PRISM
A Specification of IDEAlliance

PRISM:

Publishing Requirements for Industry Standard Metadata

Guide to
Contract Management Metadata

Referencing the PRISM 3.0 Specification

Version 3.1

March 27, 2015

IDEAlliance®

Table of Contents

1 STATUS

1.1 Document Status

The status of this document is:

✓	Draft	10/30/2014
✓	Released for Public Comment	11/18/2014
✓	Final Specification	03/27/2015

1.2 Document Location

The location of this document is:

http://www.prismstandard.org/specifications/3.1/Contract_Guide_3.1.pdf

or

http://www.prismstandard.org/specifications/3.1/Contract_Guide_3.1.htm

1.3 Version History

Version Number	Release Date	Editor	Description
Draft 1.0A	October 30, 2014	Kennedy	First Draft Guide
Draft 1.0B	November 18, 2014	Kennedy	Second Draft
V3.1	March 27, 2015	Kennedy	Final Release

2 PRISM CONTRACT MANAGEMENT GUIDE STRUCTURE

This PRISM Contract Guide was written to provide a guide to the expression of PRISM Contract Management Metadata. This guide will supplementary to the PRISM3.1 Specification. The Specification is the authoritative source for all issues of concern.

The PRISM Contract Management Guide is made up of:

1. Status: Status of the document

2. PRISM Contract Management Guide Structure: A guide to the structure of the document

3. Introduction: A brief overview of the PRISM Contract Management XML Schema and how this schema links to PRISM Usage Rights XML and PRISM Source Vocabulary for Asset Management.

4. XML Contract Management Expression: An overview of an XML structured contract management expression for PRISM Profile 1

5. PRISM Contract Management Metadata Elements

5.1 Elements in Alphabetical Order: A listing of all elements in the PRISM Contract Management Metadata fields in an XML Format in alphabetical order.

5.3 Element Glossary: A list of all elements in the PRISM Contract Management XML elements with their definition.

Two Appendices are also included in the PRISM Contract Management Elements Guide. These include:

Appendix A: PRISM Contract Management Model XSD

Appendix B: Contract Management Tagging Example

3 INTRODUCTION

The PRISM Contract Management XML format is a standardized XML schema structure for publishers to use to describe contracts for archive, management and interrogation by a Digital Rights Management, Asset Management or Contract Management System.

3.1 The Use Case for Contract Management Metadata

The PRISM Contract Management Metadata schema was developed with a very specific use case in mind:

Title	Develop Metadata for Managing Contracts as Assets
Summary:	Publishers have a requirement to move contracts from file cabinet management to a database contract management system.
Scope:	This use case strictly addresses managing contracts and legal documents as assets throughout the contracting workflow. Linking legal documents to the assets management system is out of scope. Automating summary reporting for usage rights management is out-of-scope.
Actors:	Legal, permissions staff / paralegal staff
Prerequisites:	1. Have an accounting of the current contract types, clauses and common terminology 2. Establish a standard vocabulary (terms/definitions) to be used for contract management
Scenario:	1. A contracts database will be established to manage contracts and other legal documents including printer agreements, assignment and scheduling agreements. 2. Legal documents will be organized at the discretion of the publisher. 3. Legal documents will be tracked from inception through completion. 4. The ID of each Legal document will be linked to one or more assets when appropriate. 5. General legal documents for any project may also be filed in this system.

Requirements	1. Organizational criteria for the contracts management database *must* include:
	   a. A unique contract number for each contract
	   b. The unique ID, name and contact information of the contracting organization (i.e. publisher, department or brand)
	   c. The unique ID, name and contact information of the supplier / agent
	
	2. Organizational criteria for the contracts management database *may* include:
	   a. The date the contract was executed
	   b. Dates for contract milestones (i.e. workflow tracking)
	   c. The name and contact information of the rights owner

3. Document Versioning and Status will be tracked by the system.
4. If usage rights are specified by a contract, those rights will be captured using a standard set of PRISM rights summary metadata fields
5. When legal documents, such as assignment and scheduling agreements refer to other legal documents, such as contracts and releases, those links must be captured

3.2 The PRISM Contract Management Metadata Namespace

As with all PRISM Specifications, the fields for Contract Management were developed in a separate namespace. This model enables users to select which ever PRISM modules are required to meet their unique business needs.

The namespace for PRISM Contract Management Metadata is pcmm: This namespace will be added to PRISM as of the publication of PRISM V3.1.

```
xmlns:pcmm="http://prismstandard.org/namespaces/pcmm/3.1"
```

3.3 Where Contract Management Fits with Rights Management

Contract Management metadata is one of 3 distinct, yet interrelated, metadata sets that enable the construction of a comprehensive rights management solution. Links must be made between these 3 domains. See Figure 3.1.
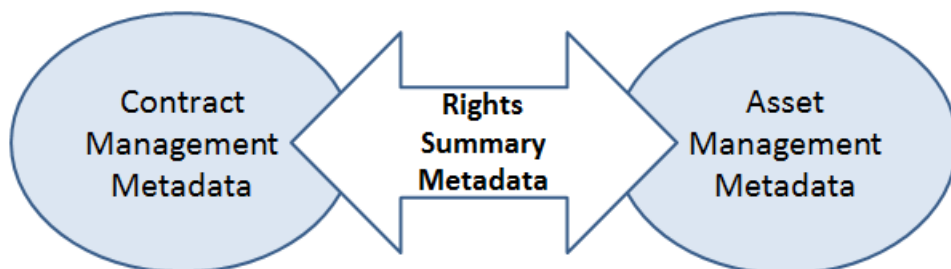
Figure 3.1 Metadata domains required to support comprehensive Rights Management

3.4 About the PRISM Contract Management XML Model

The standard PRISM Contract Management Metadata fields and their values are documented in the PRISM Contract Management Metadata Specification [PCMMS] In addition to developing a series of standard metadata fields to capture and describe contract content, the PRISM Working Group has developed a standardized XML model. This model captures the relationship between the fields that make up PRISM Contract Management Metadata and is documented in this Guide.

3.5 Advantages of Using the PRISM Contract Management XML Model

This expression of the PRISM Contract Management metadata elements is a standardized XML model for PRISM Profile 1. Adapting your processes to conform to the PRISM Contract Management XML format will provide many advantages and financial benefits to you and your business partners.

- The use of a single format for all organizations using XML for contract description will speed the processing of usage rights expressed in a contract and fosters easy integration with tools and partners within a rights management workflow.

- The use of a common industry XML model reduces the barrier to entry for all publishers and content aggregators wishing to shift from management of contracts on paper to digitized management and integration with rights management and asset management systems.

- The PRISM Contract Management XML model helps everyone in the electronic content business to track, access and use rights information contained within asset-related contracts.

- The PRISM Contract Management XML model provides an excellent starting point for those implementing a comprehensive digital rights management system.

- The PRISM Contract Management XML model provides an excellent starting point for those who wish to automate the red-light/green-light right-to-use decision using ODRL.

These advantages will enable you use your contract-based content on a wider variety of output media and products, getting more value from your information assets.

3.6 Legend for Diagrams

In this guide, the XML model is often illustrated by a model diagram. Each diagram was produced with the Altova XML Spy product. These diagrams show the elements and attributes that make up a model and their order and frequency.

The legend for reading XML model diagrams is shown in Figure 1.1. Elements that are required by the model are shown in a solid box. Elements that are optional are shown in a dotted box. Likewise attributes may be required (solid box) or optional (dotted box). A repeatable occurrence of elements is indicated by numbers below each element box to the right.

The diagrams also indicate how elements are assembled. When building some models, elements may occur in a sequence with a specified order. Other models provide a choice from among a number of elements. The legend in Figure 3.2 shows the connectors for sequence and choice.
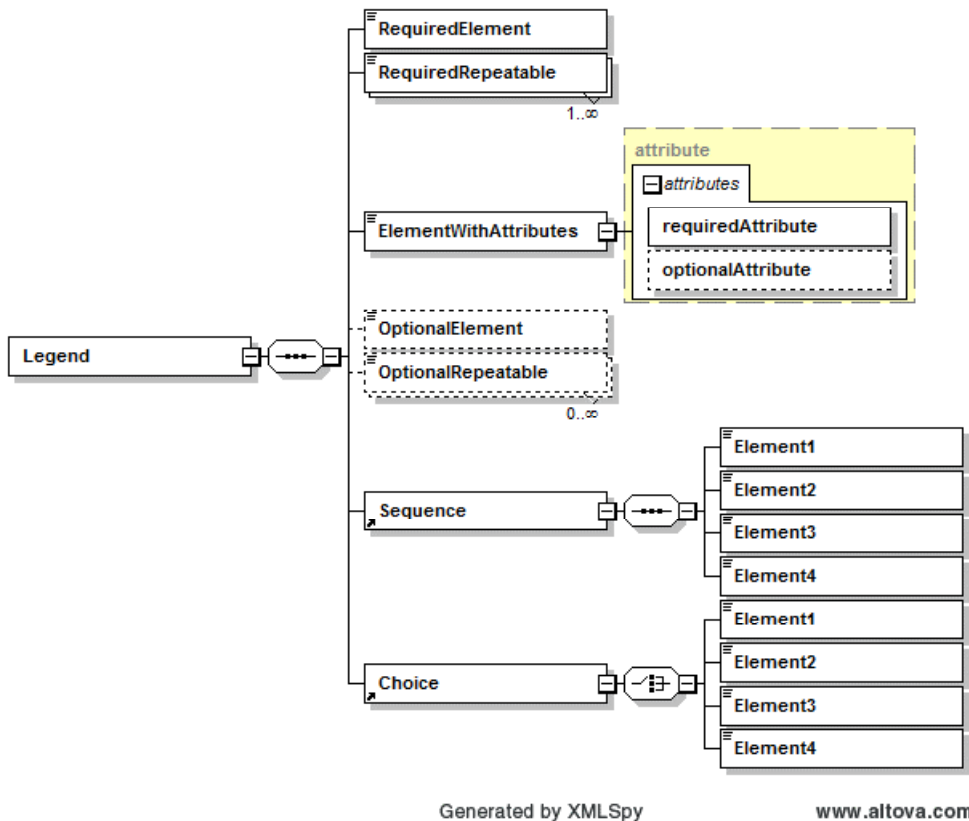
Figure 3.2 Legend for XML Diagrams

4 PRISM CONTRACT MANAGEMENT EXPRESSION XML MODEL

In order to provide an easy to implement XML model for PRISM Contracts Management, an XML schema has been constructed to standardize the order and frequency of PRISM Contract elements. Benefits of a standardized, structured model are discussed in Section 3 of this document.

> **NOTE:** PRISM Contract Management is designed to be implemented in an international environment. To facilitate multilingual deployment, all text / string fields employ the W3C xml:lang attribute so a language can be specified.

4.1 PRISM Contracts Metadata Fields

The PRISM Contracts Container was constructed to serve as a _virtual container_ within which all contract elements can be organized by sequence and frequency. See Figure 4.1.
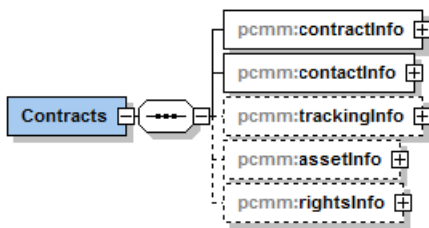
Figure 4.1 Contracts Container Structure

A virtual grouping of contract elements that is _required_ is pictured with a solid outline, such as <contractInfo and <contactInfo. Optional element groups are pictured with a dashed outline.

4.2 Contract Information

The first grouping of contract elements, <contractInfo, is required because it contains some metadata fields that are required to manage contracts as assets. (See the Use Case). These fields are shown in Figure 4.2.
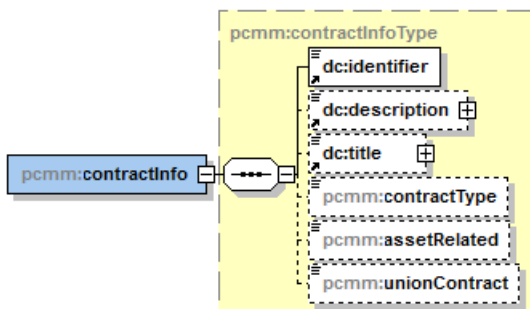
Figure 4.2 Contract Information

The first metadata field in this virtual element grouping is <dc:identifier. In order to manage contracts in a DAM or database, each contract must have a unique identifier. For clarity, set the id= attribute to "contractID". Other fields that some publishers find useful include:

- An optional dc:description
- An optional title that may be expressed as dc:title with id="contractTitle" specified for clarity
- An optional contract type (publisher defined)
- A optional field that flags whether the contract is asset related or a general contract. If assetRelated = "true" then information in this contract is key to managing the usage rights for an asset in the publisher's DAM. Default for this field is true.

- An optional true/false flag that indicates whether this contract is a union contract.

> **Note:** The metadata fields that make up contract information are documented in Chapter 5 of this Guide and in the PRISM Contract Management Metadata Specification [PRISMCMMS].

4.3 Contact Information

Two contract metadata fields that document contracting contacts are required by this XML model. These contacts are responsible from initiation of a contract through signing and filing the contract in the contract management system. See Figure 4.3.
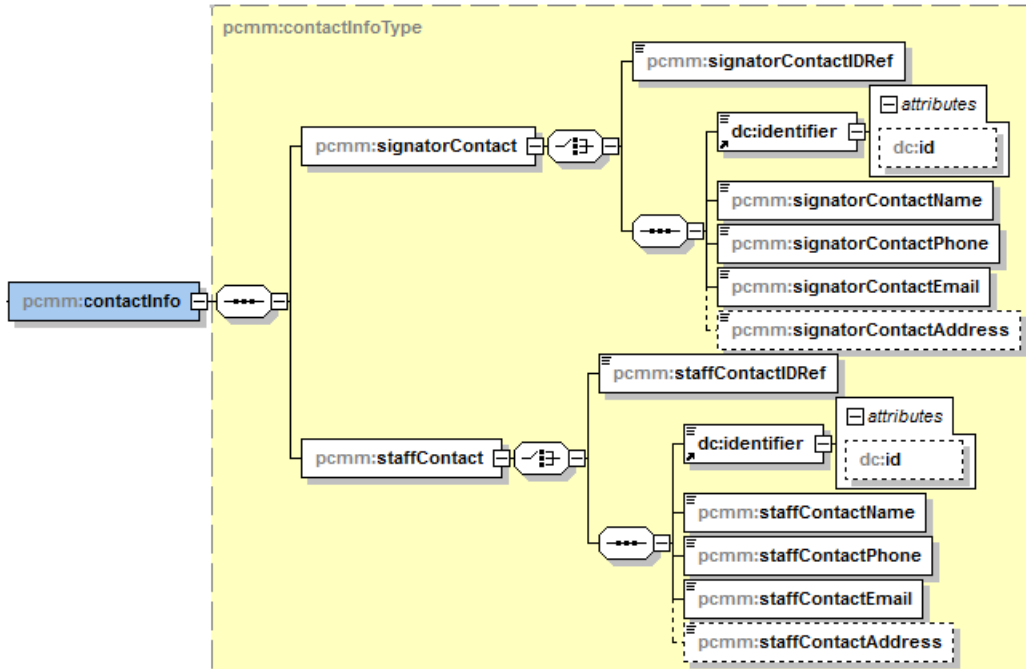
Figure 4.3 Contact Information Metadata

The metadata about contracting parties that must be provided in order to manage a contract throughout the contracting workflow is the identification of the signator or signer of the contract and the staff person that is managing the contract on the publisher side. This model allows for either direct entry of the party's name, phone, email and optional address or simply providing an ID for the party that links to another system that manages contact information.

> **Note:** Other parties may be mentioned in the content of the contract. An example is an asset creator that must be credited when the asset is used in a publication. Those parties that relate directly to asset usage are captured as part of the Usage Rights summary metadata. Only the parties that must be directly referenced during the workflow and management of a contract are included in this model.

4.4 Contract Tracking Information

If the publisher plans to track contracting workflow as outlined in the Contract Management Use Case, a group of metadata elements to enable tracking are provided. See Figure 4.4.
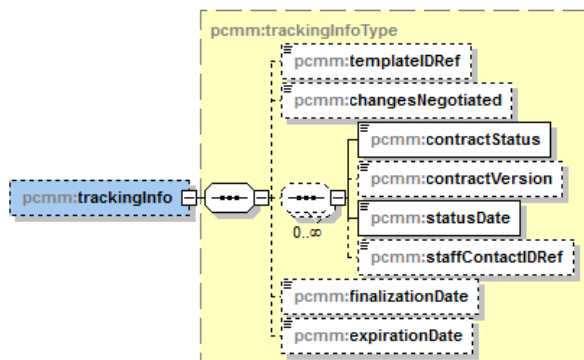
Figure 4.4 Contract Tracking Information

4.4.1 Contract Templates

Many publishers are moving toward standardizing contracts through the use of standard contract templates. The ability to use templates is directly related to the type of content being acquired and the nature of the publication. If contract templating is being employed, two fields capture information about the template:

- pcmm:templateIDRef references the template used with an ID of the template that was used
- pcmm:changesNegotiated provides information about any changes that were made to the template for this contract

4.4.2 Contract Status Tracking

If a publisher intends to track versions of a contract through the negotiation workflow, then the metadata fields in the status block may be used. Note that this block can occur 0 to many times. The enables the capture of every stage of the contract development workflow. Contract status fields include:

- pcmm:contractStatus: Indicates the current status of a contract. A list of standard values, or controlled vocabulary is provided.
- pcmm: contractVersion: If the changes negotiated are very significant, a new version of the contract may be developed and noted in this field.
- pcmm:statusDate: This field can be used to capture the date that a new contract status was reached.
- pcmm:staffContactIDRef: Since a different staff person may oversee different stages of the contract workflow, a reference to the staff person that signed off on this contract status may be noted.

4.4.3 Contract Dates

Two date fields can be used to capture when the contract was finalized and the expiration date of the contract, if such a date exists.

4.5 Asset Information

If a contract is related to acquisition and use of media assets or text content, then many publishers believe there should be the ability to capture some information about the asset in the contract management system. These fields are optional. See Figure 4.5.
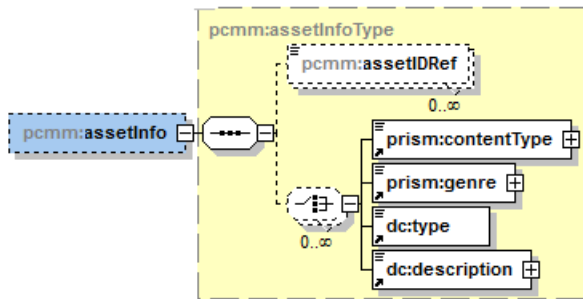
Figure 4.5 Asset Information

The <assetIDRef is a link to an asset in a DAM system. Other information that may be useful in a contract management database include a choice of the PRISM <contentType and <genre, the Dublin Core <type and the Dublin Core <description.

4.6 Rights Summary Information

Typically the contract management metadata will be linked to metadata about the usage rights granted within the contract. This may be either within the contract management system, within the asset management system (or DAM) or in a rights management system. This link is provided by the Usage Rights Summary Information element. See Figure 4.6.

Figure 4.6 Usage Rights Summary Information

5 PRISM CONTRACT MANAGEMENT METADATA ELEMENTS

5.1 Elements in Alphabetical Order

The following is an alphabetical list of the Contract Management metadata that are documented in the *PRISM Contract Management Metadata Specification* [PCMMS].

- pcmm:assetIDRef
- pcmm:assetRelated
- pcmm:changesNegotiated
- prism:contentType
- pcmm:contractStatus
- pcmm:contractType
- pcmm:contractVersion
- dc:description
- pcmm:expirationDate
- pcmm:finalizationDate
- prism:genre
- dc:identifier
- pcmm:signatorContactAddress
- pcmm:signatorContactEmail

- pcmm:signatorContactIDRef
- pcmm:signatorContactName
- pcmm:signatorContactPhone
- pcmm:staffContactAddress
- pcmm:staffContactEmail
- pcmm:staffContactIDRef
- pcmm:staffContactName
- pcmm:staffContactPhone
- pcmm:statusDate
- pcmm:templateIDRef
- dc:title
- dc:type
- pcmm:unionContract
- pcmm:usageRightsSummaryIDRef

5.2 PRISM Contract Management Element Glossary

The following is glossary of definitions for the Contract Management metadata elements in the pcmm: namespace.

assetIDRef: A link to the asset in an asset management system.

assetRelated: A true/false flag to indicate that this contract is related to the acquisition of a content or media asset(s).

changesNegotiated: A true/false flag to indicate that changes from the standard template have been negotiated.

contractStatus: Specifies the status of the contract at the time the version is stored in the system. Standard values include: draft, being negotiated, finalized, uploaded, upload approved, upload rejected -- try again.

contractType: A publisher-defined contract type to be used to organize contracts into logical groups. Examples are "Work for Hire" "World Wide Rights" and "All Rights" or "Contract" "Release" and "Assignment".

contractVersion: Version number of a contract so major changes resulting in a new version can be tracked.

expirationDate: The date or dateTime the contract lapses or expires.

finalizationDate: The date or dateTime the contract is signed and approved by the publisher and the signator.

signatorContactAddress: The street address of the party signing the legal document.

signatorContactEmail: The email of the party signing the legal document.

signatorContactIDRef: The unique ID of the party signing the legal document.

signatorContactName: The name of the party signing the legal document.

signatorContactPhone: The phone contact of the party signing the legal document.

staffContactAddress: The address of the staff person managing contract finalization.

staffContactEmail: The email of the staff person managing contract finalization.

staffContactIDRef: The unique ID of the staff person managing contract finalization.

staffContactName: The name of the staff person managing contract finalization.

staffContactPhone: The phone contact of the staff person managing contract finalization.

statusDate: The date or dateTime that each status change is reported.

templateIDRef: Link to a standard contract template that was used to construct this contract.

unionContract: A true/false flag to indicate the contract must adhere to provisions mandated by a union.

usageRightsSummaryIDRef: A link to the usage rights summary.

Appendix A   PRISM CONTRACT MANAGEMENT MODEL XSD

The PRISM Working Group has developed a standardized XML model for Contract Management Metadata. This model captures the relationship between the fields that make up PRISM Contract Management Metadata.

```xml
<?xml version = "1.0" encoding = "UTF-8"?>


<!-- +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        PRISM CONTRACT MANAGEMENT: XSD
     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:pcmm="http://prismstandard.org/namespaces/pcmm/3.1"
targetNamespace="http://prismstandard.org/namespaces/pcmm/3.1"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:prism="http://prismstandard.org/namespaces/basic/3.0/"
elementFormDefault="qualified"
attributeFormDefault="qualified"
version="03272015">

       <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="xml.xsd"/>
       <xs:import namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
schemaLocation="rdf.xsd"/>
       <xs:import namespace="http://purl.org/dc/elements/1.1/" schemaLocation="dc.xsd"/>
       <xs:import namespace="http://prismstandard.org/namespaces/basic/3.0/"
schemaLocation="prism.xsd"/>


<!-- +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ -->

<!--===                                                              ===-->
<!--===                            ELEMENT DEFINITIONS        ===-->
<!--===                                                              ===-->


<!--===  Virtual Root Container for all PRISM Contract Management Elements for the Contracts
Guide ===-->
       <xs:element name = "Contracts">
               <xs:complexType>
                       <xs:sequence>
                               <xs:element name = "contractInfo" type = "pcmm:contractInfoType"
><xs:annotation><xs:documentation>Required Contract Information to facilitate contract
management</xs:documentation></xs:annotation></xs:element>
                               <xs:element name = "contactInfo" type = "pcmm:contactInfoType"
><xs:annotation><xs:documentation>Contact information for the staff and signator involved in
the contracting process.</xs:documentation></xs:annotation></xs:element>
                               <xs:element name = "trackingInfo" type = "pcmm:trackingInfoType"
minOccurs="0"><xs:annotation><xs:documentation>Optional Workflow Tracking
metadata</xs:documentation></xs:annotation></xs:element>
                               <xs:element name = "assetInfo" type = "pcmm:assetInfoType"
minOccurs="0"><xs:annotation><xs:documentation>Information about the Asset(s) specified in by
Contract</xs:documentation></xs:annotation></xs:element>
```

```
                    <xs:element name = "rightsInfo" type = "pcmm:rightsInfoType"
minOccurs="0"><xs:annotation><xs:documentation>Rights Summary for the Asset(s) specified in
the Contract</xs:documentation></xs:annotation></xs:element>
                    </xs:sequence>
        </xs:complexType>
    </xs:element>


    <!-- Contract Information  -->

    <xs:complexType name="contractInfoType">
            <xs:sequence>
                    <xs:element ref="dc:identifier"><xs:annotation><xs:documentation>A unique
identifier for the contract.  The id= value should be set to contractID for
clarity.</xs:documentation></xs:annotation></xs:element>
                    <xs:element ref = "dc:description" minOccurs = "0"
><xs:annotation><xs:documentation>A text description of the
contract</xs:documentation></xs:annotation></xs:element>
                    <xs:element ref="dc:title" minOccurs="0"
><xs:annotation><xs:documentation>A human-readable identifier (title) for the contract to
facilitate search/retrieval.</xs:documentation></xs:annotation></xs:element>
                    <xs:element name="contractType" type="xs:string" minOccurs = "0"
><xs:annotation><xs:documentation>A publisher-defined contract type to be used to organize
contracts into logical groups. Examples are "Work for Hire" "World Wide Rights" and "All
Rights" or "Contract" "Release" and
"Assignment".</xs:documentation></xs:annotation></xs:element>
                    <xs:element name="assetRelated" type="xs:boolean" minOccurs="0"
default="true"><xs:annotation><xs:documentation>A true/false flag to indicate that this
contract is related to the acqusition of a content or media asset(s).  The default is
"true".</xs:documentation></xs:annotation></xs:element>
                    <xs:element name="unionContract" type="xs:boolean"
minOccurs="0"><xs:annotation><xs:documentation>A true/false flag to indicate the contract
must adhere to provisions mandated by a union.
</xs:documentation></xs:annotation></xs:element>
            </xs:sequence>
    </xs:complexType>


    <!-- Contract Tracking Information    -->

    <xs:complexType name="trackingInfoType">
            <xs:sequence>
                    <xs:element name="templateIDRef" type="xs:anyURI"
minOccurs="0"><xs:annotation><xs:documentation>Link to a standard contract template that was
used to construct this contract.</xs:documentation></xs:annotation></xs:element>
                    <xs:element name="changesNegotiated" type="xs:boolean"
minOccurs="0"><xs:annotation><xs:documentation>A true/false flag to indicate that changes
from the standard template have been
negotiated.</xs:documentation></xs:annotation></xs:element>
                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                            <xs:element name = "contractStatus"
type="pcmm:contractStatusType" ><xs:annotation><xs:documentation>Specifies the status of the
contract at the time the version is stored in the system.  Standard values include: draft,
being negotiated, finalized, uploaded, upload approved, upload rejected -- try again
</xs:documentation></xs:annotation></xs:element>
                            <xs:element name="contractVersion" type="xs:string"
minOccurs="0"><xs:annotation><xs:documentation>Version number of a contract so major changes
resulting in a new version can be tracked.</xs:documentation></xs:annotation></xs:element>
                            <xs:element name = "statusDate" type =
"pcmm:dateOrDateTime" ><xs:annotation><xs:documentation>The date each status change is
reported.</xs:documentation></xs:annotation></xs:element>
```

```
                                    <xs:element name = "staffContactIDRef" type="xs:anyURI"
minOccurs="0"><xs:annotation><xs:documentation>The unique ID of the staff person managing
this contract stage.</xs:documentation></xs:annotation></xs:element>
                            </xs:sequence>
                            <xs:element name = "finalizationDate" type="pcmm:dateOrDateTime"
minOccurs="0" ><xs:annotation><xs:documentation>The date the contract is signed and approved
by the publisher and the signator.</xs:documentation></xs:annotation></xs:element>
                            <xs:element name = "expirationDate" type="pcmm:dateOrDateTime"
minOccurs="0" ><xs:annotation><xs:documentation>The date the contract lapses or
expires.</xs:documentation></xs:annotation></xs:element>
                    </xs:sequence>
            </xs:complexType>


            <!-- Contact Information   -->
            <xs:complexType name="contactInfoType">
                    <xs:sequence>
                            <xs:element name = "signatorContact" >
                                    <xs:complexType>
                                            <xs:annotation><xs:documentation>The party signing the
legal document.</xs:documentation></xs:annotation>
                                    <xs:choice minOccurs = "1">
                                            <xs:element name = "signatorContactIDRef" type="xs:anyURI">
                                                    <xs:annotation><xs:documentation>The unique ID of the
party signing the legal document.</xs:documentation></xs:annotation>
                                            </xs:element>
                                            <xs:sequence>
                                                    <xs:element ref="dc:identifier"
><xs:annotation><xs:documentation>A unique identifier for party signing the legal document
with id= set to "sigantorContractID" for
clarity.</xs:documentation></xs:annotation></xs:element>
                                                    <xs:element name = "signatorContactName"
type="xs:string" ><xs:annotation><xs:documentation>The name of the party signing the legal
document.</xs:documentation></xs:annotation></xs:element>
                                                    <xs:element name = "signatorContactPhone"
type="xs:string" ><xs:annotation><xs:documentation>The phone contact of the party signing the
legal document.</xs:documentation></xs:annotation></xs:element>
                                                    <xs:element name = "signatorContactEmail"
type="xs:string" ><xs:annotation><xs:documentation>The email of the party signing the legal
document.</xs:documentation></xs:annotation></xs:element>
                                                    <xs:element name = "signatorContactAddress"
type="xs:string" minOccurs = "0" ><xs:annotation><xs:documentation>The street address of the
party signing the legal document.</xs:documentation></xs:annotation></xs:element>
                                            </xs:sequence>
                                    </xs:choice>
                                    </xs:complexType>
                            </xs:element>
                            <xs:element name = "staffContact" >
                                    <xs:complexType>
                                            <xs:annotation><xs:documentation>The staff person signing
the legal document.</xs:documentation></xs:annotation>
                                    <xs:choice minOccurs = "1">
                                            <xs:element name = "staffContactIDRef" type="xs:anyURI"
><xs:annotation><xs:documentation>The unique ID of the staff person managing contract
finalization.</xs:documentation></xs:annotation></xs:element>
                                            <xs:sequence>
                                                    <xs:element ref="dc:identifier"
><xs:annotation><xs:documentation>A unique identifier for party signing the legal document
```

17

```
with id= set to "staffContractID" for
clarity.</xs:documentation></xs:annotation></xs:element>
                                    <xs:element name = "staffContactName"
type="xs:string"  ><xs:annotation><xs:documentation>The name of the staff person managing
contract finalization.</xs:documentation></xs:annotation></xs:element>
                                    <xs:element name = "staffContactPhone"
type="xs:string"  ><xs:annotation><xs:documentation>The phone contact of the staff person
managing contract finalization.</xs:documentation></xs:annotation></xs:element>
                                    <xs:element name = "staffContactEmail"
type="xs:string"  ><xs:annotation><xs:documentation>The email of the staff person managing
contract finalization.</xs:documentation></xs:annotation></xs:element>
                                    <xs:element name = "staffContactAddress"
type="xs:string" minOccurs = "0" ><xs:annotation><xs:documentation>The address of the staff
person managing contract finalization.</xs:documentation></xs:annotation></xs:element>
                               </xs:sequence>
                           </xs:choice>
                           </xs:complexType>
                   </xs:element>
             </xs:sequence>
       </xs:complexType>


       <!-- Asset Information    -->
       <xs:complexType name="assetInfoType">
             <xs:sequence>
                          <xs:element name="assetIDRef" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded" ><xs:annotation><xs:documentation>A link to the asset in an asset
management system</xs:documentation></xs:annotation></xs:element>
                          <xs:choice minOccurs="0" maxOccurs="unbounded" >
                                 <xs:element ref="prism:contentType"
><xs:annotation><xs:documentation>Prism:contentType is the highest-level indicator of the
nature of an asset. Sample content types include article, blogEntry and
bookChapter.</xs:documentation></xs:annotation></xs:element>
                                 <xs:element ref="prism:genre"
><xs:annotation><xs:documentation>Prism:genre specifies the intellectual content of the asset
and can be used to refine prism:contentType.  Sample genres include obituaries, biographies
and reviews. When multiple genres are specified, Best Practice is to list genre from most
inclusive to most specific in sequential
order.</xs:documentation></xs:annotation></xs:element>
                                 <xs:element ref="dc:type"
><xs:annotation><xs:documentation>Dc:type defines the stereotypical form of the resource.
Sample types include audio, video, photo, slideshow or
animation.</xs:documentation></xs:annotation></xs:element>
                                 <xs:element ref="dc:description"
><xs:annotation><xs:documentation>This is a narative text field that can be used to describe
the asset or content.</xs:documentation></xs:annotation></xs:element>
                          </xs:choice>
             </xs:sequence>
       </xs:complexType>


       <!-- Rights Information    -->
       <xs:complexType name="rightsInfoType">
             <xs:sequence>
                          <xs:element name = "usageRightsSummaryIDRef" type = "xs:anyURI"
minOccurs = "0" ><xs:annotation><xs:documentation>A link to the usage rights
summary.</xs:documentation></xs:annotation></xs:element>
             </xs:sequence>
```

18

```
          </xs:complexType>
     <!--===                                    ===-->
     <!--===           COMPLEX TYPES          ===-->
     <!--===                                    ===-->


     <!-- Status Type for Contracts   -->
          <xs:simpleType name = "contractStatusType">
               <xs:restriction base = "xs:string">
                    <xs:enumeration value = "draft"/>
                    <xs:enumeration value = "beingNegotiated"/>
                    <xs:enumeration value = "final"/>
                    <xs:enumeration value = "uploaded"/>
                    <xs:enumeration value = "uploadApproved"/>
                    <xs:enumeration value = "uploadRejected"/>
               </xs:restriction>
          </xs:simpleType>


     <!--===                                    ===-->
     <!--===           SIMPLE TYPES           ===-->
     <!--===                                    ===-->
     <!--simpleType dateOrDateTime for optional time-->
     <xs:simpleType name="dateOrDateTime">
          <xs:union memberTypes="xs:date xs:dateTime"/>
     </xs:simpleType>


     <!--===                                    ===-->
     <!--===              ATTRUBUTES          ===-->
     <!--===                                    ===-->

<!--===                          ===-->
<!--===     Agreement Status Type          ===-->
<!--===                          ===-->
          <xs:simpleType name = "statusType">
               <xs:restriction base = "xs:string">
                    <xs:enumeration value = "limited"/>
                    <xs:enumeration value = "unlimited"/>
                    <xs:enumeration value = "unknown"/>
                    <xs:enumeration value = "NA"/>
                    <xs:enumeration value = "none"/>
               </xs:restriction>
          </xs:simpleType>


<!--===                          ===-->
<!--===          Distribution Channel      ===-->
<!--===                          ===-->

     <xs:attribute name="distributionChannel" type="xs:string"/>



<!--===                          ===-->
```

```
<!--===   Platform Attribute           ===-->
<!--===                                ===-->
           <xs:attribute name="platform">
                   <xs:simpleType>
                           <xs:restriction base="xs:NMTOKEN">
                                   <xs:enumeration value="email"/>
                                   <xs:enumeration value="mobile"/>
                                   <xs:enumeration value="broadcast"/>
                                   <xs:enumeration value="web"/>
                                   <xs:enumeration value="print"/>
                                   <xs:enumeration value="recordableMedia"/>
                                   <xs:enumeration value="other"/>
                           </xs:restriction>
                   </xs:simpleType>
           </xs:attribute>


<!--===                                  ===-->
<!--===          Prohibit            ===-->
<!--===                                  ===-->
           <xs:attribute name="prohibit">
                   <xs:simpleType>
                           <xs:restriction base="xs:NMTOKEN">
                                   <xs:enumeration value="yes"/>
                                   <xs:enumeration value="no"/>
                           </xs:restriction>
                   </xs:simpleType>
           </xs:attribute>




<!--===                                  ===-->
<!--===     Agreement ID Attribute        ===-->
<!--===                                  ===-->
           <xs:attribute name="agreementID" type="xs:string"/>

<!--===                                  ===-->
<!--===     Horizontal Dimension Attribute   ===-->
<!--===                                  ===-->
           <xs:attribute name="horizontalDimension" type="xs:string"/>

<!--===                                  ===-->
<!--===     Vertical Dimension Attribute     ===-->
<!--===                                  ===-->
           <xs:attribute name="verticalDimension" type="xs:string"/>

<!--===                                  ===-->
<!--===          YesNo Type          ===-->
<!--===                                  ===-->
           <xs:simpleType name="yesNo">
                   <xs:restriction base="xs:NMTOKEN">
```

```
                            <xs:enumeration value="yes"/>
                            <xs:enumeration value="no"/>
                </xs:restriction>
            </xs:simpleType>


</xs:schema>
```

Appendix B CONTRACT MANAGEMENT TAGGING EXAMPLE

This appendix contains an example of the expression of PRISM Contract Management in the XML Format.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<pcmm:Contracts xmlns:pcmm="http://prismstandard.org/namespaces/pcmm/3.1"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:prism="http://prismstandard.org/namespaces/basic/3.0/"
 xmlns:html5="http://www.w3.org/1999/xhtml"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation=" contractManagement.xsd">
    <pcmm:contractInfo>
        <dc:identifier>identifier0</dc:identifier>
        <dc:title>NFL Video October 26, 2014</dc:title>
        <pcmm:contractType>work for hire</pcmm:contractType>
        <pcmm:assetRelated>true</pcmm:assetRelated>
    </pcmm:contractInfo>
    <pcmm:contactInfo>
        <pcmm:signatorContactIDRef>ContactID00976
        </pcmm:signatorContactIDRef>
        <pcmm:staffContactIDRef> ContactID202764
        </pcmm:staffContactIDRef>
    </pcmm:contactInfo>
    <pcmm:assetInfo>
        <pcmm:assetIDRef>DAM123K14</pcmm:assetIDRef>
        <dc:type>video</dc:type>
        <dc:description>video clip of 3 NHL quarterbacks on October 16,
2014</dc:description>
    </pcmm:assetInfo>
    <pcmm:rightsInfo>
        <pcmm:usageRightsSummaryIDRef>DRM90833
        </pcmm: usageRightsSummaryIDRef>
    </pcmm:rightsInfo>
</pcmm:Contracts>
```