AdsML® Framework for E-Commerce Business Standards for Advertising

AdsMLProofOfPublication 1.5.0
Part 2
Specification & Schema

Document Authors: AdsML Technical Working Group

Document ID: AdsMLProofOfPublication-1.5.0-SpecP2Schema-AS-1

Document File Name: AdsMLProofOfPublication-1.5-SpecP2Schema-AS.pdf

Document Status: Approved Specification

Document Date: 15 April 2010

Draft Number: 1

Table of Contents

1 AdsMLProofOfPublication Standard Documentation

1.1 Document status and copyright

This is the Approved Specification of the *AdsMLProofOfPublication 1.5 Part 2 Specification & Schema*.

1.2 Non-Exclusive License Agreement for AdsML Consortium Specifications

such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the AdsML Consortium, except as needed for the purpose of developing AdsML specifications, in which case the procedures for copyrights defined in the AdsML Process document must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by AdsML or its successors or assigns.

4. Any use, duplication, distribution, or exploitation of the Specification in any manner is at your own risk.

5. 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 OTHER PRODUCT OR SERVICE PRODUCED OR SPONSORED BY THE ADSML CONSORTIUM. THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN AND INCLUDED IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE OF THE ADSML CONSORTIUM IS PROVIDED ON AN "AS IS" BASIS. THE ADSML CONSORTIUM 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. NEITHER THE ADSML CONSORTIUM NOR ITS CONTRIBUTORS SHALL BE HELD LIABLE FOR ANY IMPROPER OR INCORRECT USE OF INFORMATION. NEITHER THE ADSML CONSORTIUM NOR ITS CONTRIBUTORS ASSUME ANY RESPONSIBILITY FOR ANYONE'S USE OF INFORMATION PROVIDED BY THE ADSML CONSORTIUM. IN NO EVENT SHALL THE ADSML CONSORTIUM OR ITS CONTRIBUTORS BE LIABLE TO ANYONE FOR DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, COMPENSATORY DAMAGES, LOST PROFITS, LOST DATA OR ANY FORM OF SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY KIND WHETHER BASED ON BREACH OF CONTRACT OR WARRANTY, TORT, PRODUCT LIABILITY OR OTHERWISE.

6. The AdsML Consortium takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain 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. The AdsML Consortium does not represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication, 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 Secretariat of the AdsML Consortium.

7. By using this specification in any manner or for any purpose, you release the AdsML Consortium from all liabilities, claims, causes of action, allegations, losses, injuries, damages, or detriments of any nature arising from or relating to the use of the Specification or any portion thereof. You further agree not to file a lawsuit, make a claim, or take any other formal or informal legal action against the AdsML Consortium, resulting from your acquisition, use, duplication, distribution, or exploitation of the Specification or any portion thereof. Finally, you hereby agree that the AdsML Consortium is not liable for any direct, indirect, special or consequential damages arising from or relating to your acquisition, use,

duplication, distribution, or exploitation of the Specification or any portion thereof.

8. This User License is perpetual subject to your conformance to the terms of this User License. The AdsML Consortium may terminate this User License immediately upon your breach of this agreement and, upon such termination you will cease all use duplication, distribution, and/or exploitation in any manner of the Specification.

9. This User License reflects the entire agreement of the parties regarding the subject matter hereof and supercedes all prior agreements or representations regarding such matters, whether written or oral. To the extent any portion or provision of this User License is found to be illegal or unenforceable, then the remaining provisions of this User License will remain in full force and effect and the illegal or unenforceable provision will be construed to give it such effect as it may properly have that is consistent with the intentions of the parties. This User License may only be modified in writing signed by an authorized representative of the AdsML Consortium. This User License will be governed by the law of Darmstadt (Federal Republic of Germany), as such law is applied to contracts made and fully performed in Darmstadt (Federal Republic of Germany). Any disputes arising from or relating to this User License will be resolved in the courts of Darmstadt (Federal Republic of Germany). You consent to the jurisdiction of such courts over you and covenant not to assert before such courts any objection to proceeding in such forums.

10. Except as expressly provided herein, you may not use the name of the AdsML Consortium, or any of its marks, for any purpose without the prior consent of an authorized representative of the owner of such name or mark.

IF YOU DO NOT AGREE TO THESE TERMS PLEASE CEASE ALL USE OF THIS SPECIFICATION NOW. IF YOU HAVE ANY QUESTIONS ABOUT THESE TERMS, PLEASE CONTACT THE SECRETARIAT OF THE ADSML CONSORTIUM.

AS OF THE DATE OF THIS REVISION OF THE SPECIFICATION YOU MAY CONTACT THE AdsML Consortium at www.adsml.org.

1.3 AdsML Code of Conduct

The AdsML Code of Conduct governs AdsML Consortium activities. A reading or reference to the AdsML Code of Conduct begins every AdsML activity, whether a meeting of the AdsML Consortium, AdsML Working Groups, or AdsML conference calls to resolve a technical issue. The AdsML Code of Conduct says:

Trade associations are perfectly lawful organizations. However, since a trade association is, by definition, an organization of competitors, AdsML Consortium members must take precautions to ensure that we do not engage in activities which can be interpreted as violating anti-trust or other unfair competition laws.

For any activity which is deemed to unreasonably restrain trade, AdsML, its members and individual representatives may be subject to severe legal penalties, regardless of our otherwise beneficial objectives. It is important to realize, therefore, that an action that may seem to make "good business sense" can injure competition and therefore be prohibited under the antitrust or unfair competition laws.

To ensure that we conduct all meetings and gatherings in strict compliance with any such laws and agreements in any part of the world, the AdsML Code of Conduct is to be distributed and/or read aloud at all such gatherings.

- There shall be no discussion of rates, fares, surcharges, conditions, terms or prices of services, allocating or sharing of customers, or refusing to deal with a particular supplier or class of suppliers. Neither serious nor flippant remarks about such subjects will be permitted.
- AdsML shall not issue recommendations about any of the above subjects or distribute to its members any publication concerning such matters. No discussions that directly or indirectly fix purchase or selling prices may take place.
- There shall be no discussions of members' marketing, pricing or service plans.
- All AdsML related meetings shall be conducted in accordance with a previously prepared and distributed agenda.
- If you are uncomfortable about the direction that you believe a discussion is heading, you should say so promptly.

Members may have varying views about issues that AdsML deals with. They are encouraged to express themselves in AdsML activities. However, official AdsML communications to the public are the sole responsibility of the AdsML Consortium. To avoid creating confusion among the public, therefore, the Steering Committee must approve press releases and any other forms of official AdsML communications to the public before they are released.

1.4 Document Number and Location

This document, Document Number AdsMLProofOfPublication-1.5.0-SpecP2Schema-AS-1, is freely available. It is located at the AdsML website at http://www.adsml.org/.

1.5 Purpose of this document

This document provides rules and guidelines for how to use the messages defined in the AdsML Proof of Publication standard. AdsMLProofOfPublication is an XML-based language used for encoding and routing messages that contain metadata about when, where and how an instance of an advertisement was actually published, including, optionally, a digital representation of the published advertisement.

1.6 Audience

The intended audience for this document is primarily user and vendor organizations who seek to implement the AdsML Proof of Publication standard in their workflows, advertising systems, or software products. Those assessing the conformance of vendor products to the standard may also use the document.

Comments on this specification should be addressed to the AdsML Consortium and to the Technical Working Group of the AdsML Consortium (technical.wg@adsml.org).

1.7 Accompanying documents

This document serves as the reference guide to the AdsML Proof of Publication schema. A companion document, *AdsMLProofOfPublication 1.5 Part 1 Usage Rules & Guidelines*, provides additional rules and guidance for using AdsML Proof

of Publication messages to address specific business requirements. They are meant to be read together.

In addition, elements and structures that are used in multiple AdsML schemas are documented in the *AdsML Type Library* specification. AdsMLProofOfPublication makes extensive use of such structures, therefore the *Type Library* specification is an essential reference.

All three documents are part of the AdsML Framework, which contains a suite of related documents. Readers of this document are assumed to be familiar with the full range of relevant AdsML documentation. In particular, readers are assumed to have read the *E-Commerce Usage Rules and Guidelines* document. A description of the entire document set can be found in the *ReadMeFirst* html file associated with this release of the Framework.

1.8 Definitions & conventions

1.8.1 Definitions of key words used in the specification

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are used as described in IETF RFC 2119 (See Section 5 References).

The key word "**DEPRECATED**" is used to indicate that structures are being phased out of the AdsML specifications. Structures marked as **DEPRECATED** will be removed in the next major schema upgrade and should not be used in new implementations.

When any of these words do not appear in upper case as above, then they are being used with their usual English language sense and meaning.

1.8.2 Naming conventions – element, attribute, type, and file names

All element, attribute, and type names follow the '`CamelCase`' convention.

Element and type names begin using upper camel case and begin with capitals (`UpperCamelCase`). For example, '`AdsML`', '`MessageRef`', and '`AdsMLStatusType`'.

Attribute names begin using lower camel case and begin with lower case (*`lowerCamelCase`*). For example, '*`language`*' or '*`messageId`*'.

File names also follow the camel case convention and use upper camel case for each segment of the file name, plus dashes to separate the segments of the file name. Only the first two digits of the version number are included in the file name. The third digit of the version number (if there is one) and the Draft Number are only shown internally within the document. The full naming conventions for AdsML schema and specification file names are described in the document *AdsML Document Names and Identifiers – Guidelines and Examples*, a copy of which is included in this release of the Framework.

Schema for user-defined extensions to AdsML should use AdsML naming conventions as detailed above. For example, '`ExampleInstanceFile.xml`', '`ExampleSchemaFile-1.0.xsd`', '`ExampleSchemaFile-1.1.xsd`'.

1.8.3 Typographical conventions

Element and type names are given in Courier New font, size 10. For example, `ProofOfPublication`.

Attribute names are given in italicized Courier New font, size 10. For example, *`messageCode`*.

When citing examples of values that could be assigned to elements or attributes, the value is given in Courier New font, size 9, so "…the attribute taking the value of '`12`'.".

1.9 Change History

Version	Date	Changes	Author
1.5.0 AS-1	15 April 2010	First Approved Specification of version 1.5.0. Previous change history removed.	JC
1.0.1 AS-2	30 May 2008	Minor editorial updates.	JC
1.0.0 AS-1	10 October 2007	First Approved Specification. Previous change history removed	JC

1.9.1 Changes in version 1.5.0

Version 1.5.0 is a major upgrade to the specification; changes made in version 1.5.0 are not backwards compatible with the previous release of AdsML ProofOfPublication, version 1.0.1. The change delta between these versions is recorded here.

1.9.1.1 New structures

Proofing Party

An optional `adsml:ProofingParty` element has been added.

Provenance

A `Provenance` structure has been added to record information about the source of the proof information.

Support for Interactive ads - AppearanceInformation.Interactive

A new `AppearanceInformation.Interactive` structure supports providing proof of publication for interactive (e.g. online) advertisements.

1.9.1.2 Updated structures

Advertisement Booking Information

An optional `adsml:MediaType` element has been added.

An optional `PlacementResult` element has been added.

An optional `Provenance` element has been added.

An optional adsml:Properties element has been added.

The optional adsml:DescriptionLine element has been made optional and repeatable. Multilingual descriptive information can now be recorded.

Appearance

An optional adsml:Properties element has been added.

The optional adsml-bo:NumberOfOccurrences element has been removed. (It is superseded by PlacementResult.)

AppearanceInformation – all variants (.Generic, .Insert, .Interactive, .NewspaperMagazine)

An optional Provenance element has been added.

An optional DistributionResult element has been added.

An optional adsml:MediaType element has been added. Note: that in the AppearanceInformation.Generic variant the adsml:MediaType has moved and now appears before adsml:AdType.

An optional adsml:AdditionalService element has been added.

All variants now have optional (repeatable except in .Insert) adsml-ma:AdContentReferences.

The adsml:SpecialRequirements element has been removed from all variants of ProductionDetails, as it was not considered necessary for the Prove Publication workflow.

The optional adsml:Description element has been made optional and repeatable. Multilingual descriptive information can now be recorded.

The optional adsml:Note element has been made optional and repeatable. Multilingual notes can now be recorded.

Distribution Result

The Distribution element has been renamed DistributionResult and is no longer a child of the Publication element but appears as a sibling to it.

Production Detail – all variants (.Generic, .Insert, .Interactive, .NewspaperMagazine)

All variants are now locally declared in the ProofOfPublication schema. All variants now have a repeatable Size element; an optional adsml:Properties element has been added.

Publication – all variants

All media variants now use a Publication element declared as adsml-bo:PublicationInformationType. The Publication.Insert element has been removed, therefore.

An optional adsml:Properties element has been added.

TearSheet

An optional `Provenance` element has been added.

The `TearSheet` element has been made repeatable.

The optional `adsml:DescriptionLine` element has been made optional and repeatable. Multilingual descriptive information can now be recorded.

1.10 Acknowledgements

This document is a product of the AdsML Technical Working Group. Primary authorship and editing was performed by,

- Jay Cousins (RivCom.) jay.cousins@rivcom.com

Acknowledgements and thanks to other contributors for additional input to this document are listed in Appendix A: Acknowledgement for contributions to this document.

1.11 The AdsML Consortium

The documents comprising the AdsML standard were written by the AdsML Technical Working Group, a committee charged with creating the consortium's technical deliverables, and then approved by the entire membership.

More information about the consortium can be found on the consortium's website: www.adsml.org.

2 Introduction

The AdsMLProofOfPublication standard has been developed by the AdsML Consortium to be a global standard for the exchange of metadata about when, where and how an instance of an advertisement was actually published, including, optionally, a digital representation of the published advertisement. In addition, AdsMLProofOfPublication has been designed with extensibility as an important objective in order to be able to grow with the business and support various business models and future requirements.

2.1 Relationship to the AdsML Framework

AdsML provides an XML framework, called the "AdsML Framework", for unifying and extending XML advertising standards. Where earlier advertising standards for e-commerce such as IfraAdConnexion or CREST focused on specific parts of the overall advertising process, the AdsML specifications fill in the gaps between such standards and specifications, extend their reach and encourage convergence when they overlap. In this line of effort, the AdsMLProofOfPublication standard has been developed by the AdsML Consortium as the preferred approach to handle delivery of metadata relating to a published instance of an advertisement.

For AdsMLProofOfPublication, the AdsML Framework provides a messaging infrastructure for delivery of proof of publication messages.

An important issue in enabling automatic business message flows is the use of common well-defined message choreography. One of the main components in the AdsML Framework is a set of business process models and related documentation that includes a definition of common process models for the workflows of selected advertising classes (*AdsML Advertising Component Interactions Analysis).* All business messages from the ad materials group will eventually be supported by AdsMLProofOfPublication. In this release of the standard, just one of the "Proof Of…" messages defined in the Advertisement Component Interactions Analysis is delivered: the Proof of Publication message (PO-PB).

The PO-PB message is used to confirm the publication of an advertisement and to provide information about how, when and where it was published. It may also, optionally, convey a digital copy of the published advertisement. This release of the standard supports delivery of the PO-PB message in broadcast fashion, followed by an Administrative Acknowledgement of receipt of the message. No other message exchange patterns are supported in this release.

2.1.1 Use of the AdsML XML Envelope is optional

AdsMLProofOfPublication uses the AdsML business process model as a foundation for its message types. It also imports and reuses controlled vocabularies and the type library from the Framework. However, it is important to note that AdsMLProofOfPublication does not require use of, nor support for, the AdsML Envelope standard. The actual transfer of AdsMLProofOfPublication messages can be performed by arbitrary method and software application, with or without the use of the AdsML Envelope. For instance, an AdsMLProofOfPublication message can be transmitted using other envelopes such as ebXML or BizTalk or directly by SOAP, FTP, HTTP or SMTP services. But it should nevertheless be noted that as the AdsML Envelope has been particularly developed to support message transfer within the advertising business and it is **RECOMMENDED** for use with the AdsMLProofOfPublication message format.

Please see the *AdsML Framework - Overview* and *E-commerce Usage Rules & Guidelines* for a more thorough discussion about the AdsML approach to e-

commerce. Also see *AdsMLProofOfPublication 1.5 Part 1 Usage Rules & Guidelines* for explanations of how to use the AdsMLProofOfPublication messages to accomplish bookings-related transactions.

2.1.2 Relationship to other advertising standards

AdsMLProofOfPublication focuses on the processes involved in the provision of proof of publication for advertisements, and it is intended to be used in conjunction with AdsML standards covering other areas in the advertising work flow, for example, AdsMLBookings, AdsMLMaterials, and AdsMLFinancials. However, use of these or any other AdsML standard is not required and so it can be used 'standalone'.

AdsMLProofOfPublication makes use of structures from the AdsMLBookings and AdsMLMaterials standards to provide information pertinent to the ad's booking and for specifying the delivery of the proof itself.

AdsMLProofOfPublication itself provides structures for use in AdsMLFinancials. AdsML Financials is developed by the AdsML Consortium as an XML standard for financial documents pertaining to advertising transactions, in particular invoices, credits, statements and payment notifications. Proof of publication information may be associated with invoice and credit lines in AdsML Financials documents to support the invoicing and reconciliation processes.

3 AdsMLProofOfPublication XML Schema – Overview

This section describes the use of XML Schema in the definition of AdsMLProofOfPublication.

3.1 Schema Architecture

AdsMLProofOfPublication uses a modular schema architecture as defined by the AdsML Framework architecture and consisting of the following schemas:

- The **Main Schema** – This schema defines the root element AdsMLProofOfPublication. All other components used in the standard are defined in it's Public Type Library.
- The **Public Type Library** – This schema defines all the components used in the standard, either by local definitions or by importing and/or including other schema files. This schema includes all the components from AdsMLProofOfPublication that may be imported into other standards and reused.
- The **AdsML Type Library** – This schema defines reusable components from the AdsML Framework.
- The **AdsMLBookings Public Type Library** – This schema defines components that make up the public part of the AdsMLBookings standard, some of which are reused within AdsMLProofOfPublication.
- The **AdsMLMaterials Public Type Library** - This schema defines components that make up the public part of the AdsMLMaterials standard, some of which are reused within AdsMLProofOfPublication.
- The **AdsML Controlled Vocabularies** – This schema defines all controlled vocabularies recommended by the AdsML Consortium.

All structures specific to AdsMLProofOfPublication are defined in the Main Schema or the Public Type Library that is included in the Main schema. These structures are all defined in the AdsMLProofOfPublication namespace.

Where possible, AdsMLProofOfPublication specific structures have been defined as derivations of general AdsML Framework components defined in the AdsML Type Library that is imported into both the Main Schema and the Public Type Library.

AdsMLBookings structures are used for informational purposes inside AdsMLProofOfPublication in order to facilitate reconciliation of the published ad instance to the associated booking, and also to provide information about where, how and to whom the ad was actually distributed.

AdsMLMaterials structures are used by AdsMLProofOfPublication to describe and specify the delivery of 'tearsheet(s)' – i.e. the proof.

The AdsML Controlled Vocabularies schema provides a set of controlled vocabularies (CVs) that may be used in AdsML messages. The CVs are made available to all document instances through import into the Main Schema.

3.1.1 Schema Files

The schema files from a particular standard are named as follows:

`AdsMLProofOfPublication-1.5-Main-AS.xsd`

The format starts with the name of the standard, "AdsMLProofOfPublication" followed by the current version number and the name of the schema within the standard. The last two characters provide the status of the standard as either PS (Proposed Standard) or AS (Approved Standard) for public releases. (Internal working documents have status code WD for Working Draft.)

The complete set of schema files used in the AdsMLProofOfPublication version 1.5, Approved Specification is thus:

AdsMLProofOfPublication-1.5-Main-AS.xsd

AdsMLProofOfPublication-1.5-PublicTypeLibrary-AS.xsd

AdsMLTypeLibrary-2.0-AS.xsd

AdsMLBookings-2.5-PublicTypeLibrary-AS.xsd

AdsMLMaterials-2.5-PublicTypeLibrary-AS.xsd

AdsMLControlledVocabularies-3.0-AS.xsd

3.2 AdsMLProofOfPublication Namespaces

AdsMLProofOfPublication defines a namespace:

'http://www.adsml.org/adsmlproofofpublication/1.5'

This is defined as the default namespace of the AdsMLProofOfPublication Schema. The schema specifies this using *targetNamespace* and *xmlns* attributes as illustrated below:

```
<xs:schema
targetNamespace="http://www.adsml.org/adsmlproofofpublication/1.5"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.adsml.org/adsmlproofofpublication/1.5"  ... >
```

Components reused from other standards carry their own namespaces that also have to be declared. The following external namespace definitions are also used:

adsml="http://www.adsml.org/typelibrary/2.0"

adsml-bo="http://www.adsml.org/adsmlbookings/2.5"

adsml-ma="http://www.adsml.org/adsmlmaterials/2.5"

adsml-cv="http://www.adsml.org/controlledvocabularies/3.0"

It is **RECOMMENDED** to use namespace prefixes as listed above.

It is **RECOMMENDED** to have the AdsMLProofOfPublication namespace as the default namespace in AdsMLProofOfPublication document instances. If, however, a namespace prefix is wanted, it is **RECOMMENDED** to use "adsml-pp".

3.3 Validation and Schema Location

A trading partner **MUST NOT** send any invalid AdsMLProofOfPublication messages. However, use of XML Schema based validation of production messages in runtime is **OPTIONAL**. Systems are allowed to use any available approach to ensure that their output is valid.

For production messages, a schema location **SHOULD NOT** be given in document instances using the *xsi:schemaLocation* attribute. Systems are **REQUIRED** to be able to identify which schema a particular document instance belongs to by reading the mandatory *adsml:schemaVersion* attribute.

3.4 Empty values for elements and attributes

For the rules concerning the use of 'null' values in elements and attributes see the section 'Mandatory vs. required, blanks vs. nulls' in the AdsML *E-commerce Usage Rules & Guidelines* document.

3.5 Fixed and Default values

All fixed or default values specified for elements or attributes in the schema **MUST** be present in an XML document instance conforming to that schema; schema validation and the post-schema-validation infoset (PSVI) **SHOULD NOT** be relied upon in order to make fixed or default values available for processing.

This restriction is imposed so that a particular mode of validation (XML Schema validation and the PSVI) is not relied upon to ensure that all data content of a message is present in an instance message. This allows for non-XML Schema validation of an instance.

This constraint is enforced in the schema by specifying attributes that carry fixed values with a 'use' of required, by not specifying default values, and by the policy that element content should not be empty in instances.

3.6 Message Content Overview

For each of the business message types supported by AdsMLProofOfPublication there is a corresponding content model. Content models vary across messages but wherever possible common content models are reused. The figure below shows the content model for the `ProofOfPublication` message type.

A `ProofOfPublication` message contains required `ProofOfPublicationIdentifier` and `AppearanceInformation` elements, and optional `AuxiliaryProofOfPublicationReferences`, `adsml:BusinessMessageDate`, information from the booking associated with the published ad, optional and repeatable `TearSheet`, and `adsml:Properties` elements.

A new proof of publication message must include a reference key. The `ProofOfPublicationIdentifier` element provides the reference key for the proof message and is generated by the party that initiates the prove publication process. In addition to this reference key for the proof of publication, the sender might also include other reference identifiers for the proof, for example, internal business identifiers, using the `AuxiliaryProofOfPublicationReferences` element.

Each proof of publication message must also include an `AppearanceInformation` element, this element containing information describing how, when, where and to whom the instance of the advertisement that is the subject of the proof of publication has been published.

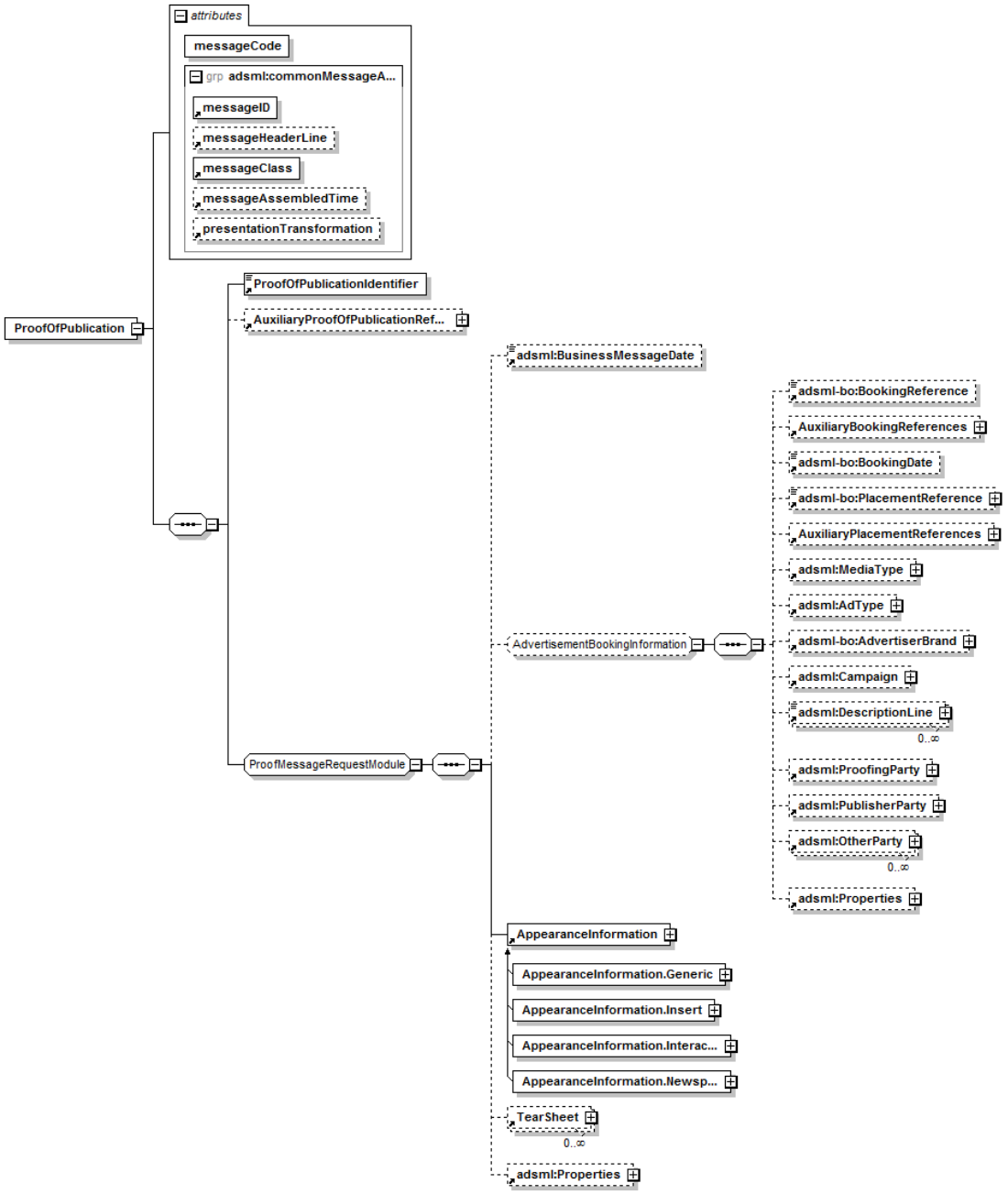
The `adsml:BusinessMessageDate` element can be used to record the business significant date for the message – that is, a business rather than technical timestamp. For example, the business date on which the message was issued.

Booking information from the booking associated with the published ad may optionally be included in a proof of publication message to support booking-proof reconciliation. Note that the booking information may be included for informational purposes only and that there is no assertion of its provenance in the proof of publication message. The assertion of provenance for included booking

information is outside this specification and would be a matter for agreement between trading partners in their Trading Partner Agreement (TPA). Within a Proof of Publication message provenance is only asserted for the proof information itself.

Each proof of publication message can contain optional and repeatable `TearSheet` elements. This element contains the 'tearsheet' (i.e. the actual proof) of the advertisement's appearance in the destination publication; the proof can be digital or physical. Using this structure, the proof file can either be carried inline or referenced externally. A `TearSheet` element can contain different `Rendering.TearSheet`(s), renditions of the advertisement's proof. Delivery information for each rendering of the tearsheet is specified by a `Delivery.TearSheet` element.

The `adsml:Properties` element allows user-specific properties to be defined, if required.



A proof of publication message is media-agnostic. All media-specific information is pushed down into the components of the design where media-specific content models allow for the delivery of content to multimedia.

The top (`TearSheet`) level is identical regardless of whether the ad will run in a newspaper or broadcast media. All media-specific information is instead pushed down into the components of the tearsheet, where distinct structures for describing the tearsheet file (`Rendering.TearSheet`) and its delivery (`Delivery.TearSheet`) provide a clear separation of content and delivery concerns, with specialization providing generic and media-specific structures as required.

In this version, the provision of proofs for print media (newspapers, inserts and magazines) are explicitly supported; a generic proof structure supports proofs for online and other media.

4 Content Model Reference

This is a reference section describing elements, attributes and other building blocks of the AdsMLProofOfPublication XML vocabulary's content model. The building blocks are listed in alphabetical order. The `AdsMLProofOfPublication` element is the root element, i.e. the top node of an AdsMLProofOfPublication message.

Each building block is briefly described with the intention of providing context and background as well as some technical detail about its usage. Particular focus is placed on issues and business rules that are not possible to express using XML Schema. Note that the XML Schema specification includes additional rules.

Components from imported external schemas are not described here; please see their specific specification documents. Such components are named with their recommended namespace prefix when discussed in the context of AdsMLProofOfPublication elements.

Elements and attributes with namespace prefix:	**Are described in the document:**
`adsml:`	*AdsMLTypeLibrary Schema & Specification*
`adsml-bo:`	*AdsMLBookings Schema & Specification*
`adsml-ma:`	*AdsMLMaterials Schema & Specification*

4.1 Root Element: AdsMLProofOfPublication

An AdsMLProofOfPublication message is an e-commerce business transaction that includes information to facilitate message transmission (a header with sender and recipient information) and the business content relevant to the transaction (e.g. proof of publication transaction data).

`AdsMLProofOfPublication` is the root element of the XML instance message where the namespace declaration is made. The namespace is defined on a string reflecting AdsML's ownership and the main version number. The namespace declaration **MUST** be based on the following string:

`'http://www.adsml.org/adsmlproofofpublication/1.5'`

The choice of namespace prefix is not defined in the standard, but it is **RECOMMENDED** that the AdsMLProofOfPublication namespace be the default namespace in AdsMLProofOfPublication messages. If a namespace prefix is required, it is **RECOMMENDED** to use `'adsml-pp'`. A namespace declaration is then **RECOMMENDED** to look like:

`xmlns:adsml-pp="http://www.adsml.org/adsmlproofofpublication/1.5"`

Every AdsMLProofOfPublication message contains a mandatory `Header` element followed by one or more elements of a specific business message type[1] such as `ProofOfPublication` for proving publication of an advertisement. The proofs contained in an AdsMLProofOfPublication message need not be related to each other in any way, other than being transmitted in the same physical XML message.

[1] Future releases of the standard will support further message types in the group of proof messages defined in the *Advertising Components Interactions Analysis* document. See this document for further information.

The root element `AdsMLProofOfPublication` is defined on the `adsml:AdsMLItemType`. Please see this type for further details.

The optional `adsml:Properties` element can be used to define application-specific extensions.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

Please see `adsml:AdsMLItemType` for details on attributes.

4.2 Transaction Messages

4.2.1 Element: ProofOfPublication

The `ProofOfPublication` element is the top level element for the corresponding business message. The proof of publication message is used to transmit the information constituting the proof of how, where, and when an advertisement has been published in a format acceptable to its intended recipient. The message is sent by the publisher or seller to the buyer of advertising.

The minimum requirement for a `ProofOfPublication` message is that it have an identifier (`ProofOfPublicationIdentifier`), and metadata describing how, when and where an advertisement has been published (`AppearanceInformation`). In some workflows metadata describing an ad's publication may be considered sufficient proof by itself. Additionally, tearsheet(s) of the published ad can be provided, represented in the message by repeatable `TearSheet` elements. Depending on the medium in which the ad was published, the proof could be digital (e.g. a graphical image file, XML document, video file, audio file, etc.) or physical (e.g. a physical representation of the ad such as a cutting from a magazine or newspaper).

The proof of publication content model consists of a sequence of a required `ProofOfPublicationIdentifier` element, optional `AuxiliaryProofOfPublicationReferences` and the `ProofMessageRequestModule` element group.

The `ProofOfPublication` must be identified using the `ProofOfPublicationIdentifier` element. A stack of other references for the proof of publication message can be provided using the `AuxiliaryProofOfPublicationReferences` element.

The `ProofMessageRequestModule` element group provides the core content model needed by all request messages for proof of publication.

For more information see these element definitions.

Attributes

messageCode (fixed: 'PO-PB')

The AdsML Framework message type code for the message.

***attribute group*: commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

4.3 Component Reference

4.3.1 Element Group: AdvertisementBookingInformation

The `AdvertisementBookingInformation` element group is an assembly of elements that carry booking information extracted from the order associated with the ad whose publication is being proved.

This booking information may come from any format of order but in an AdsML workflow it would be expected to be an AdsMLBookings `AdOrder` message. The booking information may optionally be included in the proof of publication message to support booking-proof reconciliation.

Note that this booking information comes from both the AdOrder and Placement levels of the booking and contains a subset of the data in an `adsml-bo:AdOrder` and `adsml-bo:Placement` element.

The `AdvertisementBookingInformation` contains a sequence of optional `adsml-bo:BookingReference`, `AuxiliaryBookingReferences`, `adsml-bo:BookingDate`, `adsml-bo:PlacementReference`, `AuxiliaryPlacementReferences`, `adsml:MediaType`, `adsml:AdType`, `adsml-bo:AdvertiserBrand`, `adsml:Campaign`, optional and repeatable `adsml:DescriptionLine`, optional `ProofingParty`, `PublisherParty`, `adsml:OtherParty` and `adsml:Properties` elements.

The `adsml-bo:BookingReference` and `AuxiliaryBookingReferences` elements are used to record a reference to the `BookingIdentifier` and to other reference identifiers for the booking associated with the advertisement.

The `adsml-bo:BookingDate` element is used to record the business significant date on which the booking was issued. This is a business rather than a technical timestamp for the booking.

The `adsml-bo:PlacementReference` and `AuxiliaryPlacementReferences` elements are used to record a reference to the `PlacementIdentifier` and to other reference identifiers for the placement associated with the advertisement. Note that,

- A single booking can contain many placements and so the use of placement-level reference identifiers allows the relationships between an ad proof and the placement ordering that advertisement to be specified unambiguously.
- The `adsml-bo:PlacementReference/@bookingReference` attribute is not used in the Proof context as the booking is identified using the `adsml-bo:BookingReference` element.

The `adsml:MediaType` element records the type of media in which the ad has been booked for publication. For example, 'Newspaper' or 'Outdoor'.

The `adsml:AdType` element records what kind of ad has been booked. For example, 'Classified' or 'Interactive'.

The `adsml-bo:AdvertiserBrand` element records the advertiser and the advertised brand(s) (or "products") that they advertised in the published ad instance.

The `adsml:Campaign` element records the name of an advertising campaign (also known as an "estimate" in some regions) of which the advertisement is a part.

The `adsml:DescriptionLine` element records a human-readable 'slugline' or other descriptive text describing the advertisement. The element is repeatable so a description can be provided in more than one human language.

In common with all AdsML messages, the main roles are represented by specific Party(s) at the top level of the message,

- The `ProofingParty` is the party with the overall business responsibility for the contents of the proofing message as a whole; it is the party that is distributing the proof information. This may or may not be the same as the `PublisherParty` and/or the `ProvenanceParty` (inside individual `Provenance` structures in `AppearanceInformation` or the `TearSheet`)
    - Note: the Proofing Party is the party that will distribute the proof (or has business responsibility for the proofing message as a whole). The Provenance Party is the party that takes responsibility for a section of the proofing information, e.g. Tearsheet or AppearanceInformation. Usually these will be the same party, but not always.
- The `PublisherParty` is the party with the business responsibility for publishing the advertisement
- The `adsml:OtherParty` element can be used to identify other parties in the prove publication workflow that have an interest in the ad whose publication is being proved.

The `adsml:Properties` element records any application-specific extensions recorded as user-defined properties.

See these type and element definitions for more information.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

See the '*AdsML Bookings 2.5 Specification*' for more information about the AdsML Bookings '`adsml-bo:`' namespace structures.

Attributes

No attributes.

4.3.2 Element: Appearance

The `Appearance` element records the temporal details of the point of time at which the ad actually appeared, i.e. when the ad ran in the publication.

The `Appearance` element is declared as `AppearanceType`, which is a subset of the `adsml-bo:SchedulingType`.

For more information, see the definition of `AppearanceType` in this specification and the definition of `adsml-bo:SchedulingType`.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the use of these elements in the AdsMLBookings Placement context.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.3 Element: AppearanceInformation

The `AppearanceInformation` element contains information describing how, when, where and to whom an instance of an advertisement has been published.

It is important to understand that the `AppearanceInformation` element is a container for booking placement data that reflects how the ad actually appeared when published. The `AppearanceInformation` data is, therefore, 'actual' data, and it is a subset of placement-level metadata from the AdOrder associated with the ad. By comparing the `AppearanceInformation` with the original `Placement` data from the booking, a comparison of the ad-as-ordered and the ad-as-actually-published can be made. This comparison supports the automated reconciliation of booking-to-actual data in the prove publication process.

In most cases the names of the elements in `AppearanceInformation` are identical to those used in the placement in the `adsml-bo:Placement` structure. The only exceptions to this are for placement's targeting and scheduling information, where the names change to reflect the fact that publication has taken place: the `PlacementTarget` in the booking becomes `PlacementResult` in the proof, and likewise the placement's `Scheduling` becomes the `Appearance` of the proof. Within the `Appearance` element, names are in some cases different to clearly mark that the data reflects the point of time at which the ad actually appeared when published. An example of this would be a time period, where an ad could have been ordered with a 'First Possible Time', but the appearance information states the 'First time' the ad was published. Note that the `AdvertisementBookingInformation` element group contains booking-level metadata from the AdOrder level and also metadata from the Placement level.

`AppearanceInformation` is an abstract element and thus cannot be directly used in a message, but would have to be substituted with media-specific extensions such as:

- `AppearanceInformation.Generic` - a media-agnostic generic structure
- `AppearanceInformation.Insert` - for print 'insert' advertisements
- `AppearanceInformation.Interactive` - for interactive 'online' advertisements
- `AppearanceInformation.NewspaperMagazine` - for newspaper and magazine advertisements

The `AppearanceInformation` contains a sequence of optional and repeatable `adsml:Description`, optional `adsml:MediaType`, `adsml:AdType`, `PlacementResult`, `Appearance`, `Provenance`, optional and repeatable `adsml:Note`, optional `adsml:Status` and `adsml:Properties` elements.

The `adsml:Description` element records a textual description of when, how, and where the ad appeared. The element is repeatable so a description can be provided in more than one human language.

The `adsml:MediaType` element specifies what form of media the ad was published in. Example values would be '`DigitalSign`', '`Newspaper`' or '`Online`'.

The adsml:AdType element identifies the type of the ad itself. Example values would be 'NewspaperClassified', 'Insert', 'Gatefold', 'Banner', etc.

The PlacementResult element records the details of the type and of the count of the publication event that took place, i.e. how many times the ad appeared. For further detail on PlacementResult refer to the element definition.

The Appearance element records the temporal details of the point of time at which the ad appeared.

The Provenance element records the source of the AppearanceInformation - who generated it, what with, when, and at what point in the process cycle. Within Provenance, an optional adsml:ProvenanceParty can be used to precisely identify the party providing the appearance information. Note that if the Provenance element is omitted, the provenance of all proof information in a ProofOfPublication message is assumed to be the ProofingParty authoring the proof message. A scenario for populating Provenance Party could be the use of a third party service.

The adsml:Note element records additional human-readable information about the appearance of the ad. The element is repeatable so notes can be provided in more than one human language if necessary. Although convenient, it is strongly **RECOMMENDED** not to use this element for important appearance data that can be recorded elsewhere in a more formal way.

The adsml:Status element records the status of the appearance information using a codified value.

The adsml:Properties element records any application-specific extensions recorded as user-defined properties.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the use of these elements in the AdsMLBookings Placement context.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library 'adsml:' namespace structures.

Attributes

No attributes.

4.3.4 Element: AppearanceInformation.Generic

The AppearanceInformation.Generic element is an extension of the AppearanceInformation element with generic placement data for any media type not explicitly supported by other appearance information elements. It contains a subset of the data in an adsml-bo:Placement.Generic element.

It extends the base appearance information to add a sequence of optional Publication, DistributionResult, ProductionDetail.Generic, and optional and repeatable adsml:AdditionalService and adsml-ma:AdContentReferences elements.

The Publication element records the publication where the ad has been run.

The DistributionResult element records the details of where, how and to whom the ad was distributed and the volume of its distribution to each category. For further detail on DistributionResult refer to the DistributionResult and PlacementResult element definitions.

The `ProductionDetail.Generic` element records technical production details of the ad's publication, including positioning and other technical specifications (e.g. specifications such as 'size').

The `adsml:AdditionalService` element records information about any additional value-add services that were ordered as part of the ad's placement and provided as part of the ad's appearance. A blind box provided by the publisher could be an example of such a service.

The `adsml-ma:AdContentReferences` element allows the artwork for the published ad to be optionally referenced using the AdsML QID identifier specified by the `adsml-ma:MaterialsIdentifier` or by other references using `adsml-ma:AuxiliaryMaterialsReferences`. In the scenario where the ad contained multiple sets of ad content the `adsml-ma:AdContentReferences` element is repeated.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings '`adsml-bo:`' namespace elements.

See the '*AdsMLMaterials 2.5 - Specification & Schema*' for more information about the AdsMLMaterials '`adsml-ma:`' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.5 Element: AppearanceInformation.Insert

The `AppearanceInformation.Insert` element is an extension of the `AppearanceInformation` element with media-specific content for insert advertisements. It contains a subset of the data in an `adsml-bo:Placement.Insert` element.

It extends the base appearance information to add a sequence of optional `Publication`, `DistributionResult`, `ProductionDetail.Insert`, optional and repeatable `adsml:AdditionalService` and optional `adsml-ma:AdContentReferences` elements.

The `Publication` element records the publication where the ad has been inserted and run.

The `DistributionResult` element records the details of the insert's distribution (i.e. region, zone, demographic) and the volume of its distribution to each category. For further detail on `DistributionResult` refer to the `DistributionResult` and `PlacementResult` element definitions.

The `ProductionDetail.Insert` element records technical production details of the insert's publication in a newspaper or magazine, including size, number of pages, weight, thickness, and any special requirements.

The `adsml:AdditionalService` element records information about any additional value-add services that were ordered as part of the ad's placement and provided as part of the ad's appearance.

The optional adsml-ma:AdContentReferences element allows the artwork for the insert to be optionally referenced using the AdsML QID identifier specified by the adsml-ma:MaterialsIdentifier or by other references using adsml-ma:AuxiliaryMaterialsReferences.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings 'adsml-bo:' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library 'adsml:' namespace structures.

Attributes

No attributes.

4.3.6 Element: AppearanceInformation.Interactive

The AppearanceInformation.Interactive element is an extension of the AppearanceInformation element with media-specific content for interactive (i.e. 'online') advertisements. It contains a subset of the data in an adsml-bo:Placement.Interactive element.

It extends the base appearance information to add a sequence of optional Publication, DistributionResult, ProductionDetail.Interactive, and optional and repeatable adsml:AdditionalService and adsml-ma:AdContentReferences elements.

The Publication element records the interactive media where the ad has been run.

The DistributionResult element records the details of how, where and to whom the ad was distributed, and the volume of its distribution to each category. For further detail on DistributionResult refer to the DistributionResult and PlacementResult element definitions.

The ProductionDetail.Interactive element records technical production details of the ad's publication in an interactive environment, including technical specifications such as how the ad was served, any capping applied, share of voice, the set(s) of ad content delivered as the ad was served, etc.

The adsml:AdditionalService element records information about any additional value-add services that were ordered as part of the ad's placement and provided as part of the ad's appearance.

The adsml-ma:AdContentReferences element allows the artwork for the published ad to be optionally referenced using the AdsML QID identifier specified by the adsml-ma:MaterialsIdentifier or by other references using adsml-ma:AuxiliaryMaterialsReferences. In the scenario where multiple sets of ad content were 'rotated' through the ad's appearance the adsml-ma:AdContentReferences element is repeated.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings 'adsml-bo:' namespace elements.

See the '*AdsMLMaterials 2.5 - Specification & Schema*' for more information about the AdsMLMaterialss 'adsml-ma:' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library 'adsml:' namespace structures.

Attributes

No attributes.

4.3.7 Element: AppearanceInformation.NewspaperMagazine

The `AppearanceInformation.NewspaperMagazine` element is an extension of the `AppearanceInformation` element with media-specific content for newspaper and magazine advertisements. It contains a subset of the data in an `adsml-bo:Placement.NewspaperMagazine` element.

It extends the base appearance information to add a sequence of optional `Publication`, `DistributionResult`, `ProductionDetail.NewspaperMagazine`, and optional and repeatable `adsml:AdditionalService` and `adsml-ma:AdContentReferences` elements.

The `Publication` element records the newspaper or magazine publication where the ad ran.

The `DistributionResult` element records the details of the publication's distribution (i.e. region, zone, demographic) and the volume of its distribution in each of those categories. For further detail on `DistributionResult` refer to the `DistributionResult` and `PlacementResult` element definitions.

The `ProductionDetail.NewspaperMagazine` element records technical production details of the ad's publication in a newspaper or magazine, including size, color, bleed, and positioning.

The `adsml:AdditionalService` element records information about any additional value-add services that were ordered as part of the ad's placement and provided as part of the ad's appearance.

The `adsml-ma:AdContentReferences` element allows the artwork for the published ad to be optionally referenced using the AdsML QID identifier specified by the `adsml-ma:MaterialsIdentifier`, or other references using `adsml-ma:AuxiliaryMaterialsReferences`. In the scenario where multiple sets of ad content were used – for example, an 'A/B Split' - the `adsml-ma:AdContentReferences` element is repeated.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings '`adsml-bo:`' namespace elements.

See the '*AdsMLMaterials 2.5 - Specification & Schema*' for more information about the AdsMLMaterials '`adsml-ma:`' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.8 Type: AppearanceType

The `AppearanceType` records the point of time at which an ad was published. The content model of the `AppearanceType` is a subset of the `adsml-bo:SchedulingType`.

An optional `ScheduleEntryReference` element provides a reference to the identifier for the scheduling information specifying the timing of an ad's publication in the AdsML AdOrder placement associated with the proof. This enables the scheduling of a published ad to be reconciled against the scheduling as originally ordered. The reference value is recorded as a string and is given the value of the `ScheduleEntryIdentifier` of the scheduling information in the associated placement.

An optional choice allows the temporal detail of the dates and date intervals defining when the ad actually ran to be recorded as either:

- ***Absolute time interval*** – A temporal interval with start and end points and optionally the number of times the ad was published. Recorded using `FirstTime`, `LastTime`, and optionally `adsml-bo:NumberOfOccurrences`.
- ***A pre-defined period*** – A temporal period. Recorded as a codified values for predefined periods (e.g. '`November Issue`' using the `PreDefinedPeriod` element using `adsml-bo:PreDefinedPeriod`.

Note that for the common case where an ad occurred once on a specific day, then this is recorded as an absolute time interval. The `FirstTime` and `LastTime` are given the same value; the `NumberOfOccurences` can optionally be present and given the value of '`1`' or can be omitted and assumed to have the value of '`1`'.

An optional `adsml:Properties` element allows user-specific properties to be defined, if required.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the use of these elements in the AdsMLBookings Placement context.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.9 Element: AuxiliaryBookingReferences

The `AuxiliaryBookingReferences` element is used to record additional 'other' reference identifiers specified by the publisher and other party(s) for the booking associated with an advertisement instance. The publishing party can use the optional `PublishersReference` to record their reference value for the booking. Other references for the booking may be recorded in repeatable `adsml:OtherReference` elements. The `adsml:OtherReference` element also records who the identifier was created by and is of interest to, thereby allowing additional information about who the reference should be used in communication with to be specified.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.10 Element: AuxiliaryPlacementReferences

The `AuxiliaryPlacementReferences` element is used to record other reference identifiers specified by the publisher and other party(s) for the booking associated with an advertisement instance. The publishing party can use the optional `PublishersReference` to record their reference value for the placement. Other references for the placement may be recorded in repeatable `adsml:OtherReference` elements. The `adsml:OtherReference` element also records who the identifier was created by and is of interest to, thereby allowing additional information about who the reference should be used in communication with to be specified.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.11 Element: AuxiliaryProofOfPublicationReferences

The `AuxiliaryProofOfPublicationReferences` element is used for recording additional identifiers for a proof of publication message. These referential identifiers are used to support reconciliation of the proof to the associated booking during the proof-booking reconciliation workflow. Such auxiliary references are optional and additional to the proof's primary identifier recorded by the `ProofOfPublicationIdentifier` element.

Note that the auxiliary proof of publication references are primarily identifiers for the proof of publication represented by the message, but that they also serve as reference identifiers for the proof of publication transaction.

The publishing and proofing parties can use the optional `adsml:PublishersReference` and `adsml:ProofersReference` elements to record their own reference identifier values for the proof.

Other reference identifiers assigned by other parties in the workflow may be recorded in repeatable `adsml:OtherReference` elements. Note that other references may also provide additional information about how the reference should be used in communications as well as other data – for example, by identifying who created the reference and to which party it is of interest.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.12 Element: Delivery.TearSheet

The `Delivery.TearSheet` element specifies the delivery of a proof when the proof content is not contained inline in a `Rendering.TearSheet` but is delivered externally to the message in which the `Rendering.TearSheet` appears, the delivery made by digital or physical means. For example, the proof is to be delivered by a courier, is being made available at a network location, or is

available in an asset management system. Each `Delivery.TearSheet` must reference one or more `Rendering.TearSheet`(s), allowing the same delivery method to be specified for more than one tearsheet rendering if required. For example, two tearsheets are delivered on digital media delivered by a courier. This allows for the delivery of proofs in scenarios such as the physical delivery of a hard proof by a courier service, or the digital delivery of a proof file via a digital delivery service or ftp.

The content model of the `Delivery.TearSheet` element is a restriction of the content model of the `adsml-ma:Delivery` element defined in AdsMLMaterials.

The `Delivery.TearSheet` differs from `adsml-ma:Delivery` by removing the following elements not considered relevant to the prove publication workflow:

- The `AuxiliaryDeliveryReferences` differ from `adsml-ma:AuxiliaryDeliveryReferences` by removing the `adsml-ma:MaterialsReceiversReference`.

See the '*AdsMLMaterials 2.5 - Specification & Schema*' for more information about the AdsMLMaterials '`adsml-ma:`' namespace elements.

Attributes

No attributes.

4.3.13 Element: DistributionResult

The `DistributionResult` records the outcome of where and how many times an ad has run. The element is declared as the `adsml-bo:DistributionType`.

For more information about how this element is used together with the Placement Result, please refer to the `PlacementResult` element definition.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the use of these elements in the AdsMLBookings Placement context.

Attributes

No attributes.

4.3.14 Element: FirstTime

The `FirstTime` element specifies the date and optional time at which the time interval defining the start and end points of an ad's publication begins. It is defined as a `adsml:DateTimeDateType` defined in the AdsML Type Library.

See `AppearanceType` for more information.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.15 Element: LastTime

The `LastTime` element specifies the date and optional time at which the time interval defining the start and end points of an ad's publication ends. It is defined as a `adsml:DateTimeDateType` defined in the AdsML Type Library.

See `AppearanceType` for more information.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.16 Element: PlacementResult

The `PlacementResult` element is a simplification of the `adsml-bo:PlacementTarget` element from AdsMLBookings. The Placement Target specifies the kind and number of the advertisement event ordered. The `PlacementResult` element records the details of what kind of advertisement event took place. The type of event that took place is recorded by the `adsml-bo:EventType` as a codified value; the number of times that the event happened is recorded as an integer by the `adsml-bo:EventCount` element. If required, further information about the placement result can be recorded as user defined properties in an optional `adsml:Properties` structure.

Note that the `PlacementResult` is used together with the `DistributionResult` element. In a booking, a `PlacementTarget` defines what kind of advertisement event is being purchased by the advertiser (both its type and count) while a `DistributionTarget` specifies how that advertisement will be distributed to people and places in order to achieve the placement target. In the proof message the results are recorded, that is, the actual targeting and distribution achieved.

The number of times that the purchased advertisement event took place is recorded as the `PlacementResult/EventCount`. The number or 'volume' of how often it was distributed to achieve that placement result is recorded as the `DistributionResult/TotalDistributionCount`. The counts serve different purposes, and although in many use cases they will be the same, that doesn't have to be the case. For example,

- If the target of an order is specified as 1000 Insertions published to all readers, then both `EventCount` and `TotalDistributionCount` would have the value "`1000`'
- If an order specified a target of 5 registrations to be achieved from a maximum of 1000 impressions, then the `PlacementResult/EventCount` would presumably be '`5`' and the `DistributonResult/TotalDistributionCount` would not exceed '`1000`'.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the use of these elements in the AdsMLBookings Placement Target context.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.17 Element: ProcessPoint

The `ProcessPoint` element records the point in the business process or workflow at which an event took place. In the case of provenance information, it records the process point where the provenance data was asserted. The `ProcessPoint` element is declared as `adsml:CodeType` and so the set of allowed values can be constrained to a user-defined list.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.18 Element: ProductionDetail.Generic

The `ProductionDetail.Generic` element contains a set of optional elements that record technical production details of an ad's publication.

The `ProductionDetail.Generic` element repeats the content model of the production detail specified for the ad in its original booking. It differs from the `adsml-bo:ProductionDetail.Generic` element in the following ways:

- The `Positioning` element has been simplified to record the actual positioning of the ad in the target publication media, since in the Prove Publication workflow the ad can only have one published position. If appropriate to the medium in question, the published position is recorded as an `adsml-bo:SectionCode`; further details about the ad's positioning can be recorded as codes or text strings using repeatable `adsml:Specifications` elements. The `Positioning` element is locally declared as the `ProofPositioningType.Generic` type.
- Note that the `Size` element is repeatable. This is to support the scenario where size is reported in more than one unit of measurement. In the standard proof scenario, though, size would usually be given once as an `AdSizeCode` or as a `Height` and `Width` combination rather than as an `Area`.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings '`adsml-bo:`' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.19 Element: ProductionDetail.Insert

The `ProductionDetail.Insert` element contains a set of optional elements that record technical production details of the insert's publication.

The ProductionDetail.Insert element repeats the content model of the production detail specified for the ad in its original booking by the adsml-bo:ProductionDetail.Insert element.

Note that the Size element is repeatable. This is to support the scenario where size is reported in more than one unit of measurement. In the standard proof scenario, though, size would usually be given once as an AdSizeCode or as a Height and Width combination rather than as an Area.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings 'adsml-bo:' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library 'adsml:' namespace structures.

Attributes

No attributes.

4.3.20 Element: ProductionDetail.Interactive

The ProductionDetail.Interactive element contains a set of optional elements that record technical production details of the ad's publication.

The ProductionDetail.Interactive element repeats the content model of the production detail specified for the ad in its original booking. It differs from the adsml-bo:ProductionDetail.Interactive element in the following ways:

- The Positioning element has been simplified to record the actual positioning of the ad in the target publication media, since in the Prove Publication workflow the ad can only have one published position. If appropriate to the publication in question, the published position is recorded as an adsml-bo:SectionCode; further details about the ad's positioning can be recorded as codes or text strings using repeatable adsml:Specifications elements. The Positioning element is locally declared as the ProofPositioningType.Generic type.
- Note that the Size element is repeatable. This is to support the scenario where size is reported in more than one unit of measurement. In the standard proof scenario, though, size would usually be given once as an AdSizeCode or as a Height and Width combination rather than as an Area.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings 'adsml-bo:' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library 'adsml:' namespace structures.

Attributes

No attributes.

4.3.21 Element: ProductionDetail.NewspaperMagazine

The `ProductionDetail.NewspaperMagazine` element contains a set of optional elements that record technical production details of an ad's publication.

The `ProductionDetail.NewspaperMagazine` element in AdsMLProofOfPublication differs from the `adsml-bo:ProductionDetail.NewspaperMagazine` element in the following ways:

- The color of the published ad is recorded using a locally declared `Colors` element with the same content model (`adsml-bo:ColorsType.Print`) as used in AdsMLBookings.
- The `Size` element is locally declared with the same content model (`adsml-bo:SizeType.Book`) as in AdsMLBookings, but has been made optional. Note that the `Size` element is repeatable. This is to support the scenario where size is reported in more than one unit of measurement. In the standard proof scenario, though, size would usually be given once as an `AdSizeCode` or as a `Height` and `Width` combination rather than as an `Area`.
- The `Positioning` element has been made optional and locally declared as `ProofPositioningType.NewspaperMagazine`. Its content model has been simplified in order to:
  - replace the `adsml-bo:PrimaryPositioning` and `adsml-bo:AlternativePositioning` children elements with optional `adsml-bo:PlacementInBook` and `PositionOnPage` children elements
  - replace `adsml-bo:PositionOnPage` with a simplified `adsml:PositionOnPage` element to record the actual x-y co-ordinates of the ad's position on the page.
  - make `adsml-bo:CuttablePosition` optional.

Note that the level of production detail recorded will vary with the type of the ad being produced. For example, color can often be omitted in the case of classified ads.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings '`adsml-bo:`' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.22 Element Group: ProofMessageRequestModule

The `ProofMessageRequestModule` element group is an assembly of elements that carry the information needed by request messages for providing proof of publication. It provides information about the message's business-significant issue date, information from the booking associated with the proof of publication, a set of metadata describing the actual appearance of the ad whose publication is being proved, and, optionally, tearsheets of the published ad.

The `ProofMessageRequestModule` contains a sequence of optional `adsml:BusinessMessageDate`, booking information within an `AdvertisementBookingInformation` element group, a required `AppearanceInformation`, optional and repeatable `TearSheet`, and optional `adsml:Properties` elements.

The `adsml:BusinessMessageDate` element is used to record the business significant date on which the message was issued. This is a business rather than a technical timestamp for the message.

The `AdvertisementBookingInformation` element group provides booking information extracted from the order associated with the ad whose publication is being proved (i.e. from an AdsMLBookings `AdOrder` message). The booking information may optionally be included in the proof of publication message to support booking-proof reconciliation.

The `AppearanceInformation` element must be used to describe the 'what, where, and how' of the ad's publication. Within this element metadata describing the type of ad published, the temporal aspect of the ad's publication, the publication in which the ad appeared, and technical details of the ad's production such as size and positioning can be placed.

The `TearSheet` element is a container structure for describing and specifying the delivery of a proof file, a 'tearsheet', that is included as part of the proof of publication.

The `adsml:Properties` element allows user-specific properties to be defined, if required.

For more information, see the element and type definitions.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.23 Element: ProofOfPublicationIdentifier

The `ProofOfPublicationIdentifier` element records the primary identifier for the proof delivered in a proof of publication message. The proof of publication identifier is declared as the `adsml:QIDType` defined in the AdsML Type Library. Note that a proof may or may not have associated with it an actual tearsheet (i.e. proof file) or tearsheets. A proof of publication message without a proof file may, therefore, be a sufficient proof in its own right; this allows for the usage scenario of a workflow where metadata describing the actual appearance of the ad alone provides a sufficient affidavit for an ad's publication. Where a tearsheet is provided, this can be described and delivered using the `TearSheet` element.

Note that the proof of publication identifier also serves as the identifier for the proof of publication transaction. The `ProofOfPublicationIdentifier` element provides the reference key for the proof of publication message and is generated by the party sending the message and initiating the prove publication process. In addition to this reference key for the proof of publication, the sender might also include other reference identifiers for the proof of publication, such as internal business identifiers, using the `AuxiliaryProofOfPublicationReferences` element.

See the section on "*Message References – Identifiers*" in the accompanying "*AdsMLProofOfPublication 1.5 Usage Rules & Guidelines*" document for further information about the use of the proof of publication identifier.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.24 Element: ProofOfPublicationReference

The `ProofOfPublicationReference` element provides a reference to the AdsML QID for the proof of publication.

Note: this element is intended to be reused by other AdsML specifications. It enables proof information from an AdsML `ProofOfPublication` message to be included in another business message and unambiguously associated by reference with the proof of publication message from which it was taken.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.25 Type: ProofPositioningType.Generic

The `ProofPositioningType.Generic` type contains a set of optional elements that record the actual positioning of an ad when it was published.

The positioning of the ad in the publication is recorded using optional `adsml-bo:SectionCode` and repeatable `adsml:Specifications` elements. If appropriate to the medium and publication, the published position is recorded as an `adsml-bo:SectionCode`; further details about the ad's positioning can be recorded as codes or text strings using repeatable `adsml:Specifications` elements.

The `ProofPositioningType.Generic` type is a simplification of the `adsml-bo:PositioningType.Generic` type, restricting the content model so that it records the actual positioning achieved.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings '`adsml-bo:`' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.26 Type: ProofPositioningType.NewspaperMagazine

The `ProofPositioningType.NewspaperMagazine` type contains a set of optional elements that record the actual positioning of an ad on a newspaper or magazine page.

The positioning of the ad in the 'book' (i.e. which page in the 'book' of pages) and the actual positioning of the ad on that page in the book are recorded using optional `adsml-bo:PlacementInBook`, `adsml:PositionOnPage`, and `adsml-bo:CuttablePosition` elements.

The `ProofPositioningType.NewspaperMagazine` type is a simplification of the `adsml-bo:PositioningType.NewspaperMagazine` type, restricting the content model so that it records the actual positioning achieved.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings '`adsml-bo:`' namespace elements.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.27 Element: Provenance

The `Provenance` element is used to assert the provenance (i.e. 'source' or 'origin') of metadata information. It asserts who has generated the information, with what kind of tool, at what time, and at what point in the production chain.

Note that if the `Provenance` element appears it overrides the default provenance of the proof of publication message, which is always the Proofing Party.

The `Provenance` element is declared as `ProvenanceType`.

Attributes

No attributes.

4.3.28 Element: ProvenanceDateTime

The `ProvenanceDateTime` element specifies the date and optional time at which a provenance assertion was made. It is declared as the `adsml:DateTimeDateType` defined in the AdsML Type Library.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.29 Element: ProvenanceSystem

The `ProvenanceSystem` element is used to record the name and version of a software application used to generate provenance information. The element is

declared as `adsml:VersionedStringType` – a string with a *`version`* attribute.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

adsml:version (optional)

Records the version of the tool used to generate the provenance data.

4.3.30 Type: ProvenanceType

The `ProvenanceType` contains a set of optional elements for recording provenance information.

The `ProvenanceType` content model consists of the following elements:

- An optional `adsml:DigitalSignatures` provides a digital signature for the provenance information.
- An optional `adsml:ProvenanceParty` element records the party responsible for making or generating the provenance assertion.
- An optional `ProvenanceSystem` describes the system used to generate the provenance information
- An optional `ProvenanceDateTime` records the date or date-time stamp for when the provenance assertion was made, i.e. the time the provenance data was generated.
- An optional `ProcessPoint` element specifies at what point in the production process the provenance assertion was made.

See these element definitions for more information.

Note that the Proofing Party is the party that will distribute the proof (or has business responsibility for the proofing message as a whole), while the Provenance Party is the party that takes responsibility for a section of the proofing information, e.g. `Tearsheet` or `AppearanceInformation`.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.31 Element: Publication

The `Publication` element contains a set of optional elements used to record details that identify and describe the publication in which an ad ran.

The `Publication` element is declared as `adsml-bo:PublicationInformationType`.

See the '*AdsMLBookings 2.5 - Specification & Schema*' for more information about the AdsMLBookings '`adsml-bo:`' namespace elements.

Attributes

No attributes.

4.3.32 Element: PublisherParty

The `PublisherParty` element identifies the party with the business responsibility for publishing the advertisement.

Note that the `PublisherParty` records the same data as contained by the `ProofOfPublication/AppearanceInformation/adsml-bo:PublishedBy` element.

The `PublisherParty` element is declared as the `adsml:PartyType` defined in the AdsML Type Library.

See the '*AdsML Type Library 2.0 Specification*' for more information about the AdsML Type Library '`adsml:`' namespace structures.

Attributes

No attributes.

4.3.33 Element: Rendering.TearSheet

The `Rendering.TearSheet` element is used to describe the characteristics of an individual rendering of a proof. It can also be used to contain the content of that proof inline. If the content is delivered by another means, then a sibling `Delivery.TearSheet` element will describe those delivery means.

The content model of the `Rendering.TearSheet` element is a restriction of the content model of the `adsml-ma:Rendering` element defined in AdsMLMaterials.

The `Rendering.TearSheet` differs from `adsml-ma:Rendering` by removing the following elements not considered relevant to the prove publication workflow:

- The `AuxiliaryRenderingReferences` has removed the `adsml-ma:CreatorsRenderingReference` child element.
- The `adsml-ma:ContentHandlingInstructions` has been removed.
- The `adsml-ma:Preflight` element has been removed.

See the '*AdsMLMaterials 2.5 - Specification & Schema*' for more information about the AdsMLMaterials '`adsml-ma:`' namespace elements.

Attributes

No attributes.

4.3.34 Element: TearSheet

A `TearSheet` describes the characteristics of the physical or digital proof of an advertisement and provides any associated delivery information for the proof file's delivery.

The `TearSheet` structure is media-agnostic and can be used for any media. Distinct structures for describing the proof content and its delivery provide a clear separation of content and delivery concerns.

It enables a proof to be identified and described (`Rendering.TearSheet`), for the delivery of the proof to be specified (`Delivery.TearSheet`). The

`TearSheet` structure allows multiple renditions of a proof to be described and delivered. Note that delivery information is specified separately to rendering information to allow (1) for the direct inclusion of a digital proof in a proof of publication file, in which case no additional delivery information is required, and, (2) to allow a single set of delivery information to be specified as the delivery instructions for one or more of the tearsheet rendering(s).

The `TearSheet` content model consists of the following elements:

- A required `TearSheetType` element used to identify what kind of tearsheet is being delivered, e.g. '`PageWithFacingPage`'.
- An optional and repeatable `adsml:DescriptionLine` used to provide a human-readable description of the tearsheet, e.g. '`Tear sheet of Ad for Main News Back Page`'. The element is repeatable so that a description can be recorded in more than one human language if necessary.
- An optional and repeatable `Rendering.TearSheet` used to describe the characteristics of a proof rendering, e.g. type, intended usage, status, description, identification, metadata describing the characteristics of a digital proof file, etc.
- An optional and repeatable `Delivery.TearSheet` used to describe how a proof rendering (or renderings) will be delivered, e.g. a hard proof by a courier service; a digital file via a digital delivery service or ftp.
- An optional `Provenance` element records the source of the `TearSheet` - who generated it, what with, when, and at what point in the process. Within `Provenance`, an optional `adsml:ProvenanceParty` can be used to precisely identify the party providing the tearsheet information. Note that if `Provenance` is omitted, the provenance of all proof information in a ProofOfPublication message is assumed to be the `ProofingParty` which authored the proof message. A scenario for populating Provenance Party within a tearsheet could be the use of a third party tearsheet service.
- An optional `adsml:Properties` element that allows user-defined properties

See these type and element definitions for more information.

Attributes

No attributes.

4.3.35 Element: TearSheetType

The `TearSheetType` element classifies what kind of tearsheet is being provided. The tearsheet type classification represents the context of the ad as published.

Example classifications are:

- '`AdOnly`' - a tearsheet of only the ad itself with no surrounding context from the publication
- '`FullPage`' – a tearsheet showing the ad in the context of the page in which it was published.

The `TearSheetType` element is declared as `adsml:CodeType` and so the set of allowed values can be constrained to a user-defined list.

See the AdsML Controlled Vocabularies `AdsMLTearSheetTypeCV` for a set of AdsML-defined values representing tearsheet types.

Note that to record the media type of the tearsheet as 'digital' or 'physical', the `adsml-ma:RenderingType` element is used.

See these type and element definitions for more information.

Attributes

No attributes.

5 References

5.1 Normative References

- [IETF RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. Internet Engineering Task Force (IETF), Request for Comments: 2119, March 1997 (http://www.ietf.org/rfc/rfc2119.txt)
- [W3C] W3C (World Wide Web Consortium). Ed. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. W3C Recommendation, 16 August 2006, edited in place 29 September 2006. (http://www.w3.org/TR/REC-xml)
- [W3C] W3C (World Wide Web Consortium). Ed. Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn. *XML Schema Part 1: Structures Second Edition*. W3C Recommendation, 28 October 2004. (http://www.w3.org/TR/xmlschema-1/)
- [W3C] W3C (World Wide Web Consortium). Ed. Paul V. Biron, Ashok Malhotra. *XML Schema Part 2: Datatypes Second Edition*. W3C Recommendation, 28 October 2004. (http://www.w3.org/TR/xmlschema-2/)
- [W3C] W3C (World Wide Web Consortium). Ed. Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin. Namespaces in XML 1.0 (Second Edition). W3C Recommendation, 16 August 2006. (http://www.w3.org/TR/xml-names)
- [W3C] W3C (World Wide Web Consortium). Ed. James Clark, Steve DeRose. *XML Path Language (XPath) Version 1.0*. W3C Recommendation, 16 November 1999. (http://www.w3.org/TR/xpath)
- [W3C] W3C (World Wide Web Consortium). *The "xml" Namespace*. W3C namespace for 'xml' prefix, 4 June 2001. (http://www.w3.org/XML/1998/namespace)

5.2 Other References (Non-Normative)

- [IETF RFC 1738] T. Berners-Lee, L. Masinter, M. McCahill. *Uniform Resource Locators (URL)*. Internet Engineering Task Force (IETF), Network Working Group, Request for Comments: 1738, December 2004 (http://www.ietf.org/rfc/rfc1738.txt).
- [IETF RFC 1741] P. Faltstrom, D. Crocker, E. Fair. *MIME Content Type for BinHex Encoded Files*. Internet Engineering Task Force (IETF), Network Working Group, Request for Comments: 1741, December 1994 (http://www.ietf.org/rfc/rfc1741.txt)
- [IETF RFC 1766] H. Alvestrand. *Tags for the Identification of Languages*. Internet Engineering Task Force (IETF), Request for Comments: 1766, March 1995 (http://www.ietf.org/rfc/rfc1766.txt)
- [IETF RFC 2045] N. Freed, N. Borenstein. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. Internet Engineering Task Force (IETF), Network Working Group, Request for Comments: 2045, November 1996 (http://www.ietf.org/rfc/rfc2045.txt)
- [IETF RFC 2046] N. Freed, N. Borenstein. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. Internet Engineering Task Force

(IETF), Network Working Group, Request for Comments: 2046, November 1996 (http://www.ietf.org/rfc/rfc2046.txt)

- [IETF RFC 2183] R. Troost, S. Dorner, K. Moore, Editor. *Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field*. Internet Engineering Task Force (IETF), Network Working Group, Request for Comments: 2183, August 1997 (http://www.ietf.org/rfc/rfc2183.txt)
- [IETF RFC 2387] E. Levinson. *The MIME Multipart/Related Content-type*. Internet Engineering Task Force (IETF), Network Working Group, Request for Comments: 2387, August 1998 (http://www.ietf.org/rfc/rfc2387.txt)
- [IETF RFC 2392] E. Levinson. *Content-ID and Message-ID Uniform Resource Locators*. Internet Engineering Task Force (IETF), Network Working Group, Request for Comments: 2392, August 1998 (http://www.ietf.org/rfc/rfc2392.txt)
- [IETF RFC 3066] H. Alvestrand. *Tags for the Identification of Languages*. Internet Engineering Task Force (IETF), The Internet Society, Request for Comments: 3066, January 2001. (http://www.ietf.org/rfc/rfc3066.txt)
- [IETF RFC 3986] T. Berners-Lee, R. Fielding, L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. Internet Engineering Task Force (IETF), Network Working Group, Request for Comments: 3986, January 2005 (http://www.ietf.org/rfc/rfc3986.txt).
- [ISO/IEC 10646] ISO (International Organization for Standardization). *ISO/IEC 10646-1993 (E). Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane*. International Organization for Standardization, 1993
- [ISO/IEC 10646-2000] ISO (International Organization for Standardization). ISO/IEC 10646-1:2000. *Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane*. International Organization for Standardization, 2000
- [ISO 639] ISO (International Organization for Standardization). *ISO 639:1988 (E). Code for the representation of names of languages*. International Organization for Standardization, 1988.
- [ISO 3166] ISO (International Organization for Standardization). *ISO 3166-1:1997 (E). Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes*. International Organization for Standardization, 1997.
- [ISO 8601] ISO (International Organization for Standardization). *ISO 8601:2000 Data elements and interchange formats -- Information interchange -- Representation of dates and times (2000-12-21)*. International Organization for Standardization, 2000.
- [W3C-NOTE-unicode-xml] Unicode Technical Committee, W3C Internationalization Working Group/Interest Group. *Unicode in XML and other Markup Languages. Unicode Technical Report #20, W3C Note 13 June 2003 and Time Formats*. 13 June 2003. (http://www.w3.org/TR/2003/NOTE-unicode-xml-20030613/)
- [X12] Accredited Standards Committee (ASC) X12. *X12.5 Interchange Control Structures*. Accredited Standards Committee (ASC) X12, December 1995.
- [X12] Accredited Standards Committee (ASC) X12. *X12.56 Interconnect Mailbag Control Structures*. Accredited Standards Committee (ASC) X12, December 1995.

- [X12] Accredited Standards Committee (ASC) X12. *X12.58 Security Structures*. Accredited Standards Committee (ASC) X12, December 1995.
- [X12] Accredited Standards Committee (ASC) X12. *X12.59 Implementation of EDI Structures – Semantic Impact*. Accredited Standards Committee (ASC) X12, December 1995.
- [X12] Accredited Standards Committee (ASC) X12. *X12.6 Application Control Structure*. Accredited Standards Committee (ASC) X12, December 1995.

6 Appendix A: Acknowledgement for contributions to this document

Acknowledgement and thanks for contributions to this document are also due to,

- Israel Viente (Vio Worldwide Ltd.) israel_viente@il.vio.com
- Mark Kondrad (magSend) mkondrad@magsend.com

This document is a product by the AdsML technical working group. In addition to the main authors listed above, the following members of the technical working group have provided major feedback contributions and know-how:

- The AdsML Technical Working Group.

