



AdsML[®] Framework for E-Commerce Business Standards for Advertising

AdsMLProofOfPublication 1.5.0 Part 1 Usage Rules & Guidelines

Document Authors: AdsML Technical Working Group

Document ID: AdsMLProofOfPublication-1.5.0-SpecP1Usage-AS-1

Document File Name: AdsMLProofOfPublication-1.5-SpecP1Usage-AS.pdf

Document Status: Approved Specification

Document Date: 15 April 2010

Draft Number: 1

Table of Contents

1	ADSMLEPROOFOPUBLICATION STANDARD DOCUMENTATION.....	4
1.1	DOCUMENT STATUS AND COPYRIGHT	4
1.2	NON-EXCLUSIVE LICENSE AGREEMENT FOR ADSML CONSORTIUM SPECIFICATIONS	4
1.3	ADSMLE CODE OF CONDUCT.....	6
1.4	DOCUMENT NUMBER AND LOCATION	7
1.5	PURPOSE OF THIS DOCUMENT	7
1.6	AUDIENCE	7
1.7	ACCOMPANYING DOCUMENTS	8
1.8	DEFINITIONS & CONVENTIONS	8
1.8.1	<i>Definitions of key words used in the specification</i>	8
1.8.2	<i>Naming conventions – element, attribute, type, and file names.....</i>	8
1.8.3	<i>Typographical conventions.....</i>	9
1.9	CHANGE HISTORY	9
1.9.1	<i>Changes in version 1.5.0</i>	9
1.10	ACKNOWLEDGEMENTS	10
1.11	THE ADSML CONSORTIUM	10
2	INTRODUCTION	11
2.1	IMPLEMENT ONLY WHAT YOU NEED	11
2.2	USE OF THE ADSML ENVELOPE IS OPTIONAL, BUT RECOMMENDED	12
3	BUSINESS MESSAGES OVERVIEW.....	13
3.1	MESSAGE COMPONENTS	13
3.2	INFORMATION CONTENTS	14
3.3	SUPPORTED MEDIA	14
4	MESSAGE CHOREOGRAPHY.....	16
4.1	USE CASES.....	16
4.2	ADMINISTRATIVE MESSAGES – ACKNOWLEDGMENT AND ERROR HANDLING	16
4.3	TESTING	17
4.4	RESPONSE MODES	17
4.5	BUSINESS MESSAGES	17
4.6	DATAGRAM MESSAGING FROM SENDER TO RECIPIENT.....	17
4.6.1	<i>Message References – Identifiers</i>	17
4.6.1.1	<i>Internal identifiers.....</i>	18
4.6.1.2	<i>Cross-references to Invoices and Bookings</i>	19
5	USAGE OF BUSINESS MESSAGES	20
5.1	THE STATUS OF APPEARANCE INFORMATION IN A PROOF OF PUBLICATION MESSAGE.....	20
5.2	THE STATUS OF ADVERTISEMENT BOOKING INFORMATION IN A PROOF OF PUBLICATION MESSAGE	20
6	USE CASES AND RECOMMENDED SOLUTIONS.....	21
6.1	DELIVER A DIGITAL TEARSHEET VIA AN FTP UPLOAD	21
6.2	DELIVER A DIGITAL TEARSHEET IN-LINE IN THE ADSML MESSAGE	21
6.3	DELIVER A DIGITAL TEARSHEET BY PROVIDING A LINK FROM WHICH IT CAN BE DOWNLOADED 21	21
6.4	DELIVER BOTH A DIGITAL TEARSHEET AND A HARD PROOF OF THE SAME ADVERTISEMENT	22
6.5	DELIVER MULTIPLE TEARSHEETS FOR THE SAME ADVERTISEMENT.....	23
6.6	DELIVER ONLY METADATA WHICH DESCRIBES THE PUBLICATION OF AN ADVERTISEMENT.....	23
6.7	CONVEY PROOF OF PERFORMANCE INFORMATION	24
6.8	DELIVER PERFORMANCE INFORMATION ABOUT AN ONLINE AD.....	24
6.9	DELIVER METADATA ABOUT THE PERFORMANCE OF AN ADVERTISEMENT WITH MULTIPLE SETS OF AD CONTENT	25
6.10	DELIVER AN AFFIDAVIT ASSERTING THAT PUBLICATION OCCURRED	26
6.11	PROVIDE INFORMATION ABOUT THE BOOKING WITH WHICH THIS TEARSHEET IS ASSOCIATED .	26
6.12	REFERENCE THE INVOICE WHICH RELATES TO THIS TEARSHEET	26

6.13	DELIVER TEARSHEETS FOR AN A/B SPLIT (PRINT)	27
6.14	PROVIDE DETAILS THAT DESCRIBE THE DISTRIBUTION OF AN INSERT	27
6.15	PROVIDE DETAILS THAT DESCRIBE THE TARGETED DISTRIBUTION OF AN AD	28
6.16	PROVIDE THIRD PARTY PERFORMANCE VERIFICATION	30
6.17	DELIVER PROOF METADATA IN MORE THAN ONE HUMAN LANGUAGE	31
7	CONFIGURATION CHECKLIST.....	33
7.1	CONVEYANCE OF BINARY OBJECTS – BY INLINE OR EXTERNAL TRANSMISSION	33
7.2	USE OF ENCODING AND ENCRYPTION OF DIGITAL TEARSHEET PAYLOAD	33
7.3	CONVEYANCE OF LOGIN AND PASSWORD INFORMATION	33
7.4	MULTILINGUAL METADATA.....	34
8	REFERENCES – (NON-NORMATIVE)	35
9	APPENDIX A: ACKNOWLEDGEMENT FOR CONTRIBUTIONS TO THIS DOCUMENT	36

1 AdsMLProofOfPublication Standard Documentation

1.1 Document status and copyright

This is the Approved Specification of the *AdsMLProofOfPublication 1.5 Part 1 Usage Rules & Guidelines*.

Information in this document is made available for the public good, may be used by third parties and may be reproduced and distributed, in whole and in part, provided acknowledgement is made to AdsML Consortium and provided it is accepted that AdsML Consortium rejects any liability for any loss of revenue, business or goodwill or indirect, special, consequential, incidental or punitive damages or expense arising from use of the information.

Copyright © 2010 AdsML Consortium. All rights reserved. Information in this document is made available for the public good, may be used by third parties and may be reproduced and distributed, in whole and in part, provided acknowledgement is made to AdsML Consortium and provided it is accepted that AdsML Consortium rejects any liability for any loss of revenue, business or goodwill or indirect, special, consequential, incidental or punitive damages or expense arising from use of the information.

Copyright Acknowledgements: The AdsML Non-Exclusive License Agreement is based on the "Non-Exclusive License Agreement" on Page iii of "OpenTravel™ Alliance Message Specifications – Publication 2001A", September 27, 2001, Copyright © 2001. OpenTravel™ Alliance, Inc. The AdsML Code of Conduct is based on the "OTA Code of Conduct" on Page ix of "OpenTravel™ Alliance Message Specifications – Publication 2001A", September 27, 2001, Copyright © 2001. OpenTravel™ Alliance, Inc.

1.2 Non-Exclusive License Agreement for AdsML Consortium Specifications

USER LICENSE

IMPORTANT: AdsML™ Consortium specifications and related documents, whether the document be in a paper or electronic format, are made available to you subject to the terms stated below. Please read the following carefully.

1. All AdsML™ Consortium Copyrightable Works are licensed for use only on the condition that the users agree to this license, and this work has been provided according to such an agreement. Subject to these and other licensing requirements contained herein, you may, on a non-exclusive basis, use the Specification.
2. The AdsML™ Consortium openly provides this specification for voluntary use by individuals, partnerships, companies, corporations, organizations and any other entity for use at the entity's own risk. This disclaimer, license and release is intended to apply to the AdsML™ Consortium, its officers, directors, agents, representatives, members, contributors, affiliates, contractors, or coventurers (collectively the AdsML™ Consortium) acting jointly or severally.

3. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this Usage License are included on all 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-SpecP1Usage-AS-1, is freely available. It will be 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 AdsMLProofOfPublication 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 AdsMLProofOfPublication messages to address specific business requirements. A companion document, *AdsML Proof of Publication 1.5 Part 2 Specification & Schema*, provides additional rules and guidance for using the AdsML Proof of Publication schema. 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 8 References](#)). 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'`.

In many cases, element names mentioned in usage guidelines and narrative text in this document do not include their namespace prefix. For example, the element

`adsml-bo:BookingInformation` is often referred to as simply `'BookingInformation'`. This simplification is provided in order to make the text easier to read. Element names in code fragments are always shown with their full namespace prefix.

1.8.3 Typographical conventions

Element and type names are given in Courier font as, for example, `ProofOfPublication`.

Attribute names are given in italicized Courier font as, for example, `messageCode`.

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

1.9 Change History

Version	Date	Changes	Editor
1.5.0 AS 1	15 April 2010	Approved Specification 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.	TS, JC

1.9.1 Changes in version 1.5.0

Version 1.5.0 of AdsML Proof of Publication is a major upgrade that aligns the specification with versions 2.5 of the AdsMLBookings and AdsMLMaterials specifications.

The significant changes to Version 1.5 of AdsMLProofOfPublication Usage Rules & Guidelines are in the Use Cases section to,

- Update the Use Case *'Deliver multiple tearsheets for the same advertisement'* to illustrate conveying multiple tearsheets in a single ProofOfPublication message
- Add a new Use Case *'Deliver performance information about an Online ad'*
- Add a new Use Case *'Deliver metadata about the performance of an advertisement with multiple sets of ad content'*
- Add a new Use Case *'Provide details that describe the targeted distribution of an ad'*
- Add a new Use Case *'Provide third party performance verification'*
- Add a new Use Case *'Provide proof metadata in more than one human language'*
- Add a new Configuration checklist entry for *'Multilingual metadata'*.

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
- Tony Stewart (RivCom.) tony.stewart@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.adsm.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.

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 and delivery of proof of performance 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*). This release of AdsMLProofOfPublication supports delivery of just one of the “Proof Of...” messages that are defined in the Advertisement Component Interactions Analysis: 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. This ‘how, when and where’ additionally provides proof of performance. 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 Implement only what you need

The AdsML Framework aims to provide advertisers, publishers, broadcasters and their suppliers with a consistent toolkit of standards, messages and transactions that can be used to automate every aspect of the advertising supply chain, in any media, anywhere in the world. This means that even though it is still incomplete, the Framework already contains more standards and message types, and can convey more types of information, than any single organization is likely to need.

In order to implement AdsML-based e-commerce, therefore, trading partners and their vendors (or industry associations acting on their behalf) are expected to review the AdsML Framework and decide:

- Which AdsML standards they will implement within their particular region or business activity
- Within those standards, which business transactions they will support (this determines the types of messages they will exchange)

- Within those messages, which types of information they will include (this determines the optional structures that they will implement)
- Within those types of information, which specific data values they will “control” (this determines their use of controlled vocabularies).

Each AdsML standard defines its mandatory and optional components, and where appropriate, each provides a Configuration Checklist to help users discuss and agree on the features and functionality that they will implement. These implementation decisions can be agreed privately between the trading partners, and/or codified in a formal “profile” which is made publicly available in order to encourage interoperability.

Based on their customers’ implementation decisions, vendors can decide which types of AdsML functionality to implement in their systems. In order to market a system’s AdsML capabilities, a vendor might indicate that it supports specific named Profiles, and/or the vendor might use the relevant Configuration Checklist(s) to describe the supported capabilities.

Further information about these concepts can be found in *AdsML E-Commerce Usage Rules & Guidelines*, in the *Advertising Components Interactions Analysis*, and in the Specification for each standard.

NOTE: Even though you can implement just those portions that you need, all of the standards and features in the AdsML Framework are designed to work together as a cohesive whole, in that they share a common approach to advertising e-commerce that makes them “AdsML”.

2.2 Use of the AdsML Envelope is optional, but recommended

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 *AdsML E-commerce Overview* for a more thorough discussion about the AdsML approach to e-commerce.

3 Business Messages Overview

A Proof of Publication message is sent by the publisher or invoicer of an advertisement to an interested party (typically the buyer and/or paying party(ies) of that advertisement) so as to provide them with the evidence or information they need in order to approve payment for the publication of the advertisement. The message contains metadata identifying the advertisement and describing where, when and how it was published, optionally accompanied by information about a physical or digital tearsheet which has been provided by the publisher and/or can be retrieved by the message recipient from a specified location. If the tearsheet is in digital format, it can optionally be conveyed in the Proof of Publication message itself.

AdsML Proof of Publication supports the business process model and message flow as proposed in the *AdsML™ Advertising Component Interaction Analysis*¹, a part of the AdsML Framework. In particular, AdsML Proof of Publication supports the set of business messages that belongs to the "Proof of..." message group (PO).

The business message supported in the current release of AdsML Proof of Publication is:

Message Code	Message Name
PO-PB	Proof of Publication

Note: this list of messages reflects the subset of the PO group messages that have been implemented in AdsMLProofOfPublication 1.5.

The usage of other types of proof-of-publication and proof-of-performance related messages defined in the ACIA will be described when support for those messages is given by the AdsML Proof of Publication standard.

For the complete list of PO group messages defined see the *Advertising Components Interactions Analysis*.

The Proof of Publication is sent in broadcast fashion to all interested parties. Each party is able to reply with an automated Administrative Response message (acknowledging receipt of the message) but the current release of AdsMLProofOfPublication does not support business-level responses to a Proof of Publication message.

3.1 Message components

The main components of the Proof of Publication message are:

- Identifiers of the booking, placement and advertisement that was published.
- Descriptive information about the advertisement, including advertiser, publisher and campaign names.
- Information about when and where the advertisement was actually published, in the form of date/time, publication or broadcast name, location within that publication or broadcast, page number, etc.

¹ Note: this document is often referred to by the acronym 'ACIA'.

- Information about a tearsheet provided by the publisher which represents the published advertisement exactly as it appeared in the publication. Alternative renderings of the tearsheet, such as digital and physical (i.e. hard copy) versions, may be described.
- Delivery information indicating when and by what method(s) each of the tearsheet renderings are being provided to the publisher. In the case of a digital tearsheet, this can include instructions for how to retrieve that tearsheet from a specified location.

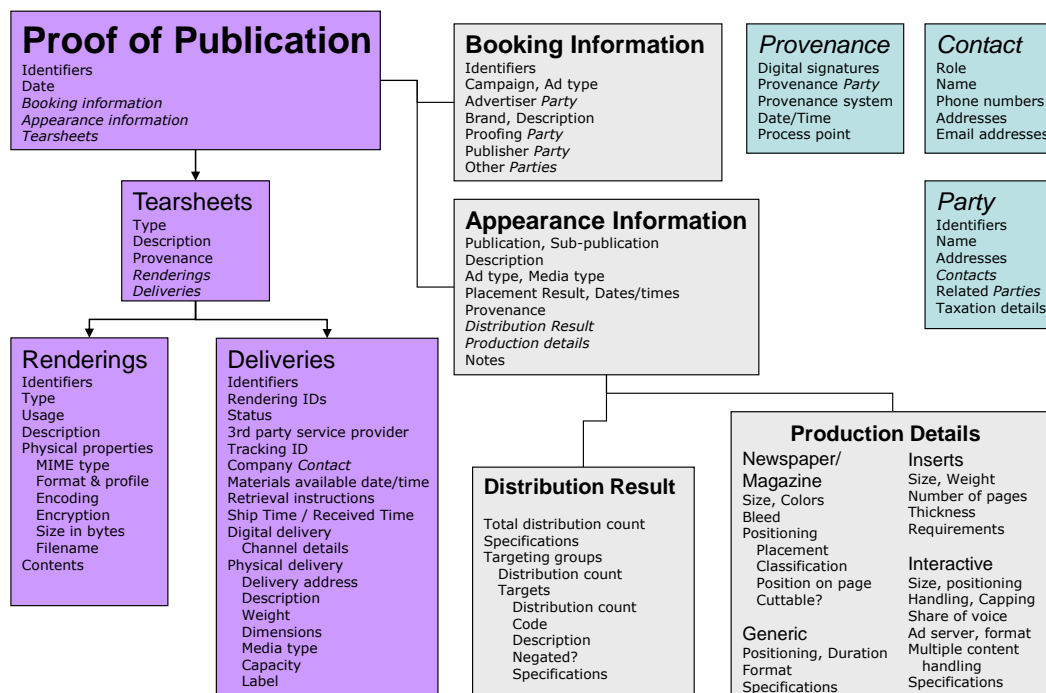
3.2 Information contents

Here is a diagrammatic overview of the potential information in a Proof of Publication message.

This view omits the generic message header as well as many of the smaller details, in order to see the main context-specific information “at a glance”. Much of the information is optional, intended for use in specific circumstances, and some of it can *only* be used in those specific circumstances. Therefore a given message instance will not contain all of the information shown here.

Overview of the potential information in an AdsML Proof of Publication message

(AdsMLProofOfPublication 1.5 April 2010)



3.3 Supported media

A proof of publication message is media agnostic in that most of the information is identical regardless of whether the ad ran in a print publication or broadcast media. All media specific information is instead pushed down into the components of the package, in the `AppearanceInformation`. `AdsMLProofOfPublication` provides specific versions of `AppearanceInformation` targeted for describing ad placements in particular media where media specific differences have been

accounted for - Newspaper Magazine, Insert and Interactive media. A generic `AppearanceInformation` structure is provided for 'other' media.

4 Message Choreography

This is a normative section describing the expected message flow between communications partners in a Proof of Publication transaction.

In addition, implementations of AdsMLProofOfPublication **MUST** support the specifications provided in the *AdsML E-commerce Usage Rules & Guidelines*.

4.1 Use cases

This release of AdsMLProofOfPublication supports the following use cases, or combinations thereof:

1. The Publisher (or an agent on its behalf) sends a digital copy of the published ad (e.g. an e-tearsheet) embedded in a proof of publication message, accompanied by metadata describing when, where, how and to whom the ad was actually published.
2. The Publisher (or an agent) sends a proof of publication message which contains instructions for retrieving an e-tearsheet, manually or automatically, accompanied by metadata describing when, where, how and to whom the ad was actually published.
3. The Publisher (or an agent) sends a proof of publication message which describes a physical copy of the published ad that the recipient can expect to receive (e.g. hard-copy/paper or on digital media), along with metadata describing when, where, how and to whom the ad was actually published
4. The Publisher (or an agent) sends metadata describing when, where, how and to whom an ad was actually published, without providing a copy of the ad itself.

Notes:

- The word “tearsheet” is used in AdsMLProofOfPublication to describe a representation of the published ad in any media, not just print ads. In print workflows tearsheets are widely used; they can represent part of a page, an entire page, multiple pages, or a copy of the entire publication, as necessary. In other media the use of tearsheets is much less common, and when used they tend to be called by different terms.
- The accompanying metadata is optional.

4.2 Administrative Messages – Acknowledgment and Error handling

Administrative messages are an integral part of the AdsML Framework. As a general case, for example, the recipient of an AdsML business message is expected to send an administrative response to that message promptly, upon receipt of the business message, in order to indicate that the business message was received and to convey any AdsML-level errors that may have been found in it. At the same time, the contents of the business message are forwarded to the appropriate application.

The rules governing administrative messages and error handling are generic and apply to the entire AdsML Framework. These rules **MUST** be followed when sending and receiving AdsMLProofOfPublication messages. For a description of administrative messages and error handling, please see the *AdsML E-commerce Usage Rules & Guidelines*.

4.3 Testing

The rules governing test messages are generic and apply to the entire AdsML Framework. These rules **MUST** be followed when sending and receiving test AdsMLProofOfPublication messages. For a description of test messages, please see the *AdsML E-commerce Usage Rules & Guidelines*.

4.4 Response Modes

This release of AdsMLProofOfPublication supports a single message type which is sent in broadcast mode using a datagram communications model.

4.5 Business Messages

AdsML Proof of Publication supports a single business operation: the delivery of proof of publication information. This information refers to a previous business transaction, which consisted of booking and publishing the advertisement in question, and is intended to lay the groundwork for the next transaction, which will consist of paying for the publication of that advertisement.

AdsMLProofOfPublication supports a single message type, PO-PB, which is expressed as a code value for the *messageCode* attribute on business message elements such as *ProofOfPublication*. The code values are defined by and used in the AdsML Framework.

4.6 Datagram messaging from sender to recipient

This release of AdsMLProofOfPublication supports only one workflow: datagram messaging from sender to recipient.

- 1) The sender sends a Proof of Publication (PO-PB) message to the recipient. Once the sender has received an Administrative Response from the recipient (indicating that the message was received), the recipient is assumed to have accepted the proof of publication.

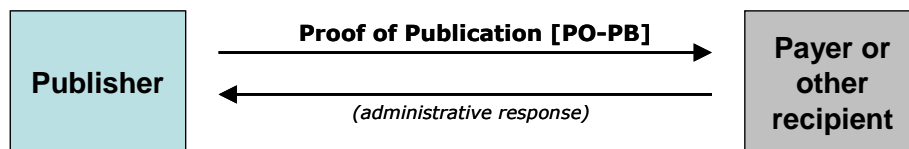


Figure: Proof of Publication delivery

- 2) If the content is unacceptable to the recipient or its delivery fails, then it is up to the recipient to contact the sender by a non-AdsML mechanism and resolve the problem.
- 3) To redeliver a proof of publication, a new proof of publication message exchange is always initiated.

4.6.1 Message References – Identifiers

The AdsMLProofOfPublication standard supports an asynchronous messaging model. For a general discussion, please see the '*AdsML E-commerce Usage Rules & Guidelines*'.

4.6.1.1 Internal identifiers

The AdsMLProofOfPublication message is delivered in broadcast fashion. There are no identifiers within the Proof of Publication message which explicitly reference another Proof of Publication message.

However, each instance of Proof of Publication business information (that is, the business contents of a proof of publication message) must be uniquely identified.

Support for this is provided by a mandatory

`ProofOfPublicationIdentifier` element, which uses the AdsML QID format. Additional references are recorded in the `AuxiliaryProofOfPublicationReferences` element using optional `adsm1:ProofersReference`, `adsm1:PublishersReference` and additional optional other references. The format of these additional references can be in any format that the parties choose.

In addition, each rendering of a tearsheet must be uniquely identified. Support for this is provided using the same elements as in the AdsMLMaterials delivery standard: a mandatory `adsm1-ma:RenderingIdentifier`, which uses the AdsML QID format, plus an optional `AuxiliaryRenderingReferences` structure whose contents include `adsm1:PublishersReference` and can be in any format that the parties choose.

In general, each message has the following identifiers:

- *The proof of publication identifier.* An identifier first issued by the party that initiates the proof of publication delivery business process. The proof of publication identifier is the primary identifier for the proof of publication information and **MUST NOT** change during the life of the proof of publication. Its structure **MUST** conform to the AdsML QID format, and it **MUST** be included in any proof of publication message transmitted, both requests and responses. The proof of publication identifier is called `ProofOfPublicationIdentifier`. The `ProofOfPublicationIdentifier` is used as the primary identifier for materials when referenced from other messages, in particular, an invoice line item.
- *The auxiliary proof of publication references.* An optional stack of identifiers that can be used to convey internal identifiers that the publisher or other parties use to reference the proof of publication. The auxiliary references are optional; however, once provided any reference identifier value **MUST** remain stable throughout the life of the proof of publication. The *auxiliary proof of publication references* are called the `AuxiliaryProofOfPublicationReferences`. The `AuxiliaryProofOfPublicationReferences` are used as secondary reference identifiers for the proof when referenced from all other AdsML messages. They contain a set of named references: `ProofersReference`, `PublishersReference` and an `OtherReference`. The `OtherReference` reference is repeatable and may be used to record additional reference identifiers for other parties involved in the workflow.

All messages have a unique message id:

- *The message ID.* A unique identifier for the business message. Each message ID **MUST** conform to the AdsML QID format and **MUST** be different from any other message ID. The message ID appears in the business message and is called *messageID*.

Note that the message ID is different from the physical transmission ID in the `AdsMLProofOfPublication` root element. The first time a message is transmitted, its message ID and transmission ID **MAY** be the same, but if the same message is later re-transmitted, its message ID **MUST** remain the same while its transmission ID **MUST** change. The transmission ID is referenced by an `AdministrativeResponse` message to identify the transmission that is being acknowledged, while the `messageID` is used in the response half of a request-response pair of messages to identify the request message to which it is a response.

Please see the section on “Globally Unique Identifiers” in *E-Commerce Usage Rules & Guidelines* for information regarding how identifiers may be expressed.

4.6.1.2 Cross-references to Invoices and Bookings

Typically, there exists a business relationship between a proof of publication and an invoice in that the line items on an invoice are not approved for payment until the payer has received the applicable proof of publication information. This relationship can be expressed in AdsML messages in two ways:

- In an AdsMLFinancials document, an Invoice line item can explicitly reference one or more Proof of Publication identifiers relating to that invoice or invoice line
- Both an AdsML Invoice and an AdsML Proof of Publication can provide the booking and placement identifiers for the advertisement in question. By this means, an invoice line item and its matching proof of publication can be indirectly identified.

5 Usage of business messages

5.1 The status of Appearance Information in a Proof of Publication message

The Proof of Publication message includes an optional `AppearanceInformation` structure in which information about the published advertisement can be conveyed. Although this structure re-uses many of the same elements that are found in an `AdsMLBookings Ad Order` message, the purpose of the `AppearanceInformation` is to convey information about the advertisement *as it actually appeared* in the specified publication or medium. This information may well differ from the publication details that were requested in the original booking, because while an ad order message conveys the *intended* specifications for the publication of an advertisement, the `AppearanceInformation` in a Proof of Publication message describes the *actual* publication of that advertisement.

If an `AppearanceInformation` structure is included in a `ProofOfPublication` element, it **MUST** describe an actual published instance of the advertisement. If a `ProofOfPublication` element also contains or references a `Tearsheet`, the `Tearsheet` **MUST** represent the same published instance of the advertisement that is described by the `AppearanceInformation` in that `ProofOfPublication` element.

5.2 The status of Advertisement Booking Information in a Proof of Publication message

The Proof of Publication message includes an optional `AdvertisementBookingInformation` group structure which allows booking information extracted from an `AdsMLBookings Ad Order` to be embedded in `ProofOfPublication` (PO-PB) messages defined by `AdsMLProofOfPublication`.

Such information is contained for informational purposes and is intended to facilitate the reconciliation of the proof of publication information with its associated booking. It is important to note that all such booking information contained in a proof of publication message is for informational purposes only and so **MUST NOT** be considered as having anything more than an informational status.

6 Use Cases and Recommended Solutions

This section provides a set of sample scenarios and their handling using AdsML Proof of Publication and explains how to handle typical proof delivery scenarios.

6.1 Deliver a digital tearsheet via an FTP upload

Scenario: A publisher (or its agent) wishes to initiate delivery of a digital tearsheet which will be transmitted via FTP.

AdsML handling: The publisher's systems create a `ProofOfPublication` (PO-PB) message and transmit it to the tearsheet recipient. The `ProofOfPublication` message must contain the full details of the delivery in the `ProofMessageRequestModule`, and must include the publisher's unique ID for this proof of publication in the `ProofOfPublicationIdentifier` element. The digital tearsheet (e.g. a PDF or similar digital file) is described as a single `Rendering.TearSheet` within the `TearSheet` element. The delivery instructions for the tearsheet are specified using a `Delivery.TearSheet` element, in this case specifying the FTP as a digital delivery.

6.2 Deliver a digital tearsheet in-line in the AdsML message

Scenario: A publisher (or its agent) wishes to initiate delivery of a digital tearsheet with the tearsheet contained in-line within the AdsML message.

AdsML handling: The publisher's systems create a `ProofOfPublication` (PO-PB) message and this is then transmitted to the tearsheet recipient to make the delivery, the tearsheet contained inline within the `adsml:ContentData` element in the `Rendering.TearSheet` structure, encoded as necessary so that it can be contained in an XML message. The type of encoding used (along with other technical metadata about the embedded tearsheet) is conveyed in the `adsml:ContentProperties` structure. As the content is inline there is no `Delivery.TearSheet` structure in the `TearSheet`, only a `Rendering.TearSheet` and associated metadata as provided by the `TearSheet`.

6.3 Deliver a digital tearsheet by providing a link from which it can be downloaded

Scenario: A publisher (or its agent) wishes to initiate delivery of a digital tearsheet by providing a URL from which the tearsheet can be retrieved.

AdsML handling: The publisher's systems create a `ProofOfPublication` (PO-PB) message and this is transmitted to the tearsheet recipient to initiate the delivery. A mandatory ID for the tearsheet and optional metadata about it are conveyed in `Rendering.TearSheet`. The delivery instructions are specified in `Delivery.TearSheet`, in this case using `adsml-ma:RetrievalInstructions` to tell the user (or an automated system) how to retrieve the tearsheet, and `adsml-ma:DigitalDelivery` to provide the URL from which it should be retrieved. For example:

```
<Delivery.TearSheet>
```

```

<adsm1-ma:RenderingReference>nyherald.com:2008-10-10:CRTV-89483-
R1</adsm1-ma:RenderingReference>

  <adsm1-ma:DeliveryIdentifier>nyherald.com:2008-10-10:CRTV-89483-
D3</adsm1-ma:DeliveryIdentifier>

  <adsm1-ma:MaterialsAvailableDateTime>2008-10-10T16:28:05-05:00</adsm1-
ma:MaterialsAvailableDateTime>

  <!-- The user is instructed to go to
http://adportal.nyherald.com/services/pop/login.jsp and use
"rgilman@creativeservices.com" in the Email address field and "CRTV-89483-
D3" in the File ID field. Password is not disclosed in this context, and
should be known by other means.-->

  <adsm1-ma:RetrievalInstructions>
    <adsm1:Code>
      <adsm1:CodeList>
        NYH Ad Portal User Authentication
      </adsm1:CodeList>

      <adsm1:CodeValue>
        rgilman@creativeservices.com
      </adsm1:CodeValue>
    </adsm1:Code>

    <adsm1:Code>
      <adsm1:CodeList>NYH Ad Portal file ID</adsm1:CodeList>
      <adsm1:CodeValue>CRTV-89483-D3</adsm1:CodeValue>
    </adsm1:Code>
  </adsm1-ma:RetrievalInstructions>

  <adsm1-ma:DigitalDelivery>
    <adsm1:CommunicationChannel.WWW>
      <adsm1:URI>
        http://adportal.nyherald.com/services/pop/login.jsp
      </adsm1:URI>
      <adsm1:Label>
        NY Herald digital proof of pub hub
      </adsm1:Label>
    </adsm1:CommunicationChannel.WWW>
  </adsm1-ma:DigitalDelivery>
</Delivery.TearSheet>

```

Note: Although the above example assumes that the message recipient already knows what password to use, it is also possible to include a password in the AdsML message. The password can be added to the URL as a parameter at the end of the string, or it can be conveyed in a retrieval instructions code.

6.4 Deliver both a digital tearsheet and a hard proof of the same advertisement

Scenario: A publisher provides both a digital tearsheet and a hard proof of the same advertisement. Both of them show the same portion of the page, the only

difference is their physical format. The tearsheet is made available for download from a web site, and the hard proof is sent by mail.

AdsML handling: The digital and hard copy versions of a tearsheet are considered two “renderings” of that tearsheet. The Proof of Publication message therefore should contain two `Rendering.Tearsheet` elements. Use the `adsml-ma:RenderingType` code to identify one of them as “`Digital.Tearsheet`” and the other as “`Physical.Tearsheet`”. If required to specify the intended usage of a tearsheet, this can be specified using the `adsml:Usage` element. These values are taken from the AdsML “Proof of Publication Type” controlled vocabulary. Because each of them is being delivered by different means, create two `Delivery.TearSheet` structures, one for the digital tearsheet and the other for the hard copy version. The `Delivery.Tearsheet` for the digital rendering provides retrieval instructions. The `Delivery.Tearsheet` for the hard proof describes the means by which it was sent and the address to which it was sent.

Note: This scenario assumes that except for their format, the two renderings are of exactly the same image and page portion. If they are different then they are considered two different tearsheets. See the next scenario.

6.5 Deliver multiple tearsheets for the same advertisement

Scenario: A publisher provides more than one tearsheet for the same advertisement, for example, a digital tearsheet showing the page on which an ad appeared, and a voucher copy of the entire publication. The tearsheet is made available for download from a web site, and the voucher copy is sent by courier.

AdsML handling: Because these are actually different types of tearsheets (rather than different renderings of the same tearsheet), they must be conveyed in different tearsheet elements. Within the `ProofOfPublication` element two `TearSheet` elements are populated, each of which will contain a single `Rendering.TearSheet` to describe the characteristics of the tearsheet, and a single `Delivery.TearSheet` to indicate how it is being delivered. Use the `TearSheetType` code to identify them as, respectively, “`AdInContext`” and “`FullCopy`” tearsheets. These values are taken from the AdsML Tearsheet Type CV.

Note: It would be possible to convey the tearsheets in two different `ProofOfPublication` elements that were either packaged in a single AdsML Proof of Publication message (PO-PB), or in two different PO-PB messages. Such alternative handling would be perfectly valid but it would not make use of the message’s functionality and so it would not represent optimized message use.

6.6 Deliver only metadata which describes the publication of an advertisement

Scenario: A publisher wishes to send textual information (metadata) which describes when, where, how and to whom the ad was published, and therefore also serves as an informal assertion that publication actually occurred. Neither a digital nor physical tearsheet will be provided.

AdsML handling: Depending on the medium in question, the publisher populates either `AppearanceInformation.NewspaperMagazine`, `AppearanceInformation.Insert`,

`AppearanceInformation.Interactive` or `AppearanceInformation.Generic` with information about when, where and how the ad appeared. Structured details about the publication can be conveyed in elements such as `Appearance` (date/time), `Publication` (publication and sub-publication name), `DistributionResult` (regions, zones, editions, demographics, etc.), `Size`, `Colors` and `Positioning`. An unstructured description, such as a sentence or two of text, can be conveyed in the `adsm1:Description` element at the top level of the `AppearanceInformation` structure. Because no tearsheet is being provided, the message does not contain a `TearSheet` element.

Notes:

- `AppearanceInformation` should contain information about what actually happened, which is not necessarily the same as what was specified at the time of booking. Use the elements in the `AdvertisementBookingInformation` group to convey information from the booking.
- AdsML does not explicitly support the transmission of a proof of publication “Affidavit”.

6.7 Convey Proof of Performance information

Scenario: A publisher provides proof of performance information to the buyer of the advertisement. Details of the ad’s appearance in the publication are provided using the Proof of Publication business message.

AdsML handling: The publisher’s system generates an AdsML ProofOfPublication business message. A ProofOfPublication element is generated and populated with metadata to provide proof.

The `AppearanceInformation` structure is used to provide the proof of performance information. The ‘when’ and ‘how’ is recorded using `Appearance`, `ProductionDetail`. The ‘where’ is recorded using the `Publication` elements to identify the publication and `PlacementResult` and `DistributionResult` to describe where the ad was published. Distribution details can be recorded as codes and/or text strings.

Notes: The definition of what constitutes “performance” information will be trading partner specific.

6.8 Deliver performance information about an Online ad

Scenario: A publisher provides textual proof of publication for an online ad.

AdsML handling: The advertisement was an online ad so its appearance is described using the `AppearanceInformation.Interactive` structure. Metadata describing the ‘when’, ‘where’, ‘how’ and ‘to whom’ of the ad’s publication is given. The date/time range of publication is recorded using the `Appearance` element. The web site where the ad ran is recorded as `Publication` metadata, the overall results that were achieved (e.g. the total number of impressions or appearances) are conveyed in the `PlacementResult` element, and information about the regions where the ad appeared and/or the demographics of the people who received it are given in `DistributionResult`. Data about the ad’s positioning in the website (the section where it appeared) as well as technical details about the content and the controls applied to it as the content was served is recorded as

`ProductionDetail.Interactive` metadata. For example, the ad serving system, the ad's format, how the ad's delivery was capped, other specifications defining how the ad was produced as required.

Notes: Because this is an interactive advertisement, the `Tearsheet` element is not used.

6.9 Deliver metadata about the performance of an advertisement with multiple sets of ad content

Scenario: A publisher provides textual proof of publication for an online ad that rotated multiple sets of ad content throughout its run.

AdsML handling: The proof is provided using a single `ProofOfPublication` element just as if there had only been one set of content. In most cases the fact that multiple sets of content were published as part of this placement is ignored.

If the parties do wish to provide details about the use of multiple ad content, then this is done by using production detail metadata and multiple ad content references.

`ProductionDetail.Interactive/adsml-bo:MultipleAdContentHandling` can be used to describe how the multiple sets of ad content were handled.

The different sets of ad content which rotated through the placement can be identified using the repeatable `AdContentReferences` element. Each set of ad content (i.e. each set of artwork rotated through the placement) is identified by an `AdContentReferences` element to record the unique AdsML identifier for the artwork and/or reference identifiers assigned by specific parties in the workflow such as the publisher or advertiser. Because no tearsheet is being provided, the message does not contain a `TearSheet` element.

Notes:

- Tearsheets are not normally provided in this scenario, or for any interactive advertisement. If tearsheets were provided, then different tearsheets would be given for each set of ad content; within a tear sheet a user defined extension property could be used to assert which set of ad content it is for. Best practice, though, would be to use a separate `ProofOfPublication` element for each different set of ad content in order to unambiguously associate the correct ad content and tearsheet combinations (See FAQ *Deliver tearsheets for an A/B Split*). Note that the `TearSheet` in a `ProofOfPublication` message is repeatable to allow for the scenario where different types of tearsheet for a set of ad content are conveyed in the message (as described in FAQ *Deliver multiple tearsheets for the same advertisement*).
- Multiple ad content handling. The `AppearanceInformation` ``.Generic`, `.Interactive` and `.NewspaperMagazine` variants all have repeatable AdContentReferences. This cardinality supports scenarios where a proof message provides a single set of AppearanceInformation for an ad that involved multiple sets of ad content. This is always optional: it would also be possible to send a separate ProofOfPublication structure for each different unit of ad content. All other data in the ProofOfPublication message applies to the combination of artwork.`

6.10 Deliver an affidavit asserting that publication occurred

Scenario: A publisher wishes to send an affidavit asserting that publication occurred.

AdsML handling: AdsML does not explicitly support the transmission of a proof of publication affidavit. An affidavit is a legal document whose format and use should be explicitly agreed between trading partners, who may determine that an affidavit would require additional metadata that is not provided in a Proof of Publication message.

Note: It is possible for trading partners to agree that the appearance information contained in a Proof of Publication message should be treated as if it were an affidavit, even though it was not designed to serve that purpose. See “Deliver only metadata which describes the publication of an advertisement”, above.

6.11 Provide information about the booking with which this tearsheet is associated

Scenario: A publisher wishes to send information about the booking which caused this advertisement to be published.

AdsML handling: Booking information can be conveyed in the optional elements contained in the `AdvertisementBookingInformation` group. If the booking was done via AdsML messages, its AdsML QID identifiers should be placed in the `adsml-bo:BookingReference` and `adsml-bo:PlacementReference` elements. Non AdsML identifiers such as the buyer’s and seller’s internal booking and placement numbers should be conveyed in `AuxiliaryBookingReferences` and `AuxiliaryPlacementReferences`. Other elements are provided for the publisher and publication names, advertiser, brand, etc.

Note: `AdvertisementBookingInformation` should contain information as it existed at the time of booking. It should not be used to convey information about what actually appeared. Use `AppearanceInformation` for that purpose.

6.12 Reference the invoice which relates to this tearsheet

Scenario: A publisher wishes to identify the invoice which includes the charge for publication of this advertisement in the Proof of Publication message metadata.

AdsML handling: AdsML does not provide structured metadata in the Proof of Publication message to reference or convey invoicing information. This scenario was deliberately not supported because in most cases the proof of publication message is generated by a workflow that does not have access to invoicing information, and may be sent to parties that will not receive the invoice.

Note: An invoice can reference a proof of publication, thus supporting the scenario in which the recipient of an invoice needs to associate it with the relevant tearsheet(s). Also, both a proof of publication and an invoice can reference the booking which triggered publication. These cross-references should be sufficient to enable recipients of an invoice or tearsheet to associate it with the other types of information that are required for their purposes.

6.13 Deliver tearsheets for an A/B Split (print)

Scenario: An advertiser orders an 'A/B split' ad for a newspaper where every second copy of the publication should have a specific artwork printed. To support this, the original order had 2 pieces of artwork associated with it. The advertiser requested tearsheets for both the 'A' and 'B' variants of the artwork of the A/B split ad. The publisher provides one tearsheet for each of the 'A' and 'B' variants of the ad.

AdsML handling: Because the ad has multiple ad content associated with it and a proof is requested for each set of ad content, 2 different tearsheets are required, 1 for each variant version of the published ad. As with any set of tearsheets based on different artwork, each of them should be placed in its own `ProofOfPublication` element, each of which contains a `Rendering.TearSheet` to describe the tearsheet, and a single `Delivery.TearSheet` to specify the digital delivery. As the proofs were booked with one order, the same booking and appearance information appears in both proofs of publication.

Note:

- To identify which set of artwork the proof is for (i.e. for the 'A' or 'B') the optional `adsm1-ma:AdContentReferences` element in `AppearanceInformation` can be used to specify the AdsML QID and/or additional reference identifiers for the artwork the proof is for.
- The two `ProofOfPublication` elements can either be packaged in a single AdsML Proof of Publication message, or can be sent in separate AdsML Proof of Publication messages.

6.14 Provide details that describe the distribution of an insert

Scenario: A publisher informs the buyer of a print insert advertisement how widely the insert was distributed. To achieve this details of the ad's appearance in the publication are provided using the Proof of Publication business message.

AdsML handling: The publisher's system generates an AdsML `ProofOfPublication` business message. For each publication where the insert was distributed a separate `ProofOfPublication` element is generated and populated with metadata.

The publisher populates `AppearanceInformation.Insert` with information about the when, where, and how of the ad's publication event, the ad's 'appearance'. The `AdType` is specified as 'Insert'. The temporal aspect of the ad's appearance is recorded using an `Appearance` element, with the insertion period of the ad – a single day in the newspaper – specified using `FirstTime` and `LastTime`. Optionally, other optional metadata can be recorded.

Details about the publication itself are given in the `Publication` element. The distribution of the ad is recorded using the `PlacementResult` and `DistributionResult` elements to specify the count of inserts distributed and where the insert was distributed, recording the regions, zones, editions, et cetera.

Notes: In this scenario, the values in `PlacementResult/EventCount` and `TotalDistributionCount` in `DistributionResult` will be identical. `PlacementResult` is the primary location in which to record the total result of a given placement and should always be populated. `TotalDistributionCount` either should not be used or should contain a copy of the same value as `PlacementResult/EventCount`.

6.15 Provide details that describe the targeted distribution of an ad

Scenario: A publisher provides proof of publication for a print insert ad, inserted into a newspaper and distributed across an intersection of specific zip codes. A detailed breakdown of the insert's total distribution across the specific zip codes is provided.

AdsML handling: The publisher's system generates an AdsML ProofOfPublication business message. For the publication where the insert was distributed a single ProofOfPublication element is generated and populated with metadata.

The publisher populates `AppearanceInformation.Insert` with information about the when, where, and how of the insert's insertion, the ad's 'appearance'. The `AdType` is specified as 'Insert'. The temporal aspect of the ad's appearance is recorded using an `Appearance` element, with the insertion period of the ad – a single day in the newspaper – specified using `FirstTime` and `LastTime`. Optionally, other metadata can be recorded.

Details about the publication itself are given in the `Publication` element. The total number of times that the insert was distributed – the result achieved by the placement – is recorded within `PlacementResult` using `EventCount`. A value of 'piece' is recorded as the `EventType`, indicating the result to be that 'x' number of 'pieces' were distributed.

A detailed breakdown of how the insert was distributed across the required combination of zip codes is recorded by a `DistributionResult` element. Within `DistributionResult` the repeatable `Targeting` structure is used to record the distribution made to those specific combinations of zip codes. Specific distribution counts for the individual zip codes are recorded within individual `Targeting` element and the sum of the total distribution made is recorded as the `DistributionResult/TotalDistributionCount`. Because it is an insert ad, the values of `DistributionResult/TotalDistributionCount` and `PlacementResult/EventCount` are the same.

For example, a proof reporting the distribution of inserts across specific zip codes requested in the AdsMLBookings message sample '3-1_AdReservation.xml' would appear as below.

```
<!--AdsMLBookings message sample '3-1_AdReservation.xml' requests the
distribution of 135500 inserts across specific zip codes. 35000 are
distributed as home delivery single copy, the remainder distributed as Total
Market Coverage. The proof information details an overall placement result
of 135500 insert ads distributed and specifies how the ads were targeted for
delivery by different methods for the designated postcodes. -->
<!-- The placement result is recorded - 135500 ads distributed. -->
<PlacementResult>
  <adsm1-bo:EventType>
    <adsm1:CodeValue xsi:type="adsm1-
cv:AdsMLUnitOfMeasureCV">piece</adsm1:CodeValue>
  </adsm1-bo:EventType>
  <adsm1-bo:EventCount>135500</ adsm1-bo:EventCount>
</PlacementResult>
<!-- The breakdown of how the ad was distributed is recorded as the
distribution result - the Targeting elements record an intersection of
listed zipcodes with single copy and home delivery distribution methods. -->
```

```

<DistributionResult>
<!-- The total distribution count result is recorded - 135500 ads
distributed. -->
  <adsm-bo:TotalDistributionCount>135500</adsm-bo:TotalDistributionCount>
  <!-- This Targeting element creates an intersection of listed zipcodes
with single copy and home delivery distribution methods. -->
  <adsm-bo:Targeting>
    <adsm-bo:DistributionCount>35000</adsm-bo:DistributionCount>
    <adsm-bo:Target>
      <adsm-bo:Code>
        <adsm:CodeList xsi:type="adsm-
cv:AdsMLTargetingTypeCodeCV">ZipPostalCode</adsm:CodeList>
        <adsm:CodeValue>10001</adsm:CodeValue>
      </adsm-bo:Code>
      <adsm-bo:Code>
        <adsm:CodeList xsi:type="adsm-
cv:AdsMLTargetingTypeCodeCV">ZipPostalCode</adsm:CodeList>
        <adsm:CodeValue>10002</adsm:CodeValue>
      </adsm-bo:Code>
      <adsm-bo:Code>
        <adsm:CodeList xsi:type="adsm-
cv:AdsMLTargetingTypeCodeCV">ZipPostalCode</adsm:CodeList>
        <adsm:CodeValue>10003</adsm:CodeValue>
      </adsm-bo:Code>
    </adsm-bo:Target>
    <adsm-bo:Target>
      <adsm-bo:Code>
        <adsm:CodeList xsi:type="adsm-
cv:AdsMLTargetingTypeCodeCV">DeliveryClass</adsm:CodeList>
        <adsm:CodeValue>Single Copy</adsm:CodeValue>
      </adsm-bo:Code>
      <adsm-bo:Code>
        <adsm:CodeList xsi:type="adsm-
cv:AdsMLTargetingTypeCodeCV">DeliveryClass</adsm:CodeList>
        <adsm:CodeValue>Home Delivery</adsm:CodeValue>
      </adsm-bo:Code>
    </adsm-bo:Target>
  </adsm-bo:Targeting>
  <!-- This Targeting element creates an intersection of listed zipcodes with
total market coverage delivery distribution methods. -->
  <adsm-bo:Targeting>
    <adsm-bo:DistributionCount>100500</adsm-bo:DistributionCount>
    <adsm-bo:Target>
      <adsm-bo:Code>
        <adsm:CodeList xsi:type="adsm-cv:AdsMLTargetingTypeCodeCV"
>ZipPostalCode</adsm:CodeList>

```

```

    <adsm1:CodeValue>10004</adsm1:CodeValue>
  </adsm1-bo:Code>
  <adsm1-bo:Code>
    <adsm1:CodeList xsi:type="adsm1-
cv:AdsMLTargetingTypeCodeCV">ZipPostalCode</adsm1:CodeList>
    <adsm1:CodeValue>10005</adsm1:CodeValue>
  </adsm1-bo:Code>
  <adsm1-bo:Code>
    <adsm1:CodeList xsi:type="adsm1-
cv:AdsMLTargetingTypeCodeCV">ZipPostalCode</adsm1:CodeList>
    <adsm1:CodeValue>10006</adsm1:CodeValue>
  </adsm1-bo:Code>
</adsm1-bo:Target>
<adsm1-bo:Target>
  <adsm1-bo:Code>
    <adsm1:CodeList xsi:type="adsm1-
cv:AdsMLTargetingTypeCodeCV">DeliveryClass</adsm1:CodeList>
    <adsm1:CodeValue>TMC</adsm1:CodeValue>
    <adsm1:Description>Total Market Coverage</adsm1:Description>
  </adsm1-bo:Code>
</adsm1-bo:Target>
</adsm1-bo:Targeting>
</DistributionResult>

```

Notes:

- o As described above, `Distribution/TotalDistributionCount` and `PlacementTarget/EventCount` serve different purposes. The `PlacementResult` provides information about the kind of advertisement event that has been provided to fulfill the original ad order. The `DistributionResult` provides information about the people/places to which the advertisement has been distributed in order to achieve the placement result. In many use cases the placement and distribution result counts are the same (e.g. Insertions which are published to all readers), but that doesn't have to be the case. In an interactive scenario, for example, a `PlacementResult` of "1000 Click-throughs" and a click-through rate of 1% might be accompanied by a `TotalDistributionCount` of 1,000,000 impressions.

6.16 Provide third party performance verification

Scenario: A publisher provides performance verification for an interactive ad that was specified as a 'Run of Site' ad (anywhere on the publisher's website) for a total of 100,000 impressions. The performance verification is provided by a third-party.

AdsML handling: The publisher's system generates an AdsML ProofOfPublication business message, providing the proof data using a single `ProofOfPublication` element. The publisher, as the provider of the ProofOfPublication message, is recorded as the `ProofingParty`. Meta data

describing the ad's appearance and verifying how it performed is provided by the third-party is recorded in the `AppearanceInformation.Interactive` structure.

The third party is clearly identified as the source of the appearance information by populating the `AppearanceInformation.Interactive/Provenance` structure. Within `Provenance`, the third party is recorded as the `ProvenanceParty`. Additional details about the systems used to generate the proof data, a timestamp for when the data was recorded, and the process point at which the data was taken can be provided using `ProvenanceSystem`, `ProvenanceDateTime` and `ProcessPoint` elements.

The verification of the ad's performance – how many impressions were achieved – is recorded using the `PlacementResult` element within the appearance information. The total number of impressions delivered is recorded as the `PlacementResult/EventCount`.

A detailed breakdown of how the ad was distributed across the publisher's website can be recorded by a `DistributionResult` element. This ad was a simple 'Run of Site' ad, and so this is recorded in a single `Targeting` structure using a single `Target` element with 'Run of Site' as a codified value. The sum of the total distribution can be recorded as the `DistributionResult/TotalDistributionCount`, which in this case is identical to `PlacementResult/EventCount`.

Notes:

- In this scenario where the ad appeared as 'Run of Site' the value of `PlacementResult/EventCount` and `DistributionResult/TotalDistributionCount` are identical. In a more complicated distribution scenario where the ad's appearance had been targeted at specific positions or target user groups, for example, then this detailed targeting and the counts achieved could have been recorded as specific `Targeting` and `Target` levels as required. For details on how such low-level targeting is recorded, see Use Case 'Provide details that describe the targeted distribution of an ad'.
- Because no tearsheet is being provided, the message does not contain a `TearSheet` element.

6.17 Deliver proof metadata in more than one human language

Scenario: A publisher provides multilingual proof of publication metadata to describe the publication of an advertisement. The human readable metadata for the ad is to be provided in French and English. The ad itself was run in English.

AdsML handling: The publisher provides the available proof information, recording the necessary metadata as described elsewhere in these Usage FAQs. French and English versions of the human-readable textual metadata is recorded using the repeatable `adsml:Description`, `adsml:DescriptionLine` and `adsml:Note` elements found at the 'advertisement booking information', 'appearance information' and 'tear sheet' structural levels of the `ProofOfPublication` message. Each language variant is recorded in a separate element and the human language is identified using the `xml:lang` attribute provided by the element's internationalization (i18n) attributes. For example, the 'appearance information' might contain two `adsml:Note` elements,

one in French and the other in English and identified as such by their `xml:lang` attributes.

Note that multilingual versions of human readable metadata can be recorded wherever textual elements are repeatable and have `i18n` attributes. For example, elements taking codified values or user-defined properties have repeatable `Description` labels to support multilingual metadata.

Notes:

- The fact that the ad content was itself in English language may or may not be recorded in the metadata if necessary – for example, as part of a description or note in the human-readable textual information or as a user-defined property.
- Multilingual metadata should not be confused with multilingual versions of advertisement content. Multilingual versions of advertisements will always be treated as different ads with different ad content: if an ad is published in French and English versions, then a separate `ProofOfPublication` message is used to provide proof for each language version. The same logical constraint applies to `TearSheet(s)` provided within that `ProofOfPublication` message: a tear sheet will always be for one set of ad content (i.e. an ad in a single language) but the tear sheet may contain multilingual metadata.
- The `i18nAttributes` group provides `xml:lang`, `dir` and `source` attributes. In a simple usage scenario only the `xml:lang` attribute will always be used when content is provided in more than one human language. The `dir` and `source` attributes support scenarios where the reading direction of the text is specified and where a language variant is being identified as the original source language from which other language translations have been derived. See the *AdsML Type Library Specification & Schema* and *E-Commerce Usage Rules & Guidelines* documents for more information about the `i18nAttributes`.

7 Configuration checklist

In order to facilitate implementation and interoperability, pre-defined packages of features and functionality are a valuable tool. Please see the “*E-Commerce Usage Rules and Guidelines*” document for a general discussion on this subject.

The following packages of features have been defined to date. Some of them provide options which directly affect the technical capabilities of the sending and receiving systems (for example, the ability to send binary content in-line in a message). Others reflect important choreography choices that need to be agreed between trading partners when they are establishing AdsML communications.

Each package consists of either:

- A set of hierarchical levels from which one must be selected (represented by a numbered list), or
- A set of non-exclusive options from which any combination can be selected (represented by a bullet list), or
- A list of mutually-exclusive choices from which one must be selected (represented by a textual description).

The packages are presented in alphabetical order. There is no implied hierarchy.

7.1 Conveyance of binary objects – by inline or external transmission

Trading partners must agree how they support the exchange of binary objects:

- Inline transmission – directly included in the message as an enclosure
- External transmission (‘Out of line’) – materials are external to the message and transmitted by a specified delivery method.

For example, external transmission methods may include a reference to a file located on a website for download, a reference to a CID url in a MIME multipart/related package, etc.

7.2 Use of encoding and encryption of digital tearsheet payload

Trading partners must agree on whether or not they support content packaging, and at which level:

- Encoding of inline content
- Encoding of external linked data
- Encryption of inline content
- Encryption of external linked data

7.3 Conveyance of login and password information

Trading partners wishing to use digital download mechanisms to retrieve tearsheets should agree on how to exchange the necessary information, including

passwords, and in particular which of the following types of information will be explicitly conveyed in the AdsML messages:

- Web address from which to retrieve the tearsheet (e.g. a URL or FTP location).
- User name, e.g. "Jan Janssen", conveyed in retrieval instructions
- Tearsheet retrieval ID, a string which uniquely identifies the tearsheet for retrieval purposes and is not necessarily the same as the tearsheet's AdsML rendering ID, conveyed as a retrieval instruction.
- Password to log into the tearsheet retrieval system, conveyed as a retrieval instruction.

7.4 Multilingual metadata

Trading partners must agree on whether or not they support the provision of alternative versions of human-readable textual metadata in more than one language. (For example, alternative versions of a description or note can be provided, each in a different language.)

If multilingual metadata is supported, trading partners need to agree on:

- Which languages will they use in their messages?
- Which language, if any, takes priority as the 'default language' of the message?
- Any rules for processing and presenting multilingual content to users.

8 References – (Non-Normative)

- [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 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 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>)

9 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.