element Content	String
Attributes	(Optional) xml:lang = designed for identifying the human language used (Optional) prism:role= value from Role Controlled Vocabulary [PRISM CVS]

PRISM Subset of the Dublin Core Metadata Version 3.0

	(Optional) prism:place = string indicating location for the creator (Optional) prism:contactInfo = string indicating contact information for the creator (Optional) xml:lang = designed for identifying the human language used
Example	<dc:creator prism:role="writer" prism:location="New York City">Erin Clark</dc:creator> <dc:creator prism:role="editor">Ed Stevenson</dc:creator> <dc:creator prism:role="graphicDesigner">Lee Vetten</dc:creator>
Profile 2 (RDF)	Recommended practice for PRISM implementations is to use a value from the PRISM Controlled Vocabulary of Creator Roles [PRISMCVS] , expressed as a URI reference. Implementations MUST also be able to handle text values, but interoperability with text values cannot be guaranteed.
Model #1	
Element Content	Plain Literal
Attributes	(Optional) xml:lang = designed for identifying the human language used (Optional) prism:role= value from Role Controlled Vocabulary [PRISMCVS] (Optional) prism:place = string indicating location for the creator (Optional) prism:contactInfo = string indicating contact information for the creator (Optional) xml:lang = designed for identifying the human language used
Example	<dc:creator rdf:resource="creatorrole.xml#writer">Stephanie Salmon</dc:creator>
Model #2	
Element Content	Plain Literal (with multiple contributors)
Examples	Model #1 <dc:creator prism:role="writer" prism:location="Carlsbad, CA">Carl Rambert</dc:creator> <dc:creator prism:role="editor">Linda Burman</dc:creator> <dc:creator prism:role="graphicDesigner">Allegra Hartley</dc:creator> Model #2 <dc:creator> <rdf:Bag> <rdf:li rdf:resource="creatorrole.xml#writer">Linda Burman</rdf:li> <rdf:li rdf:resource="creatorrole.xml#editor">Carl Rambert</rdf:li> <rdf:li rdf:resource="creatorrole.xml#graphicDesigner">Allegra Hartley</rdf:li> </rdf:Bag> </dc:creator>
Profile #3 (XMP)	
Property Value	Model #1 Full interoperability with existing XMP Property Value dc:creator seq ProperName (no support for role or place) Model #2 Parallel dc:terms element adding role and place information. Property Value dcterms:creator seq dctermsCreator structure creator.name Text creator.a-role Text, closed choice prism:role controlled vocabulary creator.a-place Text

3.4.3 dc:description

Name	Description
Identifier	dc:description
Definition	An account of the content of the resource.

PRISM Subset of the Dublin Core Metadata Version 3.0

Occurrence	Occurs 0 or more times
Comment	<p>The Dublin Core Metadata Initiative recommends that dc:description MAY contain any information (e.g., an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content) that describes the resource.</p> <p>For PRISM, this element describes the resource, such as an abstract or a deck head. Note that this is intended to appear in metadata for an article, not as inline markup. (In other words, a DTD for articles might have dc:description in the header, but would use elements like <abstract> or <deck> for the markup of such material in the body of the article.)</p> <p>Short descriptions, such as those in the Table of Contents of a magazine, or in the results list of an online search, SHOULD be given in the prism:teaser element.</p> <p>The dc:description element MAY refer to separate descriptions, such as an abstract prepared by an A&I service, by providing the URI of the description as the value of an rdf:resource attribute. In this case, the description is a separate, standalone resource which could have its own metadata. The metadata record for the separate abstract should contain a prism:genre="abstract" and a dc:source element pointing back to the original article.</p>
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	xml:lang = (optional) designed for identifying the human language used
Example	<dc:description>Browse our catalog for the right computer for you.</dc:description>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource) xml:lang = (optional) designed for identifying the human language used.
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used.
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used.
Examples	<p>Model #1</p> <pre><dc:description rdf:resource="http://www2.rhbc.ac.uk/Music/Archive/Disserts/attinell.html"/></pre> <p>Model #2</p> <pre><dc:description>Browse our catalog for the right computer for you.</dc:description></pre> <p>Model #3</p> <pre><dc:description rdf:parseType="Literal">Describes the infamous criminal and gunfighter, Billy the Kid.</dc:description></pre>

PRISM Subset of the Dublin Core Metadata Version 3.0

Profile #3 (XMP)	
Property Value	<p>Model #1 Full interoperability with existing XMP (0 or 1 occurrence, alternate versions by language) Property Value dc:description Lang Alt</p> <p>Model #2 Parallel dcterms element to enable "0 or more" occurrences Property Value dcterms:description bag Text</p>

3.4.4 dc:format

Name	Format
Identifier	dc:format
Definition	The physical or digital manifestation of the resource.
Occurrence	Occurs 0 or 1 time
Comment	<p>Dublin Core recommends that dc:format include the media-type or dimensions of the resource. Format may be used to determine the software, hardware or other equipment needed to display or operate the resource. Examples of dimensions include size and duration.</p> <p>PRISM focuses on systems where resources are digital content, not physical objects. Therefore, it is strongly encouraged that PRISM-compliant systems sending PRISM records restrict values of the dc:format element to those in list of Internet Media Types [IETF-MIMETYPES]. PRISM-compliant systems receiving descriptions MAY wish to detect when format values are strings other than media types in order to allow application-appropriate handling.</p> <p>In general, the top-level of an Internet Media Type is used to declare the general type of data, while the subtype specifies a specific format for that type of data. For example, image/jpeg or text/plain.</p>
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dc:format>application/pdf</dc:format>
Profile #2 (RDF)	
Model #1	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Example	<dc:format>application/pdf</dc:format>
Profile #3 (XMP)	
Property Value	MIME Type

3.4.5 dcterms:hasFormat

Name	Has Format
Identifier	dcterms:hasFormat

PRISM Subset of the Dublin Core Metadata Version 3.0

Definition	Identifies another resource, which is essentially the same intellectual content as the current resource, but presented in another file format, or after some mechanical transformation such as a conversion to a different resolution, different color depth, etc.
Occurrence	Occurs 0 or more times
Comment	<p>The dcterms:hasFormat element points from the original resource, to the alternative version derived from it. In other words, the metadata of the original resource will contain the dcterms:hasFormat element. The dcterms:isFormatOf element is used to point in the other direction, from the alternative back to the original. If the 'original' version cannot be determined, use dcterms:hasFormat for both directions of the relationship.</p> <p>For profile 2, if there are multiple alternative formats for the resource PRISM recommends listing the multiple formats inside one dcterms:hasFormat element using the RDF containers such as rdf:Bag, rdf:Seq or rdf:Alt to be XMP compatible. For profile 1, simple repeat the dcterms:hasFormat element multiple times.</p>
Included in PAM?	No
Included in PSV?	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dcterms:hasFormat>http://freeimages.com/pool.jpg</dcterms:hasFormat>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	<p>Model #1 <dcterms:hasFormat rdf:resource="http://wap.wanderlust.com/2000/08/Belize.wml"/></p> <p><dcterms:hasFormat rdf:resource="doi:123/p92-1293"/></p> <p>Model #2 <dcterms:hasFormat>photo1293.jpg</dcterms:hasFormat></p> <p>Model #3 <dcterms:hasFormat rdf:parseType="Literal"> doi&colon:123/p92&endash;1293</dcterms:hasFormat></p>
Profile 3 (XMP)	
Qualifier Value	bag Text, a refined version of dc:relation

3.4.6 dcterms:hasPart

Name	Has Part
Identifier	dcterms:hasPart
Definition	The described resource includes the referenced resource either physically or logically.
Occurrence	Occurs 0 or more times

PRISM Subset of the Dublin Core Metadata Version 3.0

Comment	<p>dcterms:hasPart allows the metadata for an article to identify images, sidebars, tables, graphs, maps, illustrations, etc. in the article which exist as separate, identifiable, resources. The metadata for those resources can then be fetched, based on the identifier for the included resource. The identifier must be non-literal, typically a URI.</p> <p>Recommended best practice is to describe photos, etc. as separate objects, rather than embedding their metadata in the metadata for an article, in order to ease their reuse and to simplify data maintenance when the resources are reused. Best practice is also to identify the resources with URIs, rather than human-readable text descriptions, in order to enable automated handling of the resource.</p> <p>For profile 2, if there are multiple parts for the resource, PRISM recommends listing the multiple parts inside one dcterms:hasPart element using the RDF containers such as rdf:Bag, rdf:Seq or rdf:Alt to be XMP compatible. For profile 1, simply repeat the dcterms:hasPart element multiple times.</p>
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	xml:lang = (optional) designed for identifying the human language used
Example	<dcterms:hasPart>http://freeimages.com/pool.jpg</dcterms:hasPart>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference
Attributes	Resource Reference (rdf:resource) xml:lang = (optional) designed for identifying the human language used
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	<p>Model #1 <dcterms:hasPart rdf:resource="http://www.travelmongo.com/2000/08/BelizePhoto.jpg"/></p> <p>Model #2 <dcterms:hasPart>http://www.travelmongo.com/2000/08/BelizePhoto.jpg</dcterms:hasPart></p> <p>Model #3 <dcterms:hasPart rdf:parseType="Literal">dam&endash;obj&endash;32485u2</dcterms:hasPart></p>
Profile 3 (XMP)	
Qualifier Value	bag Text, a refined version of dc:relation

3.4.7 dcterms:hasVersion

Name	Has Version
Identifier	dcterms:hasVersion
Definition	Identifies the next version of the current resource.

PRISM Subset of the Dublin Core Metadata Version 3.0

Occurrence	Occurs 0 or 1 time
Comment	Changes in version imply substantive changes in intellectual content rather than differences in format. For changes in format, use the <code>dcterms:hasFormat</code> element. For the special case of versions known as “corrections”, use <code>prism:hasCorrection</code> to point from the current resource to correction blocks. Use <code>dcterms:hasVersion</code> to point from the corrected resource back to the earlier one.
Included in PAM?	No
Included in PSV?	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<code><dcterms:hasVersion>http://freeimages.com/pool.jpg</dcterms:hasVersion></code>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (<code>rdf:resource</code>)
Model #2	
Element Content	Plain Literal
Attributes	<code>xml:lang = (optional)</code> designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	<code>rdf:parseType=“Literal”</code> <code>xml:lang = (optional)</code> designed for identifying the human language used
Examples	<p>Model #1 <code><dcterms:hasVersion rdf:resource=</code> <code>“http://travelmongo.com/2000/08/BelizeTravelUpdate.xml”/></code></p> <p>Model #2 <code><dcterms:hasVersion>BelizeTravelUpdate_04.xml</dcterms:hasVersion></code></p> <p>Model #3 <code><dcterms:hasVersion rdf:parseType=“Literal”></code> <code>dam&endash;obj&endash;32485u2</dcterms:hasVersion></code></p>
Profile 3 (XMP)	
Qualifier Value	Text, a refined version of <code>dc:relation</code>

3.4.8 dc:identifier

Name	Identifier
Identifier	<code>dc:identifier</code>
Definition	An unambiguous reference to the resource, within a given context.
Occurrence	Required 1 time for an article. May also be used for each media asset to uniquely identify that asset.
Comment	<p>In PRISM, <code>dc:identifier</code> provides a place for the unique identification of a resource. PRISM requires the use a single <code>dc:identifier</code> statement with a unique identifier to identify a particular published item. Other PRISM identifiers can be applied to provide further identification that is required.</p> <p>You may identify the resource by means of a unique string or number conforming to a formal identification system or generate your own unique identifier. Example formal identification systems include the Uniform Resource Identifier (URI) (including the</p>

PRISM Subset of the Dublin Core Metadata Version 3.0

	Uniform Resource Locator (URL) or the Digital Object Identifier (DOI). If you identify the asset with a DOI or URL in addition to a unique dc:identifier, you may identify the DOI with the prism:DOI element and the URL with the prism:URL. Consistent and thorough use of identifiers is essential for PRISM conformance.
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dc:identifier>10-234/3245</dc:identifier>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Examples	Model #1 <dc:identifier rdf:resource="doi:10.1030/03054"/> Model #2 <dc:identifier>10.1030/03054</dc:identifier> <prism:doi>http://dx.doi.org/10.1030/03054</prism:doi> <prism:url rdf:resource="http://dx.doi.org/10.1030/03054"/>
Profile 3 (XMP)	
Property Value	Text

3.4.9 dcterms:isFormatOf

Name	Is Format Of
Identifier	dcterms:isFormatOf
Definition	The current resource is the same intellectual content of the referenced resource, but presented in another format.
Occurrence	Occurs 0 or more times
Comment	This is the inverse of the dcterms:hasFormat relation. It is used to point from the modified version to an earlier version. It is only used when it is known that the referenced resource is closer to being the 'original' than the current resource. For profile 2, If there are multiple formats for the resource PRISM recommends listing the multiple formats inside one dcterms:isFormatOf element using the RDF containers such as rdf:Bag, rdf:Seq or rdf:Alt to be XMP compatible. For profile 1, simple repeat the dcterms:isFormatOf element multiple times.
Included in PAM?	No
Included in PSV?	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dcterms:isFormatOf>Beize.pdf</dcterms:isFormatOf>
Profile #2 (RDF)	

PRISM Subset of the Dublin Core Metadata Version 3.0

Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	<p>Model #1 <dcterms:isFormatOf rdf:resource="http://wanderlust.com/2000/08/Belize.qxd"/></p> <p>Model #2 <dcterms:isFormatOf>Beize.pdf</dcterms:isFormatOf></p> <p>Model #3 <dcterms:isFormatOf rdf:parseType="Literal"> doi&colon:123/p92&endash;1293</dcterms:isFormatOf></p>
Profile 3 (XMP)	
Qualifier Value	bag Text, a refined version of dc:relation

3.4.10 dcterms:isPartOf

Name	Is Part Of
Identifier	dcterms:isPartOf
Definition	The described resource is a physical or logical part of the referenced resource.
Occurrence	Occurs 0 or more times
Comment	<p>This is the inverse of the dcterms:hasPart relation. Note that it is NOT required to always have both sides of the relationship asserted, as one can be derived from the other.</p> <p>Recommended best practice is to identify the containing resource with a URI. However, textual identifiers are possible so implementations SHOULD be able to accept them, possibly with reduced functionality.</p>
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	xml:lang = (optional) designed for identifying the human language used
Example	<dcterms:isPartOf>BelizeArticle.html</dcterms:isPartOf>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource) xml:lang = (optional) designed for identifying the human language used
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal

PRISM Subset of the Dublin Core Metadata Version 3.0

Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	Model #1 <dcterms:isPartOf rdf:resource= "http://TravelMongo.com/2000/08/BelizeArticle.xml"/> Model #2 <dcterms:isPartOf>BelizeArticle.html</dcterms:isPartOf> Model #3 <dcterms:isPartOf rdf:parseType="Literal">dam&endash;obj&endash;32485u2 </dcterms:isPartOf>
Profile 3 (XMP)	
Qualifier Value	bag Text, a refined version of dc:relation

3.4.11 dcterms:isRequiredBy

Name	Is Required By
Identifier	dcterms:isRequiredBy
Definition	The described resource is required by the referenced resource, either physically or logically.
Occurrence	Occurs 0 or more times
Comment	This is the inverse of the dc:requires relation. For profile 2, If there are multiple "is required by" elements for the resource PRISM recommends listing the multiple values inside one dcterms:isRequiredBy element using the RDF containers such as rdf:Bag, rdf:Seq or rdf:Alt to be XMP compatible. For profile 1, simple repeat the dcterms:isRequiredBy element multiple times.
Included in PAM?	No
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dcterms:isRequiredBy>dl123452345.xml</dcterms:isRequiredBy>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	Model #1 <dcterms:isRequiredBy rdf:resource= "http://wanderlust.com/2000/08/BelizePhoto.jpg"/> Model #2 <dcterms:isRequiredBy>dl123452345.xml</dcterms:isRequiredBy>

PRISM Subset of the Dublin Core Metadata Version 3.0

	Model #3 <dcterms:isRequiredBy rdf:parseType="Literal"> dam&endash;obj&endash;32485u2</dcterms:isRequiredBy>
Profile 3 (XMP)	
Qualifier Value	bag Text, a refined version of dc:relation

3.4.12 dcterms:isVersionOf

Name	Is Version Of
Identifier	dcterms:isVersionOf
Definition	The described resource is a version, edition, or adaptation of the referenced resource. Changes in version imply substantive changes in intellectual content rather than differences in format.
Occurrence	Occurs 0 or 1 time
Comment	This is the inverse of dcterms:hasVersion. PRISM considers versions to be iterative and to have the same dc:identifier. For corrections, use the subproperty prism:isCorrectionOf. For alternative versions that do not have substantive changes in intellectual content, use dcterms:isAlternativeFor.
Included in PAM?	No
Included in PSV?	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dcterms:isVersionOf>Ovid's <i>Ars Amatoria</i> </dcterms:isVersionOf>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	Model #1 <dcterms:isVersionOf rdf:resource= "http://travelmongo.com/2000/08/BelizeTravel.xml"/> Model #2 <dcterms:isVersionOf>http://travelmongo.com/2000/08/BelizeTravel.xml </dcterms:isVersionOf> Model #3 <dcterms:isVersionOf rdf:parseType="Resource"> dam&endash;obj&endash;32485u2</dcterms:isVersionOf>
Profile 3 (XMP)	
Qualifier Value	Text, a refined version of dc:relation

3.4.13 dc:language

Name	Language
Identifier	dc:language
Definition	A language of the intellectual content of the resource.
Occurrence	Occurs 0 to many times
Comment	Recommended best practice for the values of the Language element is defined by RFC 3066 [RFC-3066]. It specifies the use of a two-letter (or three-letter) Language Code taken from the ISO 639 standard [ISO-639] (or from ISO 639-2), optionally followed by a two-letter Country Code (taken from the ISO 3166 standard [ISO-3166]). For example, 'en' for English, 'fr' for French, or 'en-GB' for English used in the United Kingdom.
Included in PAM?	No
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dc:language>en-US</dc:language>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Examples	Model #1 <dc:language rdf:resource=H http://www.din.de/gremien/nas/nabd/iso3166ma/a3ptnorm.htm H/> Model #2 <dc:language>en-US</dc:language>
Profile 3 (XMP)	
Property Value	bag Text, first entry (dc:language/*[1]) closed choice RFC 3066 [RFC-3066] controlled vocabulary

Note: dc:language is defined by Adobe as bag Text. PRISM specifies 0 or 1 occurrences. Compromise is to use just the first entry in the unordered list.

3.4.14 dc:publisher

Name	Publisher
Identifier	dc:publisher
Definition	The entity responsible for making the resource available.
Occurrence	Occurs 0 or 1 time
Comment	The organization or individual that released the resource for publication. For magazine title use prism:publicationName.
Included in PAM?	Yes
Included in PSV?	Yes

PRISM Subset of the Dublin Core Metadata Version 3.0

Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dc:publisher>Wanderlust</dc:publisher>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	Model #1 <dc:publisher rdf:resource="http://wanderlust.com"/> Model #2 <dc:publisher>Wanderlust</dc:publisher> Model #3 <dc:publisher rdf:parseType="literal">Town & Country</dc:publisher>
Profile 3 (XMP)	
Property Value	bag ProperName, first entry (dc:publisher/*[1])

Note: dc:publisher is defined by Adobe as bag Text. PRISM specifies 0 or 1 occurrences. Compromise, use just the first entry in the unordered list.

3.4.15 dc:relation

Name	Relation
Identifier	dc:relation
Definition	A reference to a related resource.
Occurrence	Occurs 0 or more times (0 for Profile #1)
Comment	Because the notion of "related resource" is vague in Profile #1, PRISM recommends that this element not be used for Profile #1. Preference should be given to the more specific Dublin Core [DCMI-TERMS] and PRISM relationship elements [PRISMBMS] . Relation is used for Profile 2 (RDF) and Profile 3, where it is refined by more specific dcterms elements.
Included in PAM?	No
Included in PSV?	No
Profile #1 (XML)	Not recommended
Profile #2 (RDF)	
Model #1	
Element Content	Empty Node

PRISM Subset of the Dublin Core Metadata Version 3.0

Attributes	Resource Reference (rdf:resource)
Example	Model #1 <pre><dc:relation> <dcterms:isRequiredBy>dl123452345.xml</dcterms:isRequiredBy> </dc:relation></pre>
Profile 3 (XMP)	
Property Value	bag Text

Note: Refined versions of dc:relation are: dcterms:isPartOf, dcterms:hasPart, dcterms:isRequiredBy, dcterms:Requires, dcterms:IsFormatOf, dcterms:hasFormat, dcterms:isVersionOf, dcterms:hasVersion, dcterms:source, prism:hasPreviousVersion, prism:isTranslationOf, prism:hasTranslation, prism:hasAlternative, prism:isCorrectionOf.

3.4.16 dcterms:requires

Name	Requires
Identifier	dcterms:requires
Definition	The described resource requires the referenced resource, either physically or logically.
Occurrence	Occurs 0 or more times
Comment	This is the inverse of the dc:isRequiredBy relation. For profile 2, If there are multiple “requires” elements for the resource PRISM recommends listing the multiple values inside one dcterms:requires element using the RDF containers such as rdf:Bag, rdf:Seq or rdf:Alt to be XMP compatible. For profile 1, simple repeat the dcterms:requires element multiple times.
Included in PAM?	No
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dcterms:requires>dl123452345.xml</dcterms:requires>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Examples	Model #1 <pre><dcterms:requires rdf:resource="dl123452345.xml"/></pre> Model #2 <pre><dcterms:requires>dl123452345.xml</dcterms:requires></pre>
Profile 3 (XMP)	
Qualifier Value	bag Text, a refined version of dc:relation

3.4.17 dc:rights

Name	Rights
Identifier	dc:rights
Definition	Information about rights held in and over the resource.
Occurrence	Occurs 0 or 1 time
Comment	<p>The PRISM Specification provides the PRISM Usage Rights Metadata Specification [PRISMURMS] for sophisticated rights handling. Best practice is to use elements from this namespace.</p> <p>The dc:rights element, however, has been retained for those implementors who need to express simple rights and would like to utilize a single element. dc:rights is not documented in the PRISM Rights Guide and should be treated as a standalone element.</p>
Included in PAM?	No
Included in PSV?	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dc:rights>Permissions Unknown</dc:rights>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Example	<p>Model #1</p> <p><dc:rights rdf:resource="rights.xml#permissionsUnknown"/></p>
Profile 3 (XMP)	
Property Value	Lang Alt

3.4.18 dc:source

Name	Source
Identifier	dc:source
Definition	<p>A reference to a resource from which the present resource is derived. The present resource may be a performance, production, derivation, adaptation or interpretation of the referenced resource.</p> <p>The intent is to deprecate this element in favor of dcterms:source in a future release of PRISM (v3.0).</p>
Occurrence	Occurs 0 or 1 time
Comment	When possible, use a URI for an unambiguous reference to the source. Otherwise, a textual identifier of the source may be provided.
Included in PAM?	No
Included in PSV?	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dc:source>Adapted from "The River" by Bruce Springsteen</dc:source>
Profile #2 (RDF)	

PRISM Subset of the Dublin Core Metadata Version 3.0

Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Model #3	
Element Content	XML Literal
Attributes	rdf:parseType="Literal" xml:lang = (optional) designed for identifying the human language used
Examples	<p>Model #1</p> <pre><dc:source rdf:resource= "http://example.com/classics/Romeo%20and%20Juliet"/></pre> <p>Model #2</p> <pre><dc:source>Adapted from "The River" by Bruce Springsteen</dc:source></pre> <p>Model #3</p> <pre><dc:source>Adapted from The River by Bruce Springsteen</dc:source></pre>
Profile 3 (XMP)	
Property Value	Not Used in XMP, use dcterms:source

3.4.19 dcterms:source

Name	Source
Identifier	dcterms:source
Definition	A reference to a resource from which the present resource is derived. The present resource may be a performance, production, derivation, adaptation or interpretation of the referenced resource.
Occurrence	Occurs 0 or 1 time
Comment	When possible, use a URI for an unambiguous reference to the source. Otherwise, a textual identifier of the source may be provided.
Included in PAM?	No
Included in PSV?	No
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource)
Examples	<p>Model #1</p> <pre><dcterms:source rdf:resource= "http://example.com/classics/Romeo%20and%20Juliet"/></pre>
Comment	PRISM Best Practice is to reserve dcterms:source when a URI is given as the value for

	this field. dc:source should be used in other instances.
--	--

3.4.20 dc:subject

Name	Subject
Identifier	dc:subject
Definition	The main topic or topics of the content of the resource. Defines "aboutness".
Occurrence	Occurs 0 or more times
Comment	<p>Dublin Core and PRISM recommend that dc:subject be expressed as a value from a controlled vocabulary, if available.</p> <p>PRISM defines several elements for more specific types of subjects, such as when events, locations, objects, organizations and people are the subject of the resource. Those elements SHOULD be used in preference to the dc:subject element if appropriate. You may use dc:subject to indicate that such things as a concept, a disease or a theory are the subject of an article because there is no special PRISM element to cover these types of subjects.</p> <p>For profile 2, if there are multiple subjects for the resource PRISM recommends listing the multiple subjects inside one dc:subject element using the RDF containers such as rdf:Bag, rdf:Seq or rdf:Alt to be XMP compatible. For profile 1, simple repeat the dc:subject element multiple times.</p> <p>If local operations on the name(s) or definition(s) of the vocabulary elements is needed, PRISM's recommended practice is to provide the value of the dc:subject element using the pcv:Descriptor element and its allowed elements of pcv:vocab, pcv:code, and pcv:label. Remember, PRISM element types are specified in camel case. The exception is that when elements denote Classes in the sense of the RDF Schema W3C-RDFS, they must begin with an uppercase letter. The only PRISM element to do so is pcv:Descriptor.</p>
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	xml:lang = (optional) designed for identifying the human language used. prism:namespace= specifies the namespace of the subject in order to link to the appropriate subject vocabulary
Example	<dc:subject>Seasonal Affective Disorder</dc:subject>
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource) xml:lang = (optional) designed for identifying the human language used. prism:namespace= specifies the namespace of the subject in order to link to the appropriate subject vocabulary
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used prism:namespace= specifies the namespace of the subject in order to link to the appropriate subject vocabulary
Examples	<p>Model #1</p> <p><dc:subject rdf:resource="http://prismstandard.org/vocabs/lcc/QA76"/></p>

PRISM Subset of the Dublin Core Metadata Version 3.0

	Model #2 <dc:subject>Seasonal Affective Disorder</dc:subject>
Profile 3 (XMP)	
Property Value	Model #1 Full interoperability with existing XMP Property Value dc:subject bag Text (no support for prism:namespace attribute) Model #2 Use dcterms:subject property adding prism:namespace information.

3.4.21 dcterms:subject

Name	Subject
Identifier	dcterms:subject
Definition	The main topic or topics of the content of the resource. Defines “aboutness”.
Occurrence	Occurs 0 or more times
Comment	PRISM Best Practice is to reserve dcterms:source when a URI is given as the value for this field. dc:source should be used in other instances.
Included in PAM?	Yes
Included in PSV?	Yes
Profile #2 (RDF)	
Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource) xml:lang = (optional) designed for identifying the human language used. prism:namespace= specifies the namespace of the subject in order to link to the appropriate subject vocabulary

3.4.22 dc:title

Name	Title
Identifier	dc:title
Definition	The name given to the resource.
Occurrence	Occurs 0 or 1 time
Comment	Dublin Core recommends that dc:title be a name by which the resource is formally known. PRISM recommends that magazine publishers use dc:title for the headline of an article. The name of the magazine in which the article appears can be provided in the prism:publicationName element. The PRISM Specification allows titles to contain special markup characteristics. In such cases the rdf:parseType=“Literal” MUST be given.
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	xml:lang = (optional) designed for identifying the human language used.
Example	<dc:title>Man of the Year 2002</dc:title>
Profile #2 (RDF)	

PRISM Subset of the Dublin Core Metadata Version 3.0

Model #1	
Element Content	URI Reference (empty element)
Attributes	Resource Reference (rdf:resource) xml:lang = (optional) designed for identifying the human language used.
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used
Examples	<p>Model #1</p> <pre><dc:title rdf:resource="http://www.usatoday.economy"/></pre> <p>Model #2</p> <pre><dc:title>Man of the Year 2002</dc:title> <prism:publicationName>Time Magazine</prism:publicationName> <dc:publisher>Time Inc.</dc:publisher></pre>
Profile 3 (XMP)	
Property Value	Lang Alt

3.4.23 dc:type

Name	Type
Identifier	dc:type
Definition	The style of presentation of the resource's content, such as image vs. a table.
Occurrence	Occurs 0 or more times
Comment	<p>The 'type' of a resource can be many different things. In PRISM descriptions, the dc:type element takes values that indicate the style of presentation of the content, such as "Map", "Table", or "Chart". This is in contrast to prism:genre, which represents the intellectual content type, of the resource. For example, the genre 'electionResults' can be presented in a map, a table, or a chart.</p> <p>Recommended practice for PRISM implementations is to use a value from Table 8.0 [PRISMCVS] Controlled Vocabulary of Presentation Styles, expressed as a URI reference. Implementations MUST also be able to handle text values, but interoperability with text values cannot be guaranteed.</p> <p>To describe the physical size or digital file format of the resource, use the dc:format element.</p>
Included in PAM?	Yes
Included in PSV?	Yes
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	<dc:type>photo</dc:type>
Profile #2 (RDF)	
Element Content	URI Reference (empty element)
Attributes	Authority Reference (rdf:resource)
Model #2	
Element Content	Plain Literal

PRISM Subset of the Dublin Core Metadata Version 3.0

Attributes	xml:lang = (optional) designed for identifying the human language used
Examples	<p>The three examples below show how prism:type, prism:genre, and dc:format all describe different aspects of a resource. For brevity, the examples below use relative URI references. Assume that they are within the scope of a base URI declaration:</p> <p>Model #1</p> <pre><dc:type rdf:resource="resourcetype.xml#photo"/> <prism:genre rdf:resource="genre.xml#cover"/> <dc:format>image/jpeg</dc:format></pre> <p>Model #2</p> <pre><dc:type>http://idealliance.resourcetype.xml#photo/> <prism:genre rdf:resource="genre.xml#cover"/> <dc:format>image/jpeg</dc:format></pre>
Profile #3 (XMP)	
Property Value	bag Text, closed vocabulary