PRISM:

Publishing Requirements for Industry Standard Metadata

PRISM Specification: Modular: Version 2.1

PRISM Subset of the Dublin Core Namespaces

2009 05 15

Copyright and Legal Notices

Table of Contents

1 Status

1.1 Document Status

The status of this document is:

✓	Draft
✓	Released for Public Comment
✓	Released

1.2 Document Location

The location of this document is:

http://www.prismstandard.org/ specifications/2.1/PRISM_dublin_core_2.1.pdf

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 Draft A	5/30/05	Kennedy/McConnell	Enhance element descriptions and examples. Include RDF discussion as per edits to [PRISMPRISMNS]
1.3 Draft B	6/6/05	Kennedy	Resolve comments from WG Con Calls
1.3 Final	10/01/05	Kennedy	Resolve open industry comments
2.0 Draft A	06/24/07	Kennedy	Prepare WG review at F2F meeting
2.0 Draft B	07/06/07	Kennedy	Prepare for final WG Review
2.0 Final Draft	07/12/07	Kennedy	Prepare for Public Comment
2.0 FD w Edits	09/14/07	Kennedy	Prepare for Comment Resolution
2.0 Final	10/15/07	Kennedy	Final with Comments Resolved
2.0 Release	2/19/08	Kennedy	Final Release Version
2.0 w Errata	7/03/08	Kennedy	Final with Errata
2.1 Draft A	7/10/08	Kennedy	First Draft PRISM 2.1
2.1 Draft B	8/14/08	Kennedy	Second Draft of PRISM 2.1
2.1 Draft C	9/30/08	Clark	Third Draft of PRISM 2.1
2.1 Final Draft	10/14/08	Clark	Prepare for Public Comment
2.1 Final	05/15/09	Kennedy	Comments Resolved

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:

Document	Description
PRISM Introduction [PRISMINT] http://www.prismstandard.org/specifications/2.1/PRISM_introduction_2.1.pdf	Overview, background, purpose and scope of PRISM; examples; contains no normative material.
PRISM Compliance [PRISMCOMP] http://www.prismstandard.org/specifications/2.1/PRISM_compliance_2.1.pdf	Describes two profiles of PRISM compliance for content and systems; includes normative material.
The PRISM Namespace [PRISMPRISMNS] http://www.prismstandard.org/specifications/2.1/PRISM_prism_namespace_2.1.pdf	Describes the elements contained in the PRISM namespace; includes normative material.
The PRISM Subset of the Dublin Core Namespace [PRISMDCNS] http://www.prismstandard.org/specifications/2.1/PRISM_dublin_core_namespaces_2.1.pdf	Describes the elements from the Dublin Core namespace that are included in PRISM; includes normative material.
The PRISM Inline Markup Namespace [PRISMIMNS] http://www.prismstandard.org/specifications/2.1/PRISM_inline_markup_namespace_2.1.pdf	Describes the elements contained in the PRISM Inline Markup Namespace; includes normative material.
The PRISM Rights Language Namespace [PRISMRLNS] http://www.prismstandard.org/specifications/2.1/PRISM_rights_namespace_2.1.pdf	Describes the elements contained in the PRISM Rights Language Namespace; includes normative material.
The PRISM Usage Rights Namespace [PRISMURNS] http://www.prismstandard.org/specifications/2.1/PRISM_usage_rights_namespace_2.1.pdf	Describes the 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.
The PRISM Controlled Vocabulary Namespace [PRISMCVNS] http://www.prismstandard.org/specifications/2.1/	Describes the elements contained in the PRISM Controlled Vocabulary Namespace; includes normative material. The PRISM Controlled

PRISM_controlled_vocabulary_namespace_2.1.pdf	Vocabularies are now documented in this document.
The PRISM Aggregator Message Namespace [PRISMAMNS] http://www.prismstandard.org/specifications/2.1/PRISM_prism_aggregator_message_namespace_2.1.pdf	Describes the elements contained in the PRISM Aggregator Message Namespace; includes normative material.

Table 1.0: PRISM Documentation Package

2.3.1 Additional PRISM Documentation

The PRISM Aggregator Message (PAM), an XML-based application of PRISM, adds a small namespace of its own, formally described in [PRISMAMNS]. The structure and use of PAM are described separately in Guide to the PRISM Aggregator Message V. 2.1 [PAMGUIDE]. The Guide is accompanied by both an XSD and a DTD.

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

Guide to Profile 1 PRISM Usage Rights [RIGHTSGUIDE] documents an XML-based PRISM Profile 1 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.

2.3.2 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.

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 2.1 [PRISMINT] for all changes.

Changes in this document include:

- The **dcterms:source** element has been added.
- The comments for **dc:format** have been revised to indicate that PRISM strongly recommends that PRISM-compliant systems restrict values to those in the list of Internet Media Types [MIME].
- **dcterms:isPartOf** is now included in the PRISM Aggregator Message (PAM).
- A note has been added to **dc:source** to indicate that this element will be deprecated in a future release of PRISM in favor of **dcterms:source**.
- The xml:lang attribute is now allowed for **dc:description, dc:subject**, and **dc:title**
- Occurrence for **dcterms:hasPart** has been increased to 0 or more times.
- The comments for **dc:rights** have been updated to indicate that best practice is for PRISM users to utilize the PRISM Usage Rights Specification [PRISMURNS] for rights handling. If a single, standalone element is all that is desired dc:rights can be used.

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 Deprecating Redundant Elements within PRISM 2.1

PRISM is deprecated its own relations elements in favor of the Dublin Core Relations Elements upon which they were based with the release of PRISM 2.0 in 2008. See PRISM Introduction [PRISMINT] for more information.

3.5 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 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.

3.5.1 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.
PAM	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 [PRISMCVNS]
Model #1	
Element Content	String
Attributes	prism:role = value from Contributor Role Controlled Vocabulary [PRISMCVNS] prism:place = string indicating location 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	
Element Content	Plain Literal (with multiple contributors) prism:role = value from Contributor Role Controlled Vocabulary [PRISMCVNS] prism:place = string indicating location for the contributor
Attributes	rdf:resource authority reference (rdf:resource) to indicate contributor type value from PRISM Contributor Role Controlled Vocabulary

Examples	`Model #1`
	`<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>`
	
	`Model #2`
	`<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>`

Profile 3 (XMP)	
Property Value	bag ProperName

3.5.2 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
PAM	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	prism:role= value from Contributor Role Controlled Vocabulary [PRISMCVNS] prism:place= string indicating location for the contributor

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 [PRISMCVNS], expressed as a URI reference. Implementations MUST also be able to handle text values, but interoperation with text values cannot be guaranteed.
Model #1	
Element Content	Plain Literal
Attributes	Authority Reference (rdf:resource) to indicate creator type
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	seq ProperName

3.5.3 dc:description

Name	Description
Identifier	dc:description
Definition	An account of the content of the resource.
Occurrence	Occurs 0 or more times
Comment	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. 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.) 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. 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.
PAM	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	`Model #1`
	`<dc:description rdf:resource=`
	`"http://www2.rhbnc.ac.uk/Music/Archive/Disserts/attinell.html"/>`
	
	`Model #2`
	`<dc:description>Browse our catalog for the right computer for you.</dc:description>`
	
	`Model #3`
	`<dc:description rdf:parseType="Literal">Describes the infamous criminal and gunfighter, <em>Billy the Kid</em>.</dc:description>`

Profile #3 (XMP)	
Property Value	Lang Alt

3.5.4 dc:format

Name	Format
Identifier	dc:format
Definition	The physical or digital manifestation of the resource.
Occurrence	Occurs 0 or 1 time
Comment	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. 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 [MIME]. 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. 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.
PAM	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

3.5.5 dcterms:hasFormat

Name	Has Format
Identifier	dcterms:hasFormat
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	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. 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.
PAM	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	`Model #1`
	`<dcterms:hasFormat rdf:resource=“http://wap.wanderlust.com/2000/08/Belize.wml”/>`
	
	`<dcterms:hasFormat rdf:resource="doi:123/p92-1293"/>`
	
	`Model #2`
	`<dcterms:hasFormat>photo1293.jpg</dcterms:hasFormat>`
	
	`Model #3`
	`<dcterms:hasFormat rdf:parseType=”Literal”>`
	`doi&colon:123/p92&endash;1293</dcterms:hasFormat>`

Profile 3 (XMP)	Qualifier of dc:relation
Qualifier Value	Text

3.5.6 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
Comment	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. 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. 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.
PAM	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	`Model #1`
	`<dcterms:hasPart rdf:resource=`
	`  ”http://www.travelmongo.com/2000/08/BelizePhoto.jpg”/>`
	
	`Model #2`
	`<dcterms:hasPart>http://www.travelmongo.com/2000/08/BelizePhoto.jpg`
	`</dcterms:hasPart>`
	
	`Model #3`
	`<dcterms:hasPart`
	`rdf:parseType=”Literal”>dam&endash;obj&endash;32485u2</dcterms:hasPart>`

Profile 3 (XMP)	Qualifier of dc:relation
Qualifier Value	Lang Alt

3.5.7 dcterms:hasVersion

Name	Has Version
Identifier	dcterms:hasVersion
Definition	Identifies the next version of the current resource.
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 dcterms:hasFormat element. For the special case of versions known as “corrections”, use prism:hasCorrection to point from the current resource to correction blocks. Use dcterms:hasVersion to point from the corrected resource back to the earlier one.
PAM	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	`<dcterms:hasVersion>http://freeimages.com/pool.jpg</dcterms:hasVersion>`
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:hasVersion rdf:resource=`
	`“http://travelmongo.com/2000/08/BelizeTravelUpdate.xml”/>`
	
	`Model #2`
	`<dcterms:hasVersion>BelizeTravelUpdate_04.xml</dcterms:hasVersion>`
	
	`Model #3`
	`<dcterms:hasVersion rdf:parseType=”Literal”>`
	`dam&endash;obj&endash;32485u2</dcterms:hasVersion>`

Profile 3 (XMP)	Qualifier of dc:relation
Qualifier Value	Text

3.5.8 dc:identifier

Name	Identifier
Identifier	dc:identifier
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	In PRISM, dc:identifier provides a place for the unique identification of a resource. PRISM requires the use a single dc:identifier statement with a unique identifier to identify a particular published item. Other PRISM identifiers can be applied to provide further identification that is required. 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 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.
PAM	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)	
Field Value	`Text`

3.5.9 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.
PAM	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	`<dcterms:isFormatOf>Beize.pdf</dcterms:isFormatOf>`
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:isFormatOf rdf:resource=”http://wanderlust.com/2000/08/Belize.qxd”/>`
	
	`Model #2`
	`<dcterms:isFormatOf>Beize.pdf</dcterms:isFormatOf>`
	
	`Model #3`
	`<dcterms:isFormatOf rdf:parseType=”Literal”>`
	`doi&colon:123/p92&endash;1293</dcterms:isFormatOf>`

Profile 3 (XMP)	Qualifier of dc:relation
Qualifier Value	Text

3.5.10 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	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. 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.
PAM	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
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 of dc:relation
Qualifier Value	Lang Alt

3.5.11 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.
PAM	No
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>`
	
	`Model #3`
	`<dcterms:isRequiredBy rdf:parseType=”Literal”>`
	`dam&endash;obj&endash;32485u2</dcterms:isRequiredBy>`

Profile 3 (XMP)	Qualifier of dc:relation
Qualifier Value	Text

3.5.12 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.
PAM	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	`<dcterms:isVersionOf>Ovid's *Ars Amatoria*</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 of dc:relation
Qualifier Value	Text

3.5.13 dc:language

Name	Language
Identifier	dc:language
Definition	A language of the intellectual content of the resource.
Occurence	Occurs 0 or 1 time
Comment	Recommended best practice for the values of the Language element is defined by RFC 3066 [RFC3066]. It specifies the use of a two-letter (or three-letter) Language Code taken from the ISO 639 standard [ISO639] (or from ISO 639-2), optionally followed by a two-letter Country Code (taken from the ISO 3166 standard [ISO3166]). For example, 'en' for English, 'fr' for French, or 'en-GB' for English used in the United Kingdom.
PAM	No
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=http://www.din.de/gremien/nas/nabd/iso3166ma/a3ptnorm.htm/>`
	
	`Model #2`
	`<dc:language>en-US</dc:language>`

Profile 3 (XMP)	
Field Value	Locale

3.5.14 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.
PAM	Yes
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 &amp; Country</dc:publisher>`

Profile 3 (XMP)	
Field Value	ProperName

3.5.15 dc:relation

Name	Relation
Identifier	dc:relation
Definition	A reference to a related resource.
Occurence	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 [PRISM]. Relation is used for Profile 2 (RDF) and Profile 3, where it is refined by more specific dcterms elements.
PAM	No
Profile #1 (XML)	Not recommended
Profile #2 (RDF)	
Model #1	
Element Content	Empty Node
Attributes	Resource Reference (rdf:resource)

Example	`Model #1`
	`<dc:relation>`
	`  <dcterms:isRequiredBy>dl123452345.xml</dcterms:isRequiredBy>`
	`</dc:relation>`

Profile 3 (XMP)	bag Text
Field Value	bag Text, refined by Qualifiers 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.

3.5.16 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.
PAM	No
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`
	`<dcterms:requires rdf:resource=“dl123452345.xml”/>`
	
	`Model #2`
	`<dcterms:requires>dl123452345.xml</dcterms:requires>`

Profile 3 (XMP)	Qualifier of dc:relation
Qualifier Value	Text

3.5.17 dc:rights

Name	Rights
Identifier	dc:rights
Definition	Information about rights held in and over the resource.
Occurrence	Occurs 0 or 1 time
Comment	The PRISM Specification provides the PRISM Usage Rights Specification [PRISMURNS] for sophisticated rights handling. Best practice is to use elements from this namespace. 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.
PAM	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	`Model #1` `<dc:rights rdf:resource=”rights.xml#permissionsUnknown”/>`
Profile 3 (XMP)	
Field Value	Lang Alt

3.5.18 dc:source

Name	Source
Identifier	dc: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. The intent is to deprecate this element in favor of dcterms:source in a future release of PRISM (v3.0).
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.
PAM	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	`<dc:source>Adapted from "The River" by Bruce Springsteen</dc:source>`
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:source rdf:resource=`
	`“http://example.com/classics/Romeo%20and%20Juliet”/>`
	
	`Model #2`
	`<dc:source>Adapted from "The River" by Bruce Springsteen</dc:source>`
	
	`Model #3`
	`<dc:source>Adapted from <strong>The River</strong> by Bruce Springsteen</dc:source>`

Profile 3 (XMP)	
Field Value	Text

3.5.19 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. The intent is to deprecate dc:source in favor of this element in a future release of PRISM (v3.0).
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.
PAM	No
Profile #1 (XML)	
Element Content	String
Attributes	None
Example	`<dcterms:source>Adapted from "The River" by Bruce Springsteen</dcterms:source>`
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:source rdf:resource=`
	`“http://example.com/classics/Romeo%20and%20Juliet”/>`
	
	`Model #2`
	`<dcterms:source>Adapted from "The River" by Bruce Springsteen</dcterms:source>`
	
	`Model #3`
	`<dcterms:source>Adapted from <strong>The River</strong> by Bruce Springsteen</dcterms:source>`

Profile 3 (XMP)	XMP qualifier for dc:relation
Field Value	Text

3.5.20 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	Dublin Core and PRISM recommend that dc:subject be expressed as a value from a controlled vocabulary, if available. 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. 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. 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.
PAM	Yes
Profile #1 (XML)	
Element Content	String
Attributes	xml:lang = (optional) designed for identifying the human language used.
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.
Model #2	
Element Content	Plain Literal
Attributes	xml:lang = (optional) designed for identifying the human language used

Examples	`Model #1`
	`<dc:subject rdf:resource=”http://prismstandard.org/vocabs/lcc/QA76”/>`
	
	`Model #2`
	`<dc:subject>Seasonal Affective Disorder</dc:subject>`

Profile 3 (XMP)	
Field Value	bag Text

3.5.21 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.
PAM	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)	
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	`Model #1`
	`<dc:title rdf:resource=”http://www.usatoday.economy”/>`
	
	`Model #2`
	`<dc:title>Man of the Year 2002</dc:title>`
	`<prism:publicationName>Time Magazine</prism:publicationName>`
	`<dc:publisher>Time Inc.</dc:publisher>`

Profile 3 (XMP)	
Field Value	Lang Alt

3.5.22 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	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. Recommended practice for PRISM implementations is to use a value from Table 8.0 [PRISMcontrolled vocabulary] Controlled Vocabulary of Presentation Styles, expressed as a URI reference. Implementations MUST also be able to handle text values, but interoperation with text values cannot be guaranteed. To describe the physical size or digital file format of the resource, use the dc:format element.
PAM	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
Attributes	xml:lang = (optional) designed for identifying the human language used

Examples	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:
	
	`Model #1`
	`<dc:type rdf:resource=”resourcetype.xml#photo”/>`
	`<prism:genre rdf:resource=”genre.xml#cover”/>`
	`<dc:format>image/jpeg</dc:format>`
	
	`Model #2`
	`<dc:type>http://idealliance.resourcetype.xml#photo/>`
	`<prism:genre rdf:resource="genre.xml#cover"/>`
	`<dc:format>image/jpeg</dc:format>`

Profile #3 (XMP)	
Field Value	bag Text, closed vocabulary