



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

Document Authors: AdsML Technical Working Group

Document ID: AdsML2.0-FrameworkOverview-AD-3

Document File Name: AdsML2.0-FrameworkOverview-AD.pdf

Document Status: Approved Document

Document Date: October 1 2006

Draft Number: 3

Table of Contents

1	ADSML STANDARD DOCUMENTATION	3
1.1	DOCUMENT STATUS AND COPYRIGHT	3
1.2	NON-EXCLUSIVE LICENSE AGREEMENT FOR ADSML CONSORTIUM SPECIFICATIONS	3
1.3	ADSML CODE OF CONDUCT	5
1.4	DOCUMENT NUMBER AND LOCATION.....	6
1.5	AUDIENCE.....	6
1.6	PURPOSE OF THIS DOCUMENT.....	6
1.7	ACCOMPANYING DOCUMENTS	6
1.8	CHANGE HISTORY	6
1.9	ACKNOWLEDGEMENTS	7
2	THE ADSML FRAMEWORK.....	7
2.1	INTRODUCTION	7
2.2	WHAT IS THE ADSML FRAMEWORK?	8
2.2.1	<i>Implement only what you need.....</i>	<i>9</i>
2.2.2	<i>Key Concepts – Quick Start.....</i>	<i>9</i>
2.2.2.1	Purpose	9
2.2.2.2	Formats	9
2.2.2.3	Envelopes, Items and the AdsML Processor.....	10
2.2.2.4	Item-level information	10
2.2.2.5	Envelope routing.....	11
2.2.2.6	Trading Partner Agreement – The TPA	11
3	THE ADSML ENVELOPE.....	11
3.1	INTRODUCTION	11
3.2	ENVELOPE OVERVIEW	12
3.2.1	<i>Why this approach?</i>	<i>13</i>
3.3	CONFORMANCE.....	14
3.4	TRADING PARTNER AGREEMENT.....	14
3.5	ILLUSTRATIVE USE CASE.....	15
4	ACHIEVING INTEROPERABILITY	17
4.1	BUSINESS PROCESSES AND TERMINOLOGY	17
4.1.1	<i>Glossary.....</i>	<i>18</i>
4.2	MESSAGE CHOREOGRAPHY	18
4.2.1	<i>The ‘Advertising Component Interactions’</i>	<i>18</i>
4.2.2	<i>Item-level message choreography.....</i>	<i>18</i>
4.3	REGIONAL CUSTOMIZATION OF THE ADSML MESSAGE FORMATS	19
4.3.1	<i>Controlled Vocabularies.....</i>	<i>19</i>
5	ADSML E-COMMERCE SPECIFICATIONS.....	20
5.1	ADSML BOOKINGS.....	20
5.2	ADSML MATERIALS	21
5.3	ADSML FINANCIALS	21
5.4	ADSML STRUCTURED DESCRIPTIONS.....	21
5.4.1	<i>Interchange format and non-technical aspects.....</i>	<i>22</i>
5.4.2	<i>Sample vocabularies.....</i>	<i>22</i>
5.5	RELATIONSHIPS BETWEEN THE ADSML E-COMMERCE SPECIFICATIONS.....	22
6	ADDITIONAL INFORMATION.....	22
6.1	WHY XML	22
7	APPENDIX A: ACKNOWLEDGEMENT FOR CONTRIBUTIONS TO THIS DOCUMENT	23

1 AdsML Standard Documentation

1.1 Document status and copyright

This is a Approved Document of the AdsML Framework Overview.

Copyright © 2006 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 AdsML2.0-FrameworkOverview-AD-3, is freely available. It is located at the AdsML website at <http://www.adsml.org/>.

1.5 Audience

The intended audience for this document is any prospective user of AdsML, interested parties, and the AdsML Consortium.

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

1.6 Purpose of this document

This document provides a management-level overview of the AdsML Framework. It is intended to provide the reader with a high-level understanding of AdsML's business and technical dimensions.

1.7 Accompanying documents

This document is part of the AdsML Framework, which contains a suite of related documents. A description of the entire document set can be found in the *ReadMeFirst* html file associated with this release of the Framework.

In particular, before reading this document it is recommended that you familiarize yourself with *The Vision of AdsML*, a non-technical overview of the entire AdsML project, and after reading this document, the logical next step for implementers would be to read the *E-Commerce Usage Rules and Guidelines*.

1.8 Change History

Draft	Date	Changes	Author
AD 3	October 1 2006	<ul style="list-style-type: none">• Updated AdsML references to reflect Registered Trademark status	TS

AD 2	June 1 2006	<ul style="list-style-type: none"> Converted from "specification" to "document" format because this document is informative rather than normative. Added "implement only what you need" section This draft was Approved by the technical working group 	TS
PS 1	30 November 2005	Created this document by merging portions of the AdsML1.0-FrameworkOverview-AS and AdsML2.0-FrameworkOverview-PD, and updating the result in order to reflect current activities. Both of the source documents are being removed from the Framework and replaced with this consolidated overview.	TS

1.9 Acknowledgements

This document is a product of the AdsML Technical Working Group.

Primary authorship and editing was performed by:

- Tony Stewart (RivCom Limited) - tony.stewart@rivcom.com
- Christian Ratenburg (CCI Europe) - cr@ccieurope.com
- Jay Cousins (RivCom) - jay.cousins@rivcom.com

Portions are based on material written by:

- John Iobst (NAA) - iobst@naa.org
- Ulf Wingstedt (CNET/Svenska) - ulf.wingstedt@cnet.se

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

2 The AdsML Framework

This document provides a management-level overview of the AdsML Framework. It is intended to provide the reader with a high-level understanding of AdsML's business and technical dimensions.

Readers of this document are assumed to be familiar with *The Vision of AdsML*, a non-technical overview of the entire AdsML project.

2.1 Introduction

The AdsML Framework of E-commerce Business Standards consists of a growing set of standards that are designed to work together to solve specific business problems in the advertising workflow. These standards exist at two levels:

- the "AdsML Envelope" standard defines an XML delivery envelope that can convey any number of business messages between two trading partners;
- a set of "Item-level" e-commerce standards define XML message formats for specific types of information or transactions, for example, insertion orders, invoices or artwork.

These standards are part of the AdsML "Framework". In practice this means that they share many of the same design principles, they use common element and attribute names and structures, and they support a common message choreography (i.e. the pattern by which messages are exchanged between trading partners).

Note: The AdsML Item-level standards are so called because they define message formats that can be used as "Items" in the AdsML Envelope. Use of the AdsML Envelope, while encouraged, is optional, and the Item-Level standards can be used both inside and outside of the AdsML Envelope. The information in this document remains valid regardless of whether or not the Items are conveyed in an AdsML Envelope.

2.2 What is the AdsML Framework?

At the business level, AdsML helps business partners to understand and identify at which points in the advertising lifecycle they are able to effectively conduct business with one another by electronic, system-to-system communications. At the technical level, AdsML facilitates the exchange of the information required to conduct business electronically.

AdsML therefore supports the 'what' and 'how' of advertising, and by so doing helps business create the trading partner agreements that regulate the conduct of business and business partnerships. At a high level, this is the primary purpose of AdsML: to facilitate the business and technical co-operation between partners in the advertising supply chain.

But what exactly do we mean by "the AdsML Framework"? How does AdsML achieve these ambitious goals?

As with many things, it is easier to begin by saying what AdsML is *not* to help us understand what it *is*.

AdsML is *not* a 'rigid' framework. It is *not* an inflexible structure that resists change, and it is *not* prescriptive: AdsML does *not* dictate the way you do business or run your workflows, or the specific electronic format(s) you must use to exchange information with your partners; and it does *not* impose a specific way of running your advertising workflow. AdsML is *not* static and determined by today's here and now of the business moment; it will *not* be out of date tomorrow.

If that is what AdsML is *not*, then what *is* AdsML?

AdsML *is* there to support businesses exchange data throughout the entire advertising and advertisement lifecycle, across all media.

AdsML *is* there to help businesses:

- choose which of the existing advertising standards and formats are the best option for them to use at the various stages of the advertising workflow.
- identify and agree upon the standards that they use
- verify that the data they exchange conforms to their expectations and agreements with their trading partners
- implement efficient, cost-effective e-commerce communications across the entire advertising lifecycle.

AdsML *is* flexible and is intended to be adaptive to changing business requirements – it *is* intended to evolve as business and the technology supporting that business develop.

2.2.1 Implement only what you need

In order to implement AdsML-based e-commerce, trading partners and their vendors (or industry associations acting on their behalf) are expected to review the AdsML Framework and decide which parts of it they will implement in order to solve their particular business problems. AdsML provides many tools to support this process. Further information 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 common technical components and a common approach to advertising e-commerce that makes them "AdsML".

2.2.2 Key Concepts – Quick Start

Here is an overview of some of the key concepts underlying the AdsML Framework. Each of these is then described in more detail later in this document.

2.2.2.1 Purpose

The technical purpose of AdsML is to enable the efficient exchange of advertisement-related information between software systems in electronic form through the complete advertising lifecycle. The focus is on facilitating communications between systems at different organizations, but it is expected that in many cases AdsML will also be used between systems within the same organization.

2.2.2.2 Formats

There are many different standards or formats that can be used for the electronic exchange of advertising information. These include advertising-specific XML standards like NAA's CREST, Ifra's IfraAdConnexion and AdsMLBookings, non-XML EDI standards that have been developed over the years, or even comma-delimited formats that are used privately for point-to-point communication between systems. Therefore, for any given type of advertising information (such as an Insertion Order, an Invoice or a digital artwork file), there may be several standards or formats that are currently being used somewhere in the world.

When two trading partners decide to exchange advertising information electronically, they must agree at least on the types of information they will exchange, and on the standards or formats that they will use to represent that information. This information is recorded in a Process Partnership Agreement (PPA) between the two trading partners, which in turn is part of their overall Trading Partner Agreement (TPA). While much of the TPA may never even be formalized on paper, the Process Partnership Agreement must be recorded, because it is used by their communications software (including their AdsML Processor, if they choose to implement one) to control many of the interactions between their systems.

The *Advertising Component Interactions Analysis* provides a reference context that enables trading partners to agree on the types of information – the business messages – that they will exchange, and the standards and formats they will use when doing so.

The AdsML Framework does not restrict the formats that trading partners can use when exchanging information. Instead, it provides an inclusive packaging

mechanism (the AdsML Envelope) that can accommodate any format that can be safely placed inside an XML document, and a growing set of item-level standards to communicate specific types of information (e.g. AdsMLBookings and AdsMLMaterials).

2.2.2.3 Envelopes, Items and the AdsML Processor

The AdsML Envelope and its associated Processing Model define a packaging mechanism and supporting technology that efficiently conveys the different business messages between trading partners, and provides substantial additional functionality as well.

Each individual business message carried inside an AdsML Envelope is called an 'Item'. Each item is carried inside the AdsML Envelope in its own native format. For example, a Booking could be carried inside an AdsML Envelope as an AdsMLBookings document, or an IfraAdConnexion document, or in any other appropriate format. A single Envelope may carry many Items of various types and formats.

The AdsML Envelope treats the Items that it conveys as "black boxes" that it does not understand and cannot read or manipulate. AdsML Envelopes are created and manipulated by an AdsML Processor (software that creates and manipulates an AdsML Envelope). An AdsML Processor only has access to an extremely small amount of metadata about each Item; just what is required to ensure that it can provide its stated service.

Use of the AdsML Envelope to convey item-level business messages is optional, but strongly encouraged.

The AdsML Consortium does not create software. The AdsML Framework provides a set of standards and guidelines which are meant to be implemented by vendors and IT departments as they create new applications or modify existing ones.

2.2.2.4 Item-level information

While Envelopes provide an optional packaging mechanism, the items within them contain business information which is the lifeblood of the advertising workflow.

Item-level information typically is generated by an application in one organization and sent to an application in another organization. Both of these applications are likely to be connected to databases, where the information is stored. So, the lifespan of an item-level message typically spans from the moment at which a set of information is retrieved from a database and assembled into the appropriate message format, to the moment at which the receiving organization receives the message, disassembles it into its component pieces, and stores them in another database to await further processing.

In order for two organizations to exchange item-level messages, those messages must conform to an agreed standard, and further, the two organizations must have configured their implementations of that standard in the same way.

AdsML provides a set of item-level standards and is developing additional standards on a regular basis. More information about these standards is provided later in this document.

2.2.2.5 Envelope routing

Each AdsML Envelope is sent by a single Sender to a single Recipient, after which (in most cases) the Recipient removes the Items from the Envelope and the Envelope ceases to exist. Thus, usually the Envelope is only used for a single 'hop' between two trading partners.

In some cases, an Item may travel a longer and more complex path, traversing multiple trading partners (via agents, aggregators or other participating intermediaries) between its originator and its ultimate consumer. As a result, an Item may move through many AdsML Envelopes during the course of its life. During any given 'hop' of its journey, the Item will share an AdsML Envelope with other Items that happen to be going from the same Sender to the same Recipient at that moment. In the bigger picture, however, the Items in a given Envelope may have been originated by various different senders further up the communications chain, and may be intended for various different consumers further down the chain.

2.2.2.6 Trading Partner Agreement – The TPA

The Trading Partner Agreement (TPA) is the legal undertaking establishing the terms and conditions under which trading partners will conduct business with one another and that the trading partners agree to abide by.

A TPA can be a formal set of rules that are used by software systems directly, or an informal agreement about how to conduct business messages interchange.

The content of actual instances of messages is only partly defined by the AdsML specifications. In order to provide necessary flexibility in how to design and implement business processes for many different organizations, a certain amount of optionality is required with respect to which data to include in messages. This optionality is supported in particular by the use of "controlled vocabularies", which allow trading partners to define the terms that can be included in the messages that they exchange.

In cases where two trading partners agree on more specific rules for the use of the optional elements and attributes, including controlled vocabularies that they establish, these rules constitute a natural part of the TPA between these trading partners. However, it should be noted that a TPA must never overrule the definitions in the AdsML specifications.

3 The AdsML Envelope

3.1 Introduction

The AdsML Envelope is used for exchanging and sharing information – business messages – while executing business processes during the advertising lifecycle. To achieve this exchange requires the establishment of a defined choreography between information senders and receivers. This will allow for the proper level of communication among all of the partners and make the business work smoothly.

The AdsML Envelope standard supports a send and query/response model. The standard provides 'envelope' messages for sending information, with the information contained as 'Item(s)' within the envelope, and 'envelope response' messages for responding to 'envelopes' that have been received.

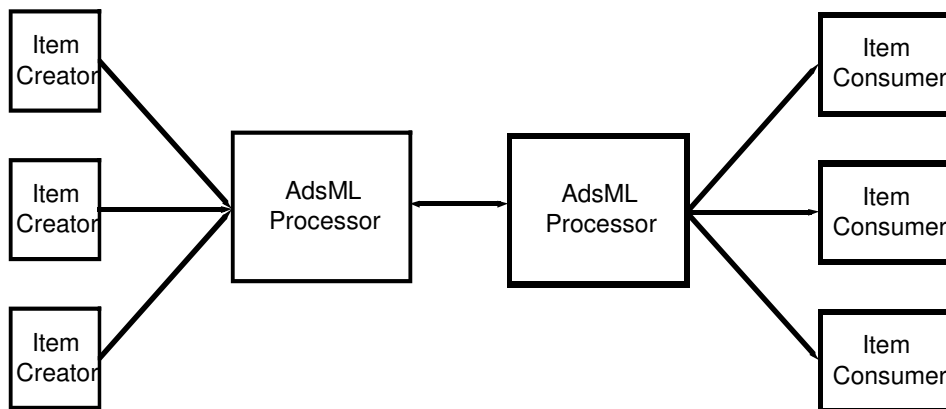
Within a given organization, there will be a number of software systems that are each capable of processing one or more of the Item types that AdsML normally transmits – for example, booking systems, publishing systems, accounting

systems, etc. From the perspective of AdsML, these software systems are “Item-level” applications, because they create and consume the Items that are transmitted inside the AdsML Envelopes.

So, a given organization is likely to have a number of Item-level applications. These Item-level applications need to send information both between themselves within the organization, and also externally to Item-level applications at other organizations with which they do business.

The same organization will probably have only one AdsML processor. The AdsML processor is a software system¹ designed to process AdsML Envelopes, where “processing” includes creating, transmitting, receiving and responding to them. It will usually be positioned close to the gateway between its organization’s communications systems and the outside world².

As shown in the illustration below, when an organization wishes to send advertising content Items to its trading partners, the Item-level applications within the organization generate the appropriate information objects, wrap them in some AdsML-specific metadata, and send them internally to the organization’s AdsML processor for packaging. The AdsML processor receives these Items from the Item-level applications, packages them into one or more AdsML Envelopes (each of which is addressed to a single external organization), and then sends each AdsML Envelope to its recipient organization.



The recipient organization’s AdsML Processor receives each incoming AdsML Envelope, un-packs it into its component Items, and then routes each of these Items internally to the appropriate Item-level application. Each Item that arrives at a recipient’s internal Item-level application is effectively identical to the Item that was created by the sender’s Item processor. At this point the AdsML Envelope no longer exists. It served its purpose by transporting the Items from the sender organization to the recipient organization.

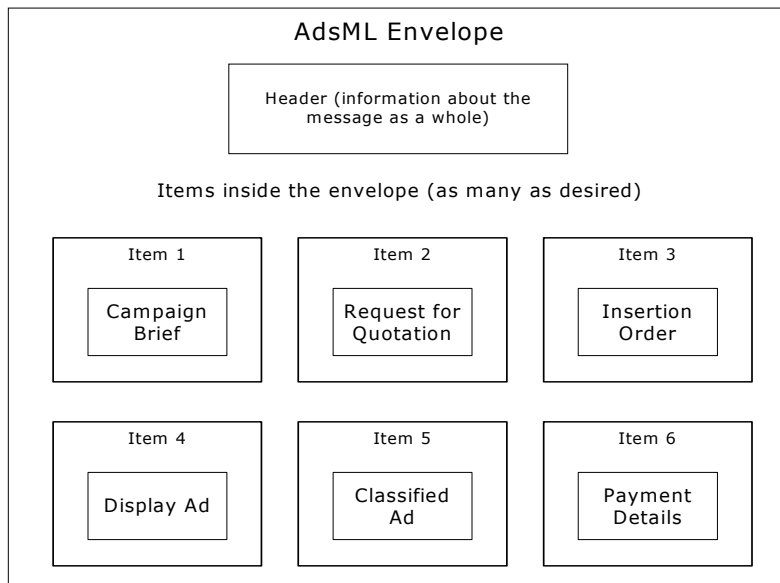
3.2 Envelope overview

Conceptually, the AdsML Envelope serves as a “wrapper”, or container, that facilitates the exchange of messages (such as those defined in the Advertising Component Interactions) between trading partners in the advertising lifecycle. While it is possible that the AdsML Envelope may be used to improve communication between internal systems within a given organization, its primary purpose is to facilitate external communications between an organization and its trading partners.

¹ The AdsML Consortium does not provide software, including AdsML Processors.

² It is also possible to use AdsML for communications between systems within a single organization.

Structurally, the AdsML Envelope schema defines an envelope consisting of a header and a body. The header contains information necessary to route, process, and respond to the AdsML message, and the body contains the advertising data content itself, in the form of one or more AdsML Items. The Item content can be encoded either in one of the existing advertising industry XML standards, or in a non-XML format such as an EDI standard or even a comma-delimited file. The only requirement is that the parties that are exchanging the information need to agree in advance on the format that they will use to exchange this type of information, and the format has to be able to be placed inside an XML document.



Consequently, the AdsML envelope serves multiple goals – enabling process automation, allowing continued use of existing investments in technological infrastructure, and the provision of a common interface to facilitate systems integration.

The *Envelope Processing Model, Addressing and Operational Conformance* documentation describes the types of processing that the AdsML Envelope was designed to support. It provides diagrams and descriptions of the components of an AdsML Processor, along with examples of how some of the key elements and attributes in the AdsML Envelope can be used to support common types of message processing.

3.2.1 Why this approach?

The advantage of the AdsML Envelope approach to message handling is that it helps businesses integrate and handle the complexity of exchanging information between multiple systems and multiple business partners. Rather than having to define multiple 'point to point' integrations with all of their trading partners, each partner handles this integration by implementing a single AdsML interface.

The complexity of integration – i.e. of how many formats and types of Items you are likely to receive – is controlled by the use of the AdsML Envelope interface, and by the Process Partnership Agreement (PPA) determining the information that can be sent to you.

The control is flexible at business and technical levels – the business can expand with the evolution of its PPAs. When the business requires the introduction of further systems to handle new types of information, then the single AdsML interface to the outside world remains, while giving it the freedom to extend the

functionality of its AdsML processor and of the Item-level applications to which that processor is connected.

3.3 Conformance

The AdsML Framework includes a definition of what is required for a system built to process AdsML Envelopes to be AdsML conformant. AdsML conformance is provided at core and enhanced levels, enhanced levels successively graded by their provision of additional functionality. Providing multiple 'conformance levels' allows basic and advanced AdsML systems to be developed according to the business requirements of the individual usage scenario, enabling a system applicable to the individual circumstances to be installed while providing a basis for incremental future development.

3.4 Trading Partner Agreement

One of the fundamental concepts in AdsML is that any two organizations wishing to exchange advertising information with each other will first establish an explicit or implicit Trading Partner Agreement (TPA). A portion of the TPA, known as the Process Partnership Agreement (PPA) is then recorded in machine-readable form³ and used to control certain aspects of AdsML Envelope and Item processing. As part of this agreement, the parties will agree upon (at least) the following:

- The types of business messages that they will exchange between each other's systems (e.g. Mediapack, Insertion Orders, Advertisements, Invoices, etc.)
- The standard or format that will be used to represent each such type of message (e.g. AdsMLBookings, AdsMLMaterials, IfraAdConnexion, a non-XML EDI standard, etc.) and the supported version numbers
- The transport mechanism that should be used (e.g. email, FTP or HTTP), and the address to which the information should be sent (e.g. a URL)
- Various other, more technical, details, as described in the AdsML specifications and the processing / addressing / conformance documentation.

Once such an agreement has been established, the AdsML Framework provides facilities that enable the efficient exchange and validation of business messages between the two parties in conformance with the terms of their agreement. This means, for example, that two trading parties can agree to exchange Insertion Orders, Advertisements and Invoices; can specify the XML or non-XML format that will be used for each of these types of information; and then can use the AdsML Framework not only to send and receive the information and route it appropriately within their internal systems, but also to verify that the type of information contained in each message conforms to the relevant agreement between them. It should be noted, however, that validation of the structure and format of each individual business message (e.g. that it is a valid AdsMLBookings message or PDF/X file) is performed not by the AdsML Processor, but by the application that receives and processes the business message that has been transmitted via the AdsML framework.)

While much of the business-level information in the TPA does not have to be recorded in any particular format, it is assumed that the relevant technical information will be stored in a Process Partnership Agreement on each party's system, and that their AdsML processors will be able to retrieve and process this information as required. The choice of whether the information is stored in a database, for example, or in a spreadsheet or XML document, is left up to each implementation.

³ The AdsML Framework does not standardize the format in which a PPA should be stored. It is up to the implementer to decide how best to record this information.

3.5 Illustrative Use Case

The following simplified use case is meant to provide context for understanding the role played by the AdsML Framework. It does not cover all eventualities, and is not intended to constrain implementers to adopt any particular usage or processing model.

Note also that this use case covers the sending of a single message in one direction between two trading partners. It does not represent the full choreography of back-and-forth messages that is required in order to perform most e-commerce activities. These choreographic patterns are discussed in more detail in *E-commerce Usage Rules and Guidelines*, and in the individual AdsML item-level standards.

1. Precondition: two trading partners (for example, an advertiser and a publisher) agree to transmit advertising content and metadata electronically between each other using the AdsML Framework.
 - a. As part of that process, they agree on the types of information that they will send between them, the specific standard(s) and format(s) that they will use to represent that information, the message choreography that they will follow, and whether an envelope response should be sent for every AdsML message that is received (along with any other contractual conditions appropriate to their business requirements). This information is contained in a Process Partnership Agreement that has been agreed between them.
2. To identify each type of information, they look up and reference the name of the relevant message or business object in the *Advertising Component Interactions Analysis* documentation.
3. The advertiser now needs to send an insertion order and a display ad to the publisher. They have agreed to use AdsMLBookings for exchanging insertion orders, and PDF/X for classified display ads.
4. The advertiser's booking system creates the relevant insertion order using the AdsMLBookings format, wraps it in a small layer of AdsML metadata containing the ID of the organization to which this order should be sent, and routes the insertion order internally to the AdsML processor.
5. Simultaneously, another of the advertiser's internal departments prepares the classified display ad, converts it to PDF/X format, wraps it in a small layer of AdsML metadata containing the ID of the organization to which it should be sent, and sends the resulting package to the advertiser's AdsML processor.
6. The advertiser's AdsML processor receives the AdsMLBookings insertion order and the PDF/X of the display ad, recognizes that they are both intended for the same organization, packages them as Items in a single AdsML Envelope, and transmits that Envelope over the Internet to the publisher.
 - a. NOTE: In order to transmit the PDF/X file, it can either be contained within or referenced by its Item.
 - b. NOTE: The Items in an AdsML Envelope are not necessarily related to the same advertisement.
7. The publisher's AdsML Processor validates that the structure of the AdsML Envelope and the contents of its metadata conform to the AdsML standard.

- a. The AdsML processor does not attempt to parse or validate the contents of the Items within the envelope. It treats them as "black boxes."
8. If the Envelope is not a valid AdsML Envelope, the AdsML processor generates an appropriate AdsML Response message and routes the Response to the advertiser.
 - a. The publisher's AdsML processor stops processing and discards the Envelope.
 - b. End of use case.
9. The publisher's AdsML processor verifies that the Items within the Envelope conform to its agreement with the advertiser governing the standard(s) and format(s) that can be used to transmit each type of content between them. (For example, whether it is acceptable to use AdsMLBookings for insertion orders and PDF/X for artwork.)
 - a. NOTE: This verification is based on a comparison between the metadata in the Envelope and the business rules that are stored in the publisher's system, especially its Process Partnership Agreement. As above, the publisher's AdsML processor does not attempt to read the contents of the Items.
10. If one or more of the Items within the Envelope do not conform to the agreement, the AdsML processor generates an Envelope Response document containing an appropriate message about each such item and routes the Envelope Response to the aggregator
 - a. Items that fail this verification test are handled according to the publisher's business rules: the publisher may try to use the information anyway, or may indicate to the advertiser that the offending Items will not be processed.
 - b. All Items that are not processed are removed from the Envelope and discarded.
 - c. If no more Items remain in the Envelope, processing stops and the use case ends.
11. If the Process Partnership Agreement or the publisher's business rules or the "response-required" flag in the Envelope header indicate that the AdsML Envelope must be acknowledged, the publisher's AdsML processor generates an appropriate Envelope Response message (for example, "AdsML Envelope received") and routes it to the advertiser.
 - a. NOTE: As indicated in steps 8-10, although a response to a valid and verified AdsML message (either an Envelope or an Item-level message) is potentially optional, a response to a message containing any form of error is always required.
12. The publisher's AdsML processor unpacks the remaining Items from the Envelope, and based on the metadata contained in the Envelope about each Item, prioritizes and routes each unpacked Item to an appropriate application for that type of Item within the publisher's organization. For example, the AdsMLBookings item would be routed to a system capable of understanding and acting upon AdsMLBookings messages.
 - a. NOTE: Prioritization and routing is based on the metadata in the Envelope. As above, the publisher's AdsML processor does not attempt to read the contents of the Items.
 - b. NOTE: at this point the original AdsML Envelope ceases to exist, as each Item that was contained within it is now routed and processed as a standalone document.

13. Within the publishing organization, each unwrapped Item is transferred into an application that is capable of processing it.
 - a. NOTE: In case the intended application is not capable of natively receiving AdsML item-level messages, a type of software called an AdsML Adaptor can be used to receive the item-level message, extract the information from it, and push that information into the application.
14. The Item-level application performs its own verification and validation tests and generates an appropriate Item-level Response message conforming to the relevant standard, in this case AdsMLBookings.
 - a. NOTE: An AdsML Adaptor can be used again when the application needs to generate the item-level response message.
15. The Response messages are routed to the publisher's AdsML processor, which packages them into one or more new AdsML Envelopes and sends them to the aggregator.
 - a. NOTE: the Envelope message(s) generated at this step is not an "Envelope Response" as defined by the AdsML Envelope standard. From the perspective of the AdsML processor, it is simply a new Envelope containing "black box" Items that have been generated by the Item-level application. The AdsML Processor does not know that these items are themselves responses to a previously-received message.
 - b. NOTE: the item-level response messages are likely to be generated at different times, and therefore packaged in different AdsML Envelopes. The fact that they are responses to content Items that the publisher originally received in a single AdsML Envelope is irrelevant.
16. End of use case.

4 Achieving Interoperability

"Interoperability" is the Holy Grail of e-commerce. When two organizations are able to connect their systems so that they can transfer information between themselves seamlessly and automatically, with all the necessary information showing up in the right places and having the right meanings, then those systems are successfully "interoperating". It is, in short, a technical term for what most people would think of as successful communications between systems.

4.1 Business processes and terminology

The first step towards achieving interoperability is to ensure that all parties exchanging messages speak the same language. For this purpose, AdsML defines a set of high-level business processes and system messages, as well as a glossary of terms. These are documented in the *Advertisement Components Interactions Analysis* and *AdsML Glossary*.

Any set of trading partners who are setting up electronic communications are encouraged to read these documents and use them as a basis for the discussions about which messages to exchange and what their information content should be. By referencing these AdsML-defined message names and glossary terms, rather than whatever terminology is used within their own organizations, trading partners will reduce misunderstandings and ensure that when they reach agreement about a message or process, they are speaking the same language.

4.1.1 Glossary

A glossary of terminology provides a set of standard advertising language by defining the meaning of specific words or 'terms' used in the AdsML Framework, fostering understanding and co-operation between trading partners by defining what is effectively a 'common language'. Each term has an accompanying definition and, where appropriate, synonym(s).

4.2 Message choreography

In order to ensure interoperability, it is essential that systems exchanging AdsML messages have the same view on which messages to send, and which to expect to receive.

4.2.1 The 'Advertising Component Interactions'

At a technical level, the *Advertising Components Interactions Analysis* provides names and codes that uniquely identify each type of business message that is included within the scope of the AdsML Framework. These codes are used within AdsML messages to identify the type(s) of advertising information included in each message, and therefore to support the automatic verification and routing of that information.

But perhaps more importantly, the *Advertising Component Interactions Analysis* provides a common model of the business processes involved in the lifecycle of an advertisement, and defines the types of business messages that are exchanged between advertising system components at each stage of the lifecycle.

The Advertising Components Interactions documentation is not prescriptive. Rather, it provides a 'higher level' common model that identifies the key components and phases of the advertising lifecycle while remaining above the workflow level, and in particular identifies the types of information that are exchanged, and used by, these processes and their associated workflow-level activities.

The benefit of defining Advertising Component Interactions in this way is that each stage of the advertising lifecycle and its associated business messages have been identified and modeled, ensuring that the AdsML Framework is based on thorough and rigorously researched coverage of the advertising lifecycle. AdsML users can use the Component Interactions as a reference point for locating and understanding their own business in the context of the 'big picture' background of the whole advertising lifecycle.

With this contextualization, users can identify the business messages, the types of data (e.g. technical specifications, production content) that their systems will need to receive and send, and the types of processes that they will need to integrate and interact with as they conduct business with trading partners. The Advertising Component Interactions therefore provides a model of the advertising lifecycle that a business can use to understand their own (and that of their trading partners) position in that lifecycle, and the business data and processes required to support that position. Consequently, the process model can be used to support business development and integration.

4.2.2 Item-level message choreography

Item-level choreography defines the pattern of back-and-forth communication between the applications in each organization. This choreography is usually much more complex than the AdsML Envelope choreography, because these Items are the lifeblood of the advertisement supply chain, and their interchange defines and supports the organization's business processes. Every organization will have a

number of rules indicating under what circumstances different types of business messages must be sent, acknowledged, queried, refused, altered, etc. The interplay of such rules between two organizations results in an intricate dance of messages flowing back and forth.

The AdsML Framework exists to support this Item-level message choreography, but the AdsML processor remains completely unaware of it. The AdsML processor and the AdsML Envelope format treat the Items that they carry as “black boxes”, and handle them in exactly the same way regardless of whether they are advertising content, queries, responses, acknowledgements, etc. The AdsML processor acts on instructions received from the Item-level applications in a relatively simple way, adding value through its role as a form of “smart transportation”, but not interacting with the underlying business processes in any other way. All of the logic and rules governing the flow of these messages must reside in the Item-level applications that create and consume them.

4.3 Regional customization of the AdsML message formats

Part of the philosophy of the AdsML Consortium is that all of the AdsML specifications must be global in scope, and should be as interoperable as possible both within and across regional boundaries. In practice, this means that each AdsML specification defines an interoperable structure that can be further customized by appropriate regional groups (whether formal industry associations or groups of trading partners that constitute de-facto regional associations), in order to ensure that the specification as implemented in a given region supports local vocabularies and business practices.

Finding the best balance between fixed and variable components in each specification is difficult, and as much art as science. The more flexibility is provided for regional customizations, the greater the danger that regional customizations of the specification will not be interoperable with each other.

The AdsML Framework provides three important tools to help users manage create regional customizations: Configuration checklists, Controlled Vocabularies, and Profiles. These can then be used as building blocks when assembling a formal Trading Partner Agreement.

A non-technical introduction to controlled vocabularies is provided below. Further information about all aspects of customizing AdsML to meet regional or domain-specific requirements can be found in *E-Commerce Usage Rules & Guidelines*.

4.3.1 Controlled Vocabularies

The approach taken to enable regional customization is primarily through the use of controlled vocabularies, commonly called code fields. While the message structure of each specification is tightly fixed, flexibility has been provided to allow regional groups to extend or replace many of the AdsML controlled vocabularies.

For example, one of the rules of the AdsML Envelope is that the party sending an AdsML Item must specify the Type of business message contained in that Item (e.g. an “Insertion Order” or an “Invoice”), and the message format that has been used for it (e.g. “AdsMLBookings” or “IfraAdConnexion”). The specific text strings that can be used to convey this information are tightly controlled: they have to appear in a list of values that the sender and recipient of the message have agreed to use. By restricting the allowed values to terms from an agreed-upon list, the recipient’s AdsML Processor can automatically verify that the Types of Items contained in an Envelope, and the Standards used for them, conform to

the relevant Process Partnership Agreement and can be processed by the recipient's systems.

Another example *outside of the context of AdsML* is a list of the colors that could be used to describe the physical features of an animal, for example:

Colors of a cat
Black
Grey
White
Brown
Red

These lists of allowed values are called "controlled vocabularies". Each controlled vocabulary contains a set of terms that can be used for a specific purpose within the message. AdsML defines specific 'AdsML' controlled vocabularies for use in AdsML messages, and provides a mechanism to validate that the values in a given message come from the appropriate controlled vocabularies.

In addition to using the default AdsML controlled vocabularies, AdsML implementers are also able to define their own controlled vocabulary values, in order to accommodate situations when trading partners have specific terms that they want to use as controlled vocabulary values in their AdsML messages. In particular, it is anticipated that regional groups will define controlled vocabularies that trading partners doing business in their region will be required to use when exchanging AdsML messages

Making such additions is called "extending" the controlled vocabulary. AdsML provides a mechanism for extending each of the controlled vocabularies that govern the information in an AdsML message, although whenever AdsML has supplied a default controlled vocabulary, we strongly caution against extending a given vocabulary and urge all users to use the default supplied terms. (The details of the extension mechanism are described in *E-commerce Usage Rules and Guidelines*.)

5 AdsML E-Commerce Specifications

The AdsML Framework includes specifications for XML message formats that can convey many important types of advertising information, and work is underway to develop specifications for other parts of the advertising workflow. This section contains a brief overview of each of the standards that has been completed or is currently being developed. Detailed information about each AdsML standard can be found in its own Specification document and Usage guide.

5.1 AdsML Bookings

The AdsML Bookings standard defines an XML format for exchanging ad booking information that can serve as a replacement for the booking capabilities of the SPACE/XML, IfraAdConnexion 2.0 and CREST 2.0 standards. The first release of this format fully supports bookings for display, classified and insert advertisements in print publications (including both newspapers and magazines), and contains initial, lightweight support for bookings in other media. The standard is designed to handle the full lifecycle of a booking, from tentative reservations through firm bookings, including any changes or cancellations along the way.

The standard is also designed to support the later addition of other media types, including broadcast and outdoor advertising.

5.2 AdsML Materials

The AdsML Materials standard defines an XML format for transmitting formatted or unformatted advertisement content. ("Unformatted" content consists of metadata that can be used to generate a generically formatted advertisement, for example, the list of things being sold in a classified ad.) The content may consist of any combination of XML documents with accompanying stylesheets or templates, and/or non-XML media files such as PDF/X, EPS, JPEG, GIF, MPEG, etc. It may be transmitted in the same physical package as the Ad Materials delivery message, or transported by other means, in which case the Ad Materials delivery message provides information about how to identify and/or retrieve the ad content.

This standard is intended to support two contrary scenarios: situations in which the ad content arrives on its own, possibly even before its related booking information; and situations in which ad content is being sent as part of a "package" of related information – such as the booking, content and categorization for a set of classified ads – that will all be processed together. This latter scenario is common for certain types of classified ads, and extremely uncommon for most other advertisements.

5.3 AdsML Financials

As of this writing (November 2005), the Technical Working Group has just begun to develop a set of AdsML Financials messages.

The AdsML Financials standard will define an XML format for an electronic invoice that can be transmitted by publishers to their customers (agencies and advertisers) in place of the current paper invoices. This will be supported by best practices and workflows for replacing printed tearsheets with an electronic proof of publication, so as to automate order reconciliation, claim handling and payment authorization processes. XML formats for electronic messages to transmit claims and payment notifications will also be created.

The AdsML e-invoicing project will build on the work previously done for the AdsML Bookings format as well as similar projects for other industries. The goal is to create a fully automated electronic "round trip" workflow, from an initial insertion order through to invoicing, reconciliation, and payment for that order. This will eliminate transaction costs for all parties, reduce the level of claims and queries, and enable invoicing at a more granular level (i.e. more invoices containing fewer orders per invoice), which should improve cash flow by ensuring that only orders with claims against them are held up for payment.

5.4 AdsML Structured Descriptions

The AdsML Structured Descriptions of Advertisement Objects standard defines an XML format and approach for describing in a structured, machine-processable way the objects that are mentioned in an advertisement. This includes both the products and services that are being offered or sought, and also secondary objects relevant to the advertisement such as the location of offering, keywords and the advertiser's reply-to details. The goal of this activity is to support the automated classification, syndication and aggregation of advertisements by multiple publishers, and then searching and querying of the contents of those ads by potential consumers. While the format is specifically required to be able to convey the description of items in classified ads, it is capable of describing the advertised items in any kind of advertisement in any medium.

5.4.1 Interchange format and non-technical aspects

This standard includes both an XML interchange format, and also a relatively non-technical mechanism by which domain experts can define controlled vocabularies of terms and measures (including constraints such as data types, value ranges, and lists of allowed values) that can be used to describe the features of specific types of things offered for sale in advertisements, for example, Cars or Houses. The mechanism includes tools and techniques to support both the business process of agreeing on the terms and constraints that will be used in a specific vocabulary, and the technical process of converting those terms and constraints into a structured AdsML vocabulary. Once an AdsML vocabulary has been created, it will be possible to use that vocabulary to support data-entry routines in ad order processing software (for example, by providing picklists of allowed values) and to validate that the information recorded about a given advertisement conforms to the governing vocabulary.

5.4.2 Sample vocabularies

The AdsML Technical Working Group has used these mechanisms to create several model AdsML vocabularies: Housing, Recruitment, Transportation, Travel, Miscellaneous Products and Miscellaneous Services. These AdsML vocabularies build on the information contained in the CREST 2.0 standard, while adding additional domain knowledge. We anticipate that in coming years, we will create model vocabularies to cover other common types of objects mentioned in classified ads.

Advertisers and publishers in a given country or region will be able either to modify the supplied AdsML model vocabularies so that they meet their needs, or to use the AdsML mechanisms defined in this project to create their required vocabularies from scratch.

5.5 Relationships between the AdsML e-commerce specifications

While each of the AdsML E-commerce specifications can be used on its own, they are designed to support additional functionality when used together.

- Ad content can be embedded in or transmitted with an Ad Bookings message, in order to support delivery of ad content at the same time as the booking.
- The Structured Descriptions format can be embedded in an Ad Bookings or Ad Materials delivery message, in order to support transmission of the metadata about an ad in the same package as the ad itself
- At a technical level, the schemas (data definitions) of these formats use shared structures and Types to represent common components.

6 Additional information

6.1 Why XML

AdsML uses XML for three main reasons – *extensibility*, *self-definition*, and *application independence*.

As an XML format, AdsML is inherently *extensible*. Message formats defined in XML can be added to or changed without breaking backwards compatibility, and so will be able to evolve as the AdsML initiative grows and the scope of its coverage broadens.

XML is also *self-defining*. It can be used to describe the structure and meaning of the data content that it carries, and it enables that structure and meaning to be enforced in an XML document by validating the document against a 'schema' defining the allowed structure and meaning of that document.

The structure and meaning of AdsML is defined in a supplied "AdsML schema" that forms a core part of the AdsML standard⁴. The AdsML schema defines the allowed data content, structure, and meaning of the information that is carried by an AdsML document. AdsML documents can be validated against the AdsML schema during routine XML processing of the document to make sure that the right data is present and in the right place. XML validation ensures that the constraints imposed on an XML document by its defining schema are enforced, the validation failing those documents whose structure or content fails to meet that specified by the schema. For example, if the schema says that element 'A' must be of data type 'x', but in a particular AdsML document element 'A' contains data of data type 'y', then that document will fail validation because the incorrect data type is present. Accordingly, AdsML documents can be validated against the AdsML Schema, thereby enabling document content to be tested by using standard XML tools to check that the correct data is present before the data even enters the workflow. This checking creates much more reliable workflows and systems; as an XML document can be reliably interpreted against its schema it is easier to identify problems before they become critical.

Being *application independent* means that XML can be used to exchange data between computer systems irrespective of the platforms those systems are running on. Consequently, XML acts as 'glue' that enables systems based on different computer code to exchange data.

7 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
- Members of the AdsML Technical Working Group,

Acknowledgement and thanks for contributions to the Framework 1 Overview, which was one of the sources of this document, are also due to:

- Members of the AdsML Technical Working Group,
 - Adrian Davis (Quickcut) - adavis@quickcut.com
 - Marcel Dumont (Rosetta) - marcel@rosetta.nl
 - Joe Kirk (Cyber UK Ltd) - joe_kirk@compuserve.com
 - Christian Rohrbach (iware - PubliGroupe)
- Reviewers of the Last Call Working Draft,
 - Martin Bryan (IS-Thought) - martin@is-thought.co.uk
 - Hans Faye-Schjøll (Knowlex as) - schjoell@knowlex.no
 - Kurt Fleckner (CCI Europe)
 - Carsten Nielsen (CCI Europe)
 - Jon Simcox (Oppolis Software Ltd) - jon.simcox@oppolis.com
 - Israel Viente (Vio Worldwide Limited) - israel_viente@il.vio.com

⁴ Although AdsML 1.0 relies on XML Schema for this purpose, use of the XML Schema format is not a fundamental requirement of the standard. For example, a future release of AdsML might use another mechanism to define its structure and contents, if that mechanism were deemed to be a better choice at the time.

Acknowledgement and thanks for contributions to the review of the AdsML Framework as a whole are also due to:

Jens Wilms (ppi Media) Jens.Wilms@ppi.de

Anthony B. Coates. (Director, Information Design, Messaging and Management) abcoates@idmm.co.uk