AdsML® Framework for E-Commerce Business Standards for Advertising

AdsMLMaterials 2.5.0
Part 1
Usage Rules & Guidelines

Document Authors: AdsML Technical Working Group

Document ID: AdsMLMaterials-2.5.0-SpecP1Usage-AS-1

Document File Name: AdsMLMaterials-2.5-SpecP1Usage-AS.pdf

Document Status: Approved Specification

Document Date: 15 April 2010

Draft Number: 1

Copyright © 2010 AdsML Consortium. All rights reserved.

Table of Contents

1 AdsMLMaterials Standard Documentation

1.1 Document status and copyright

This is the Approved Specification of *AdsMLMaterials 2.5 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.

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 AdsMLMaterials-2.5.0-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 AdsMLMaterials standard. AdsMLMaterials is an XML-based language used for encoding and routing advertisement materials transaction messages.

1.6 Audience

The intended audience for this document is primarily user and vendor organizations who seek to implement the AdsML Materials 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 AdsMLMaterials messages to address specific business requirements. A companion document, *AdsML Materials 2.5 Part 2 Specification & Schema*, provides additional rules and guidance for using AdsML Materials 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. AdsMLMaterials 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).

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

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

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

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

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

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

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

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

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, AdMaterial.

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
2.5.0 Approved Specification	15 April 2010	Approved Specification.	JC
2.0.0 Approved Specification	30 May 2008	Approved Specification.	JC
1.0.2-AS-3	1 October 2006	Updated to use Controlled Vocabularies 3.0. No other changes.	UW
1.0.1-AS-2	1 October 2006	Added data overview diagram and message exchange diagrams. Revisions to FAQs for third party delivery service scenarios	TS, JC
1.0.0 Approved Specification	1 June 2006	Approved Specification.	JC

1.9.1 Changes in version 2.5.0

Version 2.5.0 of AdsML Materials is a major upgrade that provides support for component delivery and aligns the specification with versions 2.5 of the AdsMLBookings specification.

The significant changes to Version 2.5 of AdsMLMaterials Usage Rules & Guidelines are in the Use Cases section to,

- Update the Use Case '*Deliver only the structured description of an advertisement*' to align with the new support for delivering artwork components
- Add a new Use Case '*Deliver artwork components for make-up by a Repro House*'
- Add a new Use Case '*Deliver artwork for a specific region*'
- Add a new Use Case '*Deliver multiple renderings for an interactive advertisement*'
- Add a new Use Case '*Deliver ad materials with textual metadata describing the materials 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.adsml.org.

2 Introduction

The AdsMLMaterials standard has been developed by the AdsML Consortium to be a global standard for the exchange of advertising materials. It relies on earlier experience and standards that have been embraced and extended in order to support current advertising business requirements. In addition, AdsMLMaterials 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 AdsMLMaterials standard has been developed by the AdsML Consortium as the preferred approach to handle ad materials delivery.

For AdsMLMaterials, the AdsML Framework provides a messaging infrastructure for delivery of materials messages. The AdsMLMaterials 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.

AdsMLMaterials 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.

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).* A subset of the business messages from the ad materials group are supported in this release of AdsMLMaterials; this subset is sufficient to deliver new and replacement materials, to order the suppression ('kill') of materials already delivered, and to report on the status of a materials delivery.

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

AdsMLMaterials 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 AdsMLMaterials does not require use of, nor support for, the AdsML Envelope standard. The actual transfer of AdsMLMaterials messages can be performed by arbitrary method and software application, with or without the use of the AdsML Envelope. For instance, an AdsMLMaterials message can be transmitted using other envelopes such as ebXML or BizzTalk 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 AdsMLMaterials 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

AdsML Materials 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 Materials supports the set of business messages that belongs to the ad material message group (AM).

There are three main classes of business messages defined in the ACIA, used to manage materials during:

- Content production
- Content transmission
- Copy chasing

In the current release of the AdsML Materials standard only the Content Transmission message group is supported. The Content Transmission messages support the delivery of ad content from the artwork creator to the publisher, either directly or via an intermediate deliverer. (For a diagrammatic representation of the message flows comprising the Content Transmission group, see the relevant section in the '*Advertising Component Interactions Analysis*' document.)

The Content Transmission message class consists of several sets of request-response transaction pairs. For instance, a materials deliverer can deliver materials and a publisher can respond, accepting or denying the materials delivery. (The terminology 'request-response' is confusing at first in the context of materials delivery but becomes clearer if considered more. For example, a publisher can accept or refuse materials delivered to them that fail to meet their technical specifications and so a delivery is actually a request.)

At the start of the content transmission workflow, a materials producer can send a Delivery Order message to a third party deliverer which contains a set of ad materials and instructions to deliver them to one or more recipients. The materials are then delivered to their final destination (a publisher or broadcaster) in a Materials Delivery message.

During the 'monitor delivery' process, a material delivery status might be issued by the publisher to the deliverer or producer and by the deliverer to the producer, to report on the current state of the materials delivery in the deliverer or publisher's systems. A status enquiry message might be sent by the deliverer to the publisher to request the state of the delivery in the publisher's systems.

And finally, a publisher can send a Materials Resend Request message if something has happened to previously received materials and they need to be replaced.

3.1 Message components

The main components of the Content Transmission messages are:

- The ad contents (i.e. the digital file to be used by the publisher; commonly referred to as the 'content', 'artwork', 'materials').
- Optionally, alternative renderings of the file such as low-res or thumbnail versions

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

- Optional preflighting information, including the preflighting tool and profile that were used for the rendering file
- Instructions to a deliverer identifying the recipients to whom the ad contents should be delivered.
- Delivery information intended for the ultimate recipient of the materials, indicating when and by what method(s) the materials are being (or will be) transmitted by the deliverer to the recipient.
- Booking information about the first appearance or insertion of the ad, to assist the recipient in matching the arriving artwork to its booking
- Status reporting for response messages to indicate the state of the materials transaction and the delivered ad content

The business messages supported in the current release of AdsML Materials are:

Message Code	Message Name
AM-DO	Ad Material Delivery Order
AM-DOC	Ad Material Delivery Order Change
AM-DOR	Ad Material Delivery Order Response
AM-DOS	Ad Material Delivery Order Status
AM-DOSE	Ad Material Delivery Order Status Enquiry
AM-DOX	Ad Material Delivery Order Cancellation
AM-M	Ad Material
AM-MR	Ad Material Response
AM-MS	Ad Material Status
AM-MSE	Ad Material Status Enquiry
AM-MX	Ad Material Kill Order
AM-MXR	Ad Material Kill Order Response
AM-RES	Ad Material Resend Request

Note: this list of messages reflects the subset of the AM group messages that have been implemented in AdsMLMaterials 2.5.

The usage of other types of materials-related messages defined in the ACIA will be described when support for those messages is given by the AdsML Materials standard.

For the complete list of AM group messages defined see the *Advertising Components Interactions Analysis* for AdsML Framework 3.0.

3.2 Information contents

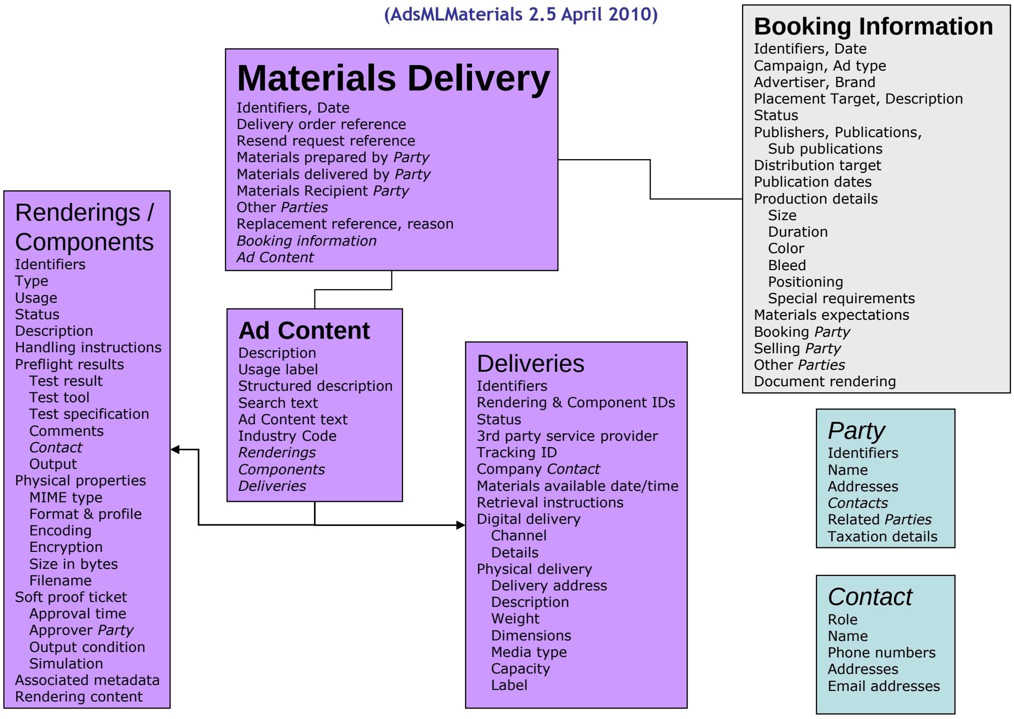
The diagrams below provide an overview of the potential information in a Materials Delivery, Delivery Order or Resend Request message.

These views omit 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.

3.2.1 Materials Delivery

Overview of the potential information in an AdsML Materials Delivery message

(AdsMLMaterials 2.5 April 2010)



Materials Delivery
Identifiers, Date
Delivery order reference
Resend request reference
Materials prepared by *Party*
Materials delivered by *Party*
Materials Recipient *Party*
Other *Parties*
Replacement reference, reason
Booking information
Ad Content

Booking Information
Identifiers, Date
Campaign, Ad type
Advertiser, Brand
Placement Target, Description
Status
Publishers, Publications,
  Sub publications
Distribution target
Publication dates
Production details
  Size
  Duration
  Color
  Bleed
  Positioning
  Special requirements
Materials expectations
Booking *Party*
Selling *Party*
Other *Parties*
Document rendering

Renderings / Components
Identifiers
Type
Usage
Status
Description
Handling instructions
Preflight results
  Test result
  Test tool
  Test specification
  Comments
  Contact
  Output
Physical properties
  MIME type
  Format & profile
  Encoding
  Encryption
  Size in bytes
  Filename
Soft proof ticket
  Approval time
  Approver *Party*
  Output condition
  Simulation
Associated metadata
Rendering content

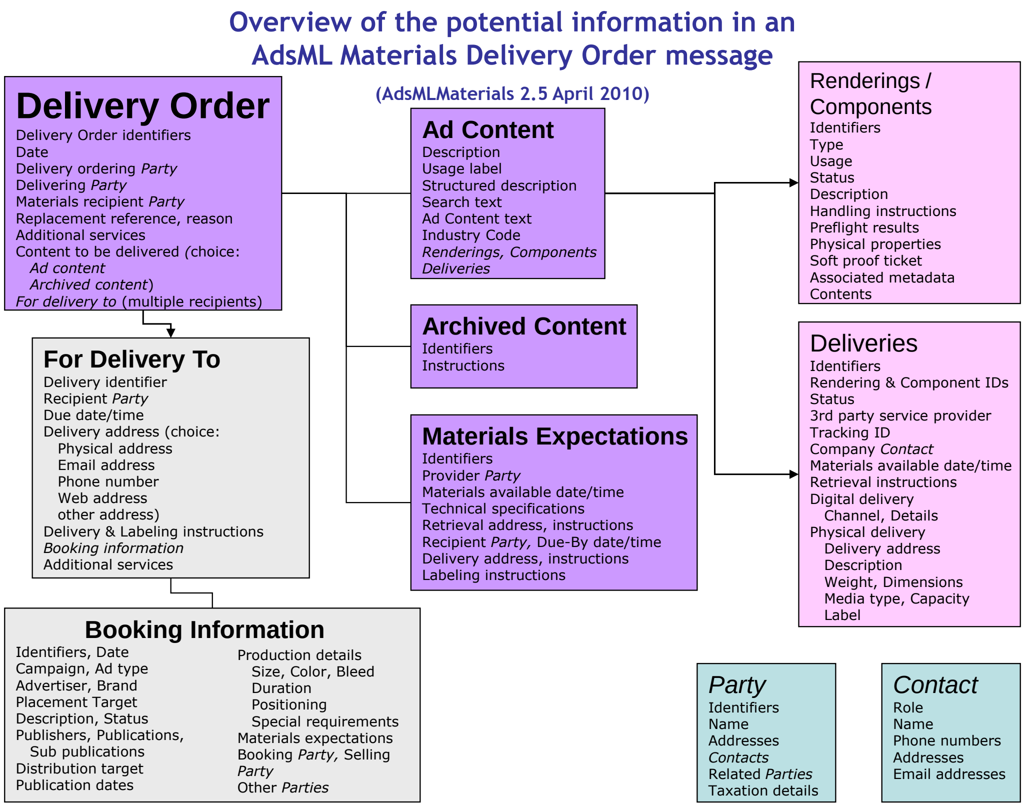
Ad Content
Description
Usage label
Structured description
Search text
Ad Content text
Industry Code
Renderings
Components
Deliveries

Deliveries
Identifiers
Rendering & Component IDs
Status
3rd party service provider
Tracking ID
Company *Contact*
Materials available date/time
Retrieval instructions
Digital delivery
  Channel
  Details
Physical delivery
  Delivery address
  Description
  Weight
  Dimensions
  Media type
  Capacity
  Label

Party
Identifiers
Name
Addresses
Contacts
Related *Parties*
Taxation details

Contact
Role
Name
Phone numbers
Addresses
Email addresses

3.2.2 Delivery Order



Overview of the potential information in an AdsML Materials Delivery Order message

(AdsMLMaterials 2.5 April 2010)

Delivery Order
Delivery Order identifiers
Date
Delivery ordering *Party*
Delivering *Party*
Materials recipient *Party*
Replacement reference, reason
Additional services
Content to be delivered (choice:
Ad content
Archived content)
For delivery to (multiple recipients)

For Delivery To
Delivery identifier
Recipient *Party*
Due date/time
Delivery address (choice:
Physical address
Email address
Phone number
Web address
other address)
Delivery & Labeling instructions
Booking information
Additional services

Booking Information
Identifiers, Date
Campaign, Ad type
Advertiser, Brand
Placement Target
Description, Status
Publishers, Publications,
Sub publications
Distribution target
Publication dates
Production details
Size, Color, Bleed
Duration
Positioning
Special requirements
Materials expectations
Booking *Party*, Selling *Party*
Other *Parties*

Ad Content
Description
Usage label
Structured description
Search text
Ad Content text
Industry Code
Renderings, Components
Deliveries

Archived Content
Identifiers
Instructions

Materials Expectations
Identifiers
Provider *Party*
Materials available date/time
Technical specifications
Retrieval address, instructions
Recipient *Party*, Due-By date/time
Delivery address, instructions
Labeling instructions

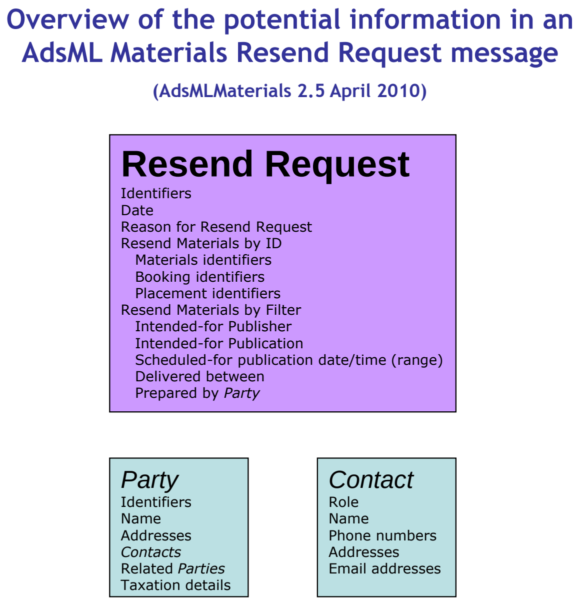
Renderings / Components
Identifiers
Type
Usage
Status
Description
Handling instructions
Preflight results
Physical properties
Soft proof ticket
Associated metadata
Contents

Deliveries
Identifiers
Rendering & Component IDs
Status
3rd party service provider
Tracking ID
Company *Contact*
Materials available date/time
Retrieval instructions
Digital delivery
Channel, Details
Physical delivery
Delivery address
Description
Weight, Dimensions
Media type, Capacity
Label

Party
Identifiers
Name
Addresses
Contacts
Related *Parties*
Taxation details

Contact
Role
Name
Phone numbers
Addresses
Email addresses

3.2.3 Materials Resend Request



Overview of the potential information in an AdsML Materials Resend Request message

(AdsMLMaterials 2.5 April 2010)

Resend Request
Identifiers
Date
Reason for Resend Request
Resend Materials by ID
Materials identifiers
Booking identifiers
Placement identifiers
Resend Materials by Filter
Intended-for Publisher
Intended-for Publication
Scheduled-for publication date/time (range)
Delivered between
Prepared by *Party*

Party
Identifiers
Name
Addresses
Contacts
Related *Parties*
Taxation details

Contact
Role
Name
Phone numbers
Addresses
Email addresses

Copyright © 2010 AdsML Consortium. All rights reserved.

4 Message Choreography

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

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

AdsMLMaterials includes two main categories of messages:

- Business messages, i.e. messages such as materials delivery, status requests, and 'kill orders' that are part of the parties' materials management process.
- Administrative messages, i.e. house-keeping messages for the systems involved in exchange of business messages. Examples are error messages and receipts of received AdsMLMaterials XML files.

4.1 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, from which (in due course) a business response message will be generated.

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 AdsMLMaterials messages. For a description of administrative messages and error handling, please see the *AdsML E-commerce Usage Rules & Guidelines*.

4.2 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 AdsMLMaterials messages. For a description of test messages, please see the *AdsML E-commerce Usage Rules & Guidelines*.

4.3 Response Modes

The preferred messaging model is the Request-Response model as specified in the AdsML Framework.

However, since legacy applications may have limited ability to provide appropriate responses, it is also possible to use a model where only requests are transmitted, assuming an acceptance on the receiver's side. If a problem occurs when a materials message cannot be accepted, it has to be solved manually. This kind of model is called a datagram model.

As a summary:

1) Applications **SHOULD** apply the full Request-Response model

2) If agreed by communication parties, implementations **MAY** use a datagram model (no responses required)

4.4 Business Messages

AdsML Materials supports several short content transmission "transactions" involving the following messages: the delivery of materials to a delivery agency along with instructions to send them to one or more ultimate recipients (AM-DO, AM-DOC, AM-DOX, AM-DOR); the delivery of materials to a publisher or broadcaster (AM-M, AM-MR); reporting on the status of a delivery or delivery order (AM-MS, AM-MSE, AM-DOS, AM-DOSE); a request by a publisher to resend previously received materials (AM-RES); and a materials 'kill' transaction (AM-MX, AM-MXR).

The delivery, delivery order, resend request and 'kill' of materials are the primary transactions, with status reporting a supporting transaction.

Each business message type is identified by a message code that specifies if the message is, for instance, an ad material delivery message, an ad material kill message, or a response to these messages. AdsMLMaterials supports a subset of business messages as defined in *AdsML Advertising Component Interactions Analysis*, namely messages from the ad materials group (AM).

The message type is expressed as a code value for the *messageCode* attribute on business message elements such as AdMaterial. The code values are defined by and used in the AdsML Framework.

The sections below give a summary of the messages in the "Content transmission" subgroup. For more information, see the reference section for each message element (named as the message name in CamelCase) below and refer to the *AdsML Advertising Component Interactions Analysis*.

4.4.1 Delivery Order and Delivery Order Status Messages

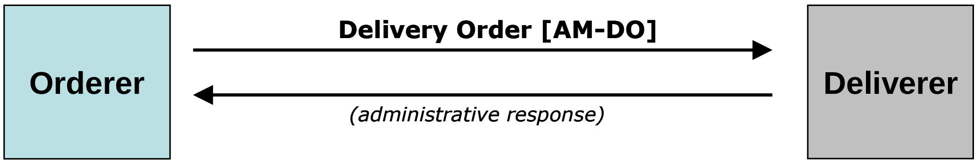
A Delivery Order is initiated by an Advertiser or one of its agents (such as a materials Producer) and is sent to a Deliverer of ad materials, which can confirm or deny acceptance of the Delivery Order. Acceptance of a Delivery Order constitutes a business agreement between Orderer and Deliverer according to the terms of their TPA.

Each Delivery Order (AM-DO) provides, references or describes the content that is to be delivered and specifies the delivery recipients. Metadata about the intended delivery such as due date can be included in the message, but the delivery mechanism to be used by the deliverer is not specified in the delivery order. The deliverer is free to use any appropriate mechanism to satisfy the delivery order. (Any Delivery elements contained inside an AdContent structure in the delivery order only describe how the materials are being (or have been) transmitted to the deliverer, and not how the deliverer should re-send them.)

The Delivery Order Response (AM-DOR) message is issued based on the information that was provided in the AM-DO to which it is a response, and may be sent before the delivery has been completed. Therefore, the AM-DOR message may not represent the final state of the delivery. In this case, status messages can be used to subsequently report on the status of the delivered materials after the initial AM-DOR message has been sent. Status reporting can be either directly requested by the deliverer issuing a status enquiry (AM-DOSE, AM-MSE), or the recipient can spontaneously issue one or more status messages (AM-DOS, AM-MS) in a 'broadcast' style to report on the status of the delivered content in the recipient's systems.

4.4.1.1 Datagram messaging from orderer to deliverer

1) The orderer sends a Delivery Order (AM-DO) message to the deliverer. Once the orderer has received an Administrative Response from the deliverer (indicating that the message was received), the deliverer is assumed to have accepted the order.



Delivery order datagram messaging. This pattern is also used for changes and cancellations.

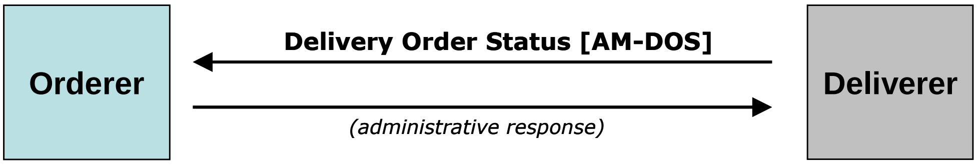
2) To change a previously transmitted order, the orderer sends a Delivery Order Change (AM-DOC) message to the deliverer. Once the orderer has received an Administrative Response from the deliverer (indicating that the message was received), the deliverer is assumed to have accepted the changes to the delivery order.

3) To cancel a previously transmitted delivery order, the orderer sends a Delivery Order Cancel (AM-DOX) message to the deliverer. Once the orderer has received an Administrative Response from the deliverer (indicating that the message was received), the deliverer is assumed to have cancelled the order.

In all cases, when a request includes unacceptable conditions, it is up to the deliverer to contact the orderer and resolve the problem using non-AdsML mechanisms. The parties may then agree to amend the existing order in their respective systems, in which case no further AdsML messages are sent, or to use a Delivery Order Change message to update the terms of the order.

4.4.1.2 Datagram messaging from deliverer to orderer

1) A deliverer **MAY** send a Delivery Order Status (AM-DOS) message spontaneously at any time during the order's lifecycle.



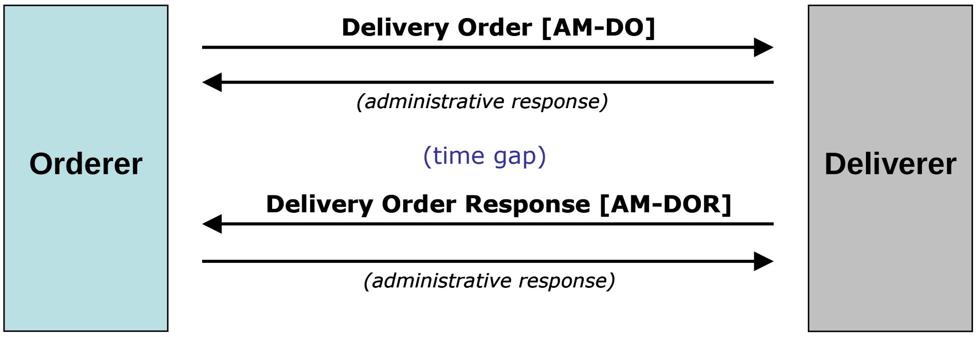
Delivery Order status message

4.4.1.3 Request-Response messaging model

1) A Delivery Order (AM-DO) **MUST** result in a Delivery Order Response (AM-DOR) that either confirms or denies the order.

2) If an order is denied, the reason **MUST** be specified in the response message using the `adsml:RequestDenied` element.

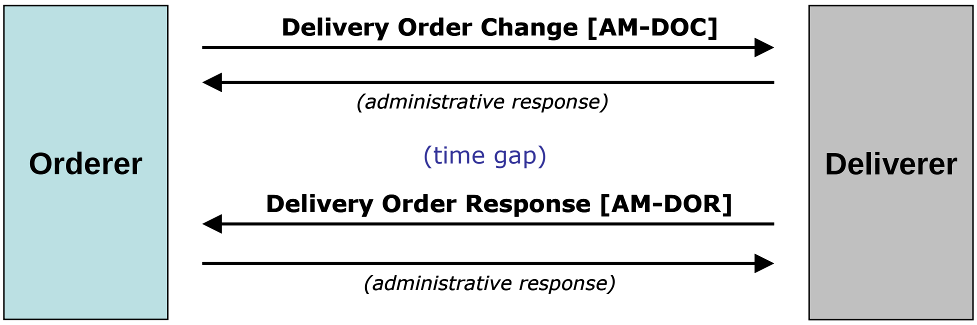
3) If an order is confirmed, the response message **MUST** echo back to the orderer the details of the accepted order as it exists in the deliverer's system (to the level of detail agreed by the parties in their TPA). Partial acceptance of an order is not permitted.


Orderer
Delivery Order [AM-DO]
(administrative response)
(time gap)
Delivery Order Response [AM-DOR]
(administrative response)
Deliverer

Delivery order request-response, including administrative responses.

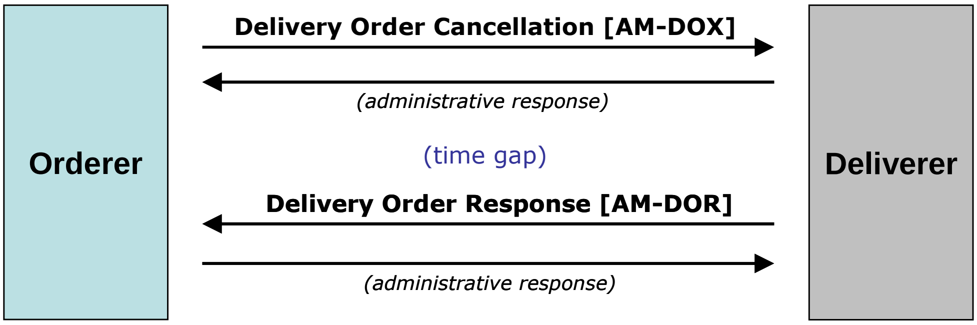
4) A delivery order can be changed by the orderer using the Delivery Order Change (AM-DOC) message.

5) All change requests **MUST** be fully specified delivery orders and must completely replace a previous order. Partial updates are not allowed, and if received by the deliverer **MUST** either trigger a "Request Denied" response message (if the change request is evidently incomplete) or be treated as a complete replacement of the prior order.


Orderer
Delivery Order Change [AM-DOC]
(administrative response)
(time gap)
Delivery Order Response [AM-DOR]
(administrative response)
Deliverer

Delivery order change, followed by the deliverer's response.

6) An orderer can cancel an entire delivery order using the Delivery Order Cancellation (AM-DOX) message. The receiver of the cancellation message **MUST** confirm or reject the cancellation with a Delivery Order Response (AM-DOR). (Note: cancellation of just part of a delivery order, for example removing one delivery from a set of intended deliveries, is done using a Change message rather than a Cancellation message.)

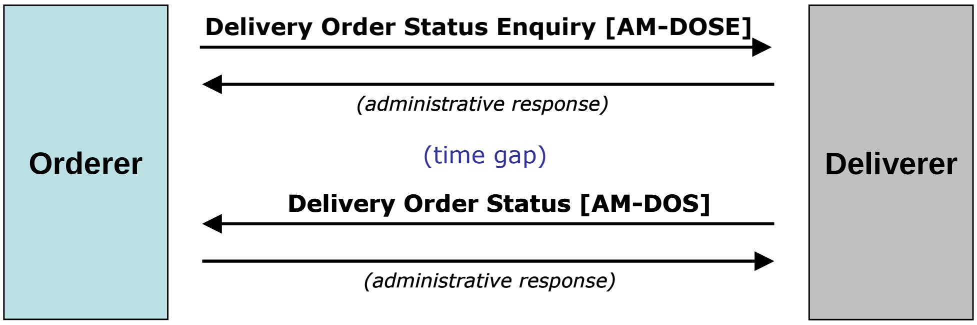

Orderer
Delivery Order Cancellation [AM-DOX]
(administrative response)
(time gap)
Delivery Order Response [AM-DOR]
(administrative response)
Deliverer

Delivery order cancellation, followed by the deliverer's response.

7) A Delivery Order Status Enquiry (AM-DOSE) **MUST** result in a Delivery Order Status (AM-DOS) response.

Copyright © 2010 AdsML Consortium. All rights reserved.

8) If a Delivery Order Status message is a response to a Delivery Order Status Enquiry, it **MUST** reference the Delivery Order Status Enquiry's message identifier and delivery order identifier.


Orderer
Delivery Order Status Enquiry [AM-DOSE]
(administrative response)
(time gap)
Delivery Order Status [AM-DOS]
(administrative response)
Deliverer

Order status enquiry and response.

9) A Delivery Order Status (AM-DOS) **MAY** be issued by the deliverer without a previous Delivery Order Status Enquiry (AM-DOSE) having been received.

4.4.2 Ad Material and Material Status Messages

A materials delivery transaction between a sender and the end user (typically a publisher or broadcaster) of a single set of materials is initiated by the sender of the materials according to the terms of the parties' TPA. A delivery is requested by the sender and confirmed or denied by the recipient. The recipient's acceptance of a delivery is determined by the terms of their TPA.

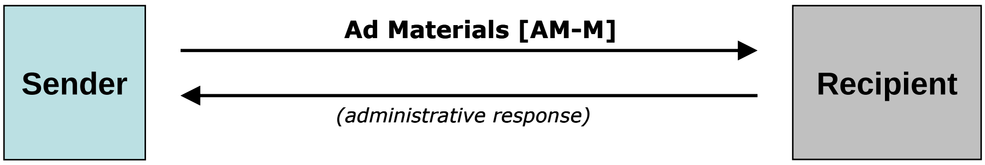
Each AM-M is a notification of a delivery that is in process, completed or will start in the future. The AM-MR message is a response to that notification, and may constitute a rejection of the delivery.

The AM-MR message is issued based on the information that was provided in the AM-M to which it is a response, and may be sent before the contents of that message, i.e. the delivered content, have been processed or the delivery has been completed. Therefore, the AM-MR message may not represent the final state of the delivery. In this case, status messages can be used to subsequently report on the status of the delivered materials after the initial AM-MR message has been sent. Status reporting can be either directly requested by the deliverer issuing a status enquiry (AM-MSE), or the recipient can spontaneously issue one or more status messages (AM-MS) in a 'broadcast' style to report on the status of the delivered content in the recipient's systems.

Note: An AM-M message can also be used to transmit ad materials to a Deliverer of Ad Materials, although it is not optimized for that purpose and does not support complex delivery scenarios. It is recommended that the AM-DO (Delivery Order) message be used when sending materials or instructions to a Deliverer of Ad Materials.

4.4.2.1 Datagram messaging from sender to recipient

1) The sender sends an Ad Material (AM-M) 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 delivery.



Ad materials datagram messaging. This message exchange is used for sending either new or replacement materials.

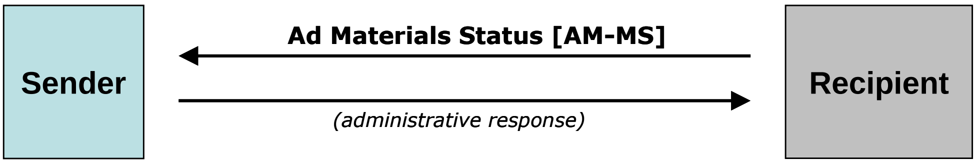
2) To deliver replacement content the sender initiates a new delivery transaction, the transaction explicitly identifying the content-to-be-replaced using the `ReplacesMaterialsReference` structure in the AM-M. Once the sender has received an Administrative Response from the recipient (indicating that the message was received), the recipient is assumed to have accepted the replacement content delivery.

In all cases, if the content is unacceptable to the recipient or its delivery fails, and the recipient is unable to send an Ad Material Status (AM-MS) message, then it is up to the recipient to contact the sender by a non-AdsML mechanism and resolve the problem.

To redeliver ad content, a new materials delivery transaction is always initiated.

4.4.2.2 Datagram messaging from recipient to sender

1) An Ad Material Status (AM-MS) **MAY** be issued by the recipient without a previous Ad Material Status Enquiry (AM-MSE) having been received. This can be used to provide notification that previously delivered content did not pass preflight or other business validation.



Materials status message.

4.4.2.3 Request-Response messaging model

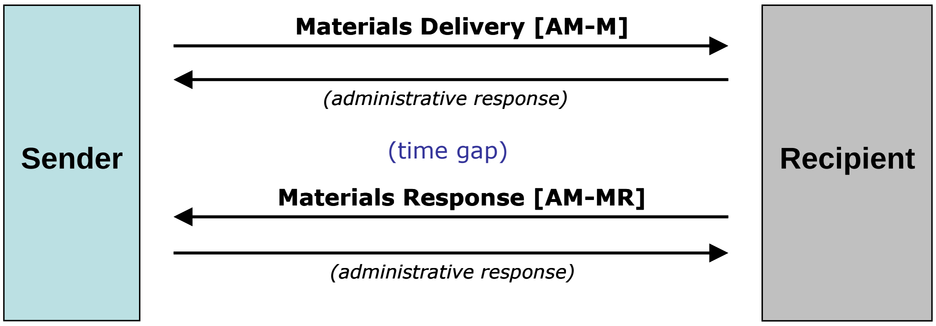
1) An Ad Material delivery (AM-M) message is issued by a Sender of ad material to the Recipient of the ad material in order to notify the recipient of the delivery and provide supporting metadata about the ad materials. An Ad Material delivery message **MUST** result in an Ad Material Response (AM-MR) message that either accepts or denies the delivery. The Ad Material Response message **MUST** explicitly identify the delivery message to which it is a response.

2) If an Ad Material delivery is refused, then the reason **MUST** be specified in the Ad Material Response message using the `adsml:RequestDenied` element.

3) If a delivery is confirmed, the response message **MUST** echo back the status and identification of the transaction and the ad content delivered by it (i.e. the Status, *Identifier and *Reference elements.)

4) A delivery transaction can be used to deliver content that replaces content that was previously delivered. To deliver replacement content the sender initiates a new delivery transaction, the transaction explicitly identifying the content-to-be-replaced using the `ReplacesMaterialsReference` structure in the AM-M. All

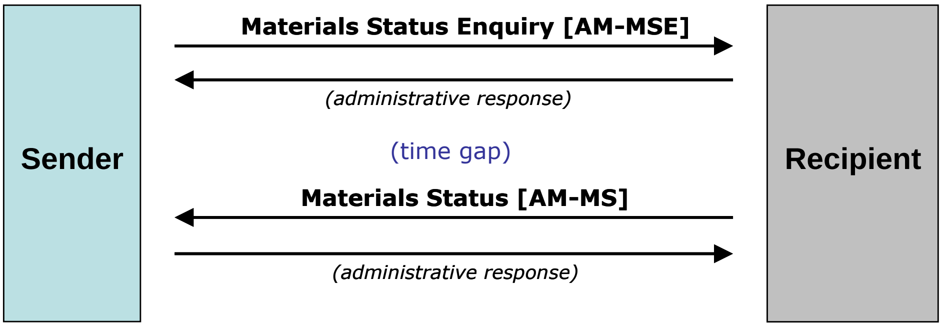
replacement content requests **MUST** be fully specified delivery transactions and must completely replace the previously delivered ad content.


Sender
Materials Delivery [AM-M]
(administrative response)
(time gap)
Materials Response [AM-MR]
(administrative response)
Recipient

Materials delivery and response.

5) An Ad Material Status Enquiry (AM-MSE) request is issued by a Producer or Sender at any time to request the state of ad material in a Recipient's or publisher's systems, and **SHOULD** result in an Ad Material Status (AM-MS) message that provides information about the current status of the materials in the recipient's system.

6) If the Ad Materials Status message is a response to an Ad Material Status Enquiry, it **MUST** reference the status request message to which it is a response.


Sender
Materials Status Enquiry [AM-MSE]
(administrative response)
(time gap)
Materials Status [AM-MS]
(administrative response)
Recipient

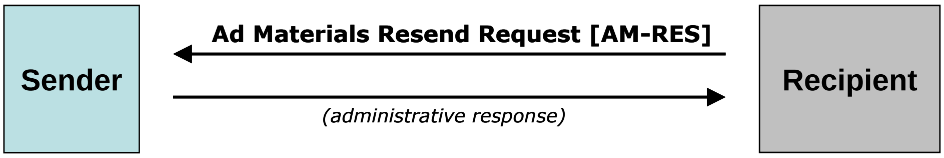
Materials status enquiry and response.

4.4.3 Ad Material Resend Request Messages

A resend request is a request from a recipient to a sender to resend some previously delivered (and received) ad materials, usually because the materials were accidentally damaged or misplaced during the production process. It is sent in datagram mode.

4.4.3.1 Datagram messaging from recipient to sender

1) A recipient **MAY** send an Ad Materials Resend Request (AM-RES) message at any time after having previously received ad materials.


Sender
Ad Materials Resend Request [AM-RES]
(administrative response)
Recipient

Ad Materials Resend Request message

Copyright © 2010 AdsML Consortium. All rights reserved.

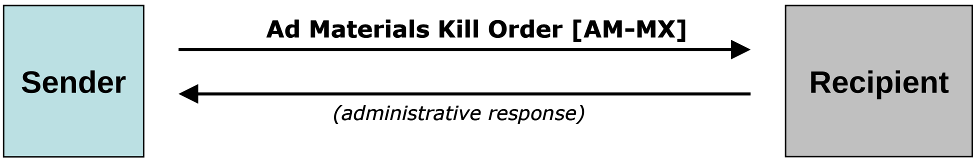
A resend request will usually result in the Deliverer sending a new delivery message (AM-M) containing the requested materials. In this case the new AM-M message **SHOULD** identify the Resend Request message that triggered it.

4.4.4 Ad Material 'Kill Order' Messages

A materials 'kill order' is issued to 'kill' a set of materials whose delivery is in progress or that have been delivered. A materials 'kill order' is considered a separate transaction between a sender and recipient, which is initiated by the sender of the 'kill order' and **MUST** be confirmed or denied by the recipient with a 'kill order' response message.

4.4.4.1 Datagram messaging model from sender to recipient

1) To 'kill' previously delivered ad content, the sender sends an Ad Material Kill Order (AM-MX) request 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 'killed' the delivered ad content.

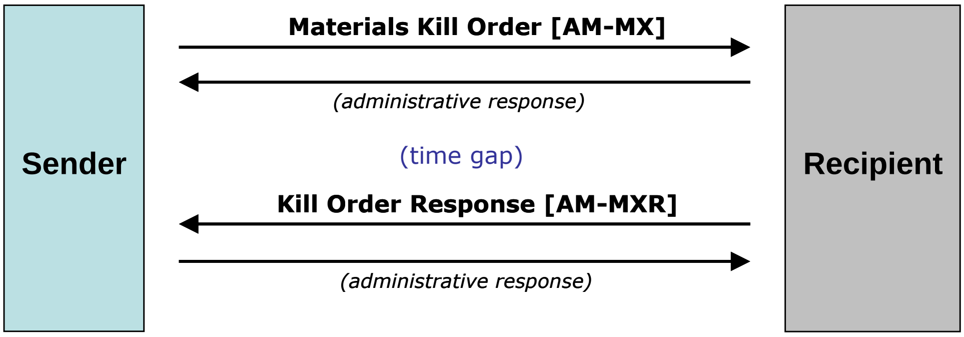


Materials kill order datagram message.

If the recipient is unable to 'kill' the ad content, it is up to the recipient to contact the sender by a non-AdsML mechanism and inform the sender of that fact.

4.4.4.2 Request-Response messaging model

1) To 'kill' previously delivered ad content, the sender sends an Ad Material Kill Order (AM-MX) request to the recipient.
2) The recipient **MUST** respond with an Ad Material Kill Order Response (AM-MXR) which either confirms or denies their intention to carry out the kill order request.


Sender
Materials Kill Order [AM-MX]
(administrative response)
(time gap)
Kill Order Response [AM-MXR]
(administrative response)
Recipient

Materials kill order and response.

4.4.5 Message References – Materials Identifiers

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

Copyright © 2010 AdsML Consortium. All rights reserved.

AdsML Materials supports content transmission transactions for the delivery of materials (AM-M, AM-MR), the ordering of a Materials Delivery (AM-DO*, AM-DOR), reporting the status of delivered materials (AM-MS, AM-MSE) or the status of delivery orders (AM-DOS, AM-DOSE), and the 'killing' of materials already delivered (AM-MX, AM-MXR).

In order to maintain the relationship between these request-response message pairs, and to support the use case where a materials delivery transaction references previously delivered content that it is replacing or a 'kill order' is suppressing delivered materials, stable identification **MUST** be used during the complete suite of possible messages. In the case of a materials delivery order transaction this is the *delivery order identifier*. In the case of a materials delivery transaction it is the *materials identifier*. In the case of the 'kill order' where delivered content is being 'killed', then the 'kill' message references the materials identifier of the materials to-be-killed and also has its own unique transaction identifier, the *materials kill order identifier*. Status messages (AM-MS, AM-MSE) reporting on the status of a materials delivery transaction or a delivery order **MUST** use the same identifier as was used in the initiating materials delivery message (AM-M) or delivery order (AM-DO), i.e. reference the *materials identifier* or the *delivery order identifier*.

An implication of the above is that content management systems **MUST** be able to store the identifier for the materials transaction in its internal data storage.

In general, each Materials Delivery message has the following identifiers:

- *The materials identifier*. An identifier first issued by the party that initiates the materials delivery business process. The materials identifier is the primary identifier for the materials delivery and **MUST NOT** change during the life of the delivery. Its structure **MUST** conform to the AdsML QID format, and it **MUST** be included in any materials delivery message transmitted, both requests and responses. The materials identifier is called `MaterialsIdentifier`. The `MaterialsIdentifier` is used as the primary identifier for materials when referenced from other materials messages. That is, to identify materials being replaced by a new delivery in an ad material message (AM-M), when reporting on the status of a material delivery from an ad material status message (AM-MS, AM-MSE), and when ordering the suppression of materials already delivered using the ad material 'kill order' messages (AM-MX, AM-MXR).
- *The auxiliary materials references*. An optional stack of identifiers that can be used to convey internal identifiers that the buyer, seller, sender and receiver, and other parties use to reference the materials delivery. The auxiliary references are optional; however, once provided any reference identifier value **MUST** remain stable throughout the life of the materials delivery. The auxiliary references include `BuyersReference`, `SellersReference`, `OrderersReference`, `DeliverersReference`, `ReceiversReference`, and `OtherReference`. The `OtherReference` reference is repeatable and may be used to record additional reference identifiers for other parties involved in the workflow. The *auxiliary materials references* are called the `AuxiliaryMaterialsReferences`. The `AuxiliaryMaterialsReferences` are used as secondary reference identifiers for materials when referenced from all other materials messages except the ad material status enquiry where only the `MaterialsIdentifier` is used.
- *The replaces materials reference*. A reference identifier used in an ad materials delivery or delivery order message to identify materials replaced

by the current delivery. The replaces materials reference is optional and is only used in the case of a new delivery replacing materials already delivered. This is designed to facilitate materials management by explicitly identifying the materials to be replaced. The replaces materials reference, if used, **MUST** record the *materials identifier* of the materials being replaced. The *replaces materials reference* is called the `ReplacesMaterialsReference`.

In Delivery Order messages, the materials identifiers listed above appear in the middle of the message (as part of the metadata about the materials that will be delivered pursuant to the delivery order) and are superseded in importance by the following identifiers:

- *The delivery order identifier*. An identifier first issued by the party that initiates the delivery order. The delivery order identifier is the primary identifier for a Delivery Order and **MUST NOT** change during the life of that Order. Its structure **MUST** conform to the AdsML QID format, and it **MUST** be included in any order message transmitted, both responses and requests. The delivery order identifier is called `DeliveryOrderIdentifier`.
- *The orderer's reference*. A reference identifier issued by the ordering party in a delivery ordering transaction. The orderer's reference conveys the internal identifier that the orderer uses to reference this Delivery Order. The orderer's reference is optional; however, once provided, its value **MUST** remain stable throughout the life of the Delivery Order. The orderer's reference is called `OrderersReference` and is contained in `AuxiliaryDeliveryOrderReferences`.
- *The deliverer's reference*. A reference identifier issued by the delivering party in a delivery ordering transaction. The deliverer's reference conveys the internal identifier that the deliverer uses to reference this Delivery Order. The deliverer's reference is optional; however, once provided, its value **MUST** remain stable throughout the life of the Delivery Order. The deliverer's reference is called `DeliverersReference` and is contained in `AuxiliaryDeliveryOrderReferences`.

Response messages need to identify the message they respond to:

- *The "in response to" message ID*. The message ID for the message that the response is about. The ID appears in responses as the attribute *`inResponseToMessageID`*. The *`inResponseToMessageID`* is used in the ad material, ad material status, and ad material kill order response messages.

The Status message also includes:

- *The last received message ID.* A status message **MAY** include the message ID of the last status enquiry received about the particular materials delivery.

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 AdsMLMaterials 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 Administrative Response message to identify the transmission that is being acknowledged, while the message ID 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.

5 Usage of Business Messages

5.1 Delivery Orders

5.1.1 Acceptance

A Deliverer who receives a Delivery Order **MUST** either fully accept or fully reject all of the instructions contained in that Delivery Order. Partial acceptances, "accept with changes", "accept with conditions" and other intermediate forms of response are not permitted.

An acceptance of an order by a deliverer indicates that the deliverer accepts what it believes were the significant terms requested by the orderer, not necessarily the precise text by which those terms were expressed. The amount of latitude available to the deliverer in interpreting the orderer's instructions, and the degree to which the deliverer's response message should echo back the precise terms used by the orderer, **MUST** be agreed in advance between orderer and deliverer.

5.1.2 Changes and cancellations

5.1.2.1 Change a previously-accepted Delivery Order

Requests for changes to a Delivery Order can be specified using the `AdMaterialDeliveryOrderChange` (AM-DOC) business message.

An AM-DOC change message may only be used to deliver changed metadata about a delivery order transaction that is already in process. Change messages **MUST** be used when requesting business-significant changes to the order – for example, addition or removal of an intended recipient. Updates to supplemental or supporting metadata, for example a change in the booking information for an intended recipient, **SHOULD** be communicated via status messages (`AdMaterialDeliveryOrderStatus`) and not via change messages. If the materials themselves have changed, the Producer of Ad Materials should cancel the previous delivery order using an `AdMaterialDeliveryOrderCancellation` (AM-DOX) message, and initiate a new delivery order using another `AdMaterialDeliveryOrder` (AM-DO) message.

The approach to updates is that an order **MUST** be completely specified within the change message, i.e. even though a change only affects details of a single part of the order, the complete delivery order data must be included.

In addition to specifying the delivery order, the `AdMaterialDeliveryOrderChange` business message also allows for a summary of the changes requested in the `adsml:ChangeSpecification` element. The changes can be specified as a list of codes and descriptive text. Also, it is possible for the sender to include a pointer to where each change is made by referencing the primary identifier of the section of the message in which the change occurred. For example, `ChangeLocationReference` elements can be used to reference the `DeliveryIdentifier` in a changed `ForDeliveryTo` element and the `MaterialsIdentifier` in a changed `AdContent` element. If a section of the message has been deleted as part of the change (for example, a `ForDeliveryTo` structure was removed), the `ChangeLocationReference` can contain the identifier of the deleted information.

5.1.2.1.1 Response to a Change request

A Deliverer who receives a Delivery Order Change request **MUST** either accept or reject all of the instructions contained in that Delivery Order Change. Partial acceptances, "accept with changes", "accept with conditions" and other intermediate forms of response are not permitted.

A deliverer **MAY** reject a Delivery Order Change message if it is received after the deliverer has begun to execute any part of the original delivery order. In this case, after the deliverer rejects the change request, the orderer has the option to try to cancel the remainder of the delivery order by sending an `AdMaterialDeliveryOrderCancellation`.

5.1.2.2 Cancel a previously-accepted Delivery Order

Only in the case where a complete delivery order is cancelled should the cancellation message be used. A partial cancellation of, e.g. a single delivery instruction in a multi-delivery order, **MUST** be handled using a change message and **MUST NOT** be handled using a cancellation message.

5.1.2.2.1 Response to a Cancellation request

A Deliverer who receives a Delivery Order Cancellation request **MUST** either accept or reject the complete cancellation. There is no AdsML mechanism for partially accepting a cancellation.

However, if a Delivery Order Cancellation request is received after the deliverer has begun to execute any part of the original delivery order, but while it is possible to cancel *some* of the remaining deliveries, the deliverer **MAY** either reject or accept the cancellation request but either way **MUST** send an `AdMaterialDeliveryOrderResponse` business message. If the deliverer rejects the cancellation request then the deliverer **MUST** send a "Request Denied" response message stating the reason for the request denial. In that case the deliverer will continue to execute the original delivery order until all the deliveries have been accomplished. If the deliverer accepts the cancellation request even though some of the deliveries have been initiated,, the deliverer **MUST** use `ForDeliveryTo/Status` in its Delivery Order Response message to indicate which deliveries were successfully cancelled and which deliveries were made or being made at the time the cancellation request was accepted.

5.1.3 Use of a Delivery Order (AM-DO) to communicate materials delivery expectations

Earlier releases of the *Advertising Components Interactions Analysis* workflows included an Ad Material Delivery Expectations (AM-MDE) message which allowed data derived from a Placement to be sent by the Publisher/Broadcaster or Buyer of advertising to any organization expected to create or deliver the ad materials for that Placement. The purpose of the AM-MDE was to improve the likelihood that those materials would be prepared and labeled properly and then delivered on time and by an acceptable mechanism.

With the implementation of the AM-DO family of messages, it is now possible to use an AM-DO to fulfill the purpose of communicating materials delivery expectations, as an AM-DO is capable of containing Booking information and metadata about the expected materials. Accordingly, the AM-MDE message(s) have been removed from the latest edition of the *Advertising Components Interactions Analysis* and replaced with AM-DO messages.

Note that this usage of the AM-DO, while permitted, overloads the intended usage of the AM-DO in that materials expectations do not represent an "order" to the deliverer, and the deliverer may later receive a second AM-DO message from a different sender that relates to the same delivery. Trading partners wishing to use AM-DO messages for the purpose of conveying delivery expectations **MUST** agree in advance on the business implications of this use of the message.

5.1.4 Generating AM-M messages from a Delivery Order

Except when it is used solely to convey materials delivery expectations, as described in the previous section, a Delivery Order message contains instructions to deliver a single set of ad content to one or more intended recipients.

The Ad Content that is to be delivered is either provided to the deliverer as described in the `AdContent` element, or the deliverer is instructed to re-deliver content that is already in their possession as described in the `ArchivedContent` element. Note with regard to `AdContent` that,

- If the AM-DO instructs the deliverer to re-deliver archived content that is already in its possession, then it will not contain an AdContent element. In this case the deliverer should populate the AM-M AdContent element using the metadata from the AM-DO message by which that ad content was originally provided to the deliverer.
- In the case where an AM-DO specifies delivery to multiple recipients, and the AdContent in it contains artwork with Renderings that are intended for specific recipients (i.e. not all of the renderings in the delivery order are relevant to each recipient), the entire AdContent package should be passed on to the set of recipients and then it is up to each recipient to select and use the Renderings for their situation by examining the Rendering Type and Usage associated with each Rendering. If any Renderings are only intended for specific recipients, then these should be packaged in separate AdContent elements and more than one AM-DO should be used to order their delivery.

Each intended recipient of the ad content that will be delivered is described in an instance of a `ForDeliveryTo` element. Each instance of `ForDeliveryTo` will result in a single AM-M message being sent by the deliverer to the `adsml:MaterialsRecipientParty` specified in that `ForDeliveryTo`.

All of the business information in the resulting AM-M can be populated using information from the AM-DO according to the following mapping table. Except for the `TransmissionTo` element, paths for the elements in the table below begin at the message's business payload element, e.g. `AdMaterial/` in the AM-M message and `AdMaterialDeliveryOrder/` in an AM-DO.

When generating the AM-M, then it should be populated with data from the AM-DO as described in the following tables.

The information in these elements in an AM-DO message, IF present…	**… MUST appear in this corresponding location in each resulting AM-M message**
`AdContent/ MaterialsIdentifier`	`MaterialsIdentifier`
`ArchivedContent/ MaterialsReference`	`MaterialsIdentifier`
`AdContent/ AuxiliaryMaterialsReferences`	`AuxiliaryMaterialsReferences`

ArchivedContent/ AuxiliaryMaterialsReferences	AuxiliaryMaterialsReferences
DeliveryOrderIdentifier	DeliveryOrderReferences/ DeliveryOrderReference
AuxiliaryDeliveryOrderReferences	DeliveryOrderReferences/ AuxiliaryDeliveryOrderReferences
ReplacesMaterialReference	ReplacesMaterialReference
ForDeliveryTo/ BookingInformation	BookingInformation
adsml:MaterialsPreparerParty	adsml:MaterialsPreparerParty
adsml:DeliveringParty	adsml:DeliveringParty
ForDeliveryTo/ adsml:MaterialsRecipientParty	adsml:MaterialsRecipientParty
AdContent (all elements beginning with adsml:DescriptionLine)	AdContent

The information in these elements in an AM-DO message, IF present…	**… SHOULD appear in this corresponding location in each resulting AM-M message**
MaterialsExpectations (all elements)	BookingInformation/PlacementInformation/MaterialsExpectations

5.2 Resend Requests

A resend request (AM-RES) is a request from a recipient of ad materials (typically a publisher) to a deliverer to resend some previously delivered (and received) materials, usually because the materials were accidentally damaged or misplaced during the production process. The request can be for anything from a single instance of ad materials (e.g. "please resend materials number 12345") to a potentially large set of previous deliveries (e.g. "please resend everything you sent me last week").

The resend request structure provides a choice between two approaches for identifying the materials that should be resent:

- by specific identification, for use when the relevant identifiers are known
- by a filter which, when executed, defines the set of materials to be resent.

Only one of these two approaches can be used in any given resend request message. The sender of the message should use whichever approach most efficiently identifies the required materials.

5.2.1 Requesting materials by specific identification

When requesting the resend of materials for which the Materials identifiers, Bookings identifiers or Placement identifiers are known, the requester **SHOULD** specify the materials to be resent by populating the ResendMaterialsByReference element. This element allows for the transmission of one or more Materials IDs (and/or references), Bookings IDs (and/or references) and Placement IDs (and/or references). The structure allows the requester to identify exactly which ad materials should be resent, either by identifying those materials directly, or by identifying the bookings or placements with which they were associated when they were previously delivered.

A populated ResendMaterialsByReference element consists of a request to resend all previously sent materials which match any of the provided identifiers. For example, if both a materials identifier and a bookings identifier are included in ResendMaterialsByReference, then the recipient of the resend request **MUST** attempt to resend both any ad materials having the specified materials identifiers and any previously delivered ad materials that were associated with the specified booking(s).

Because the sender of a Resend Request may not know which types of information about previously sent materials are readily available to the recipient of the message, the sender **SHOULD** populate as many of the elements in ResendMaterialsByReference as it can. The Materials identifiers of the previously sent materials **MUST** be populated if known, because they provide the most direct mechanism by which the deliverer can identify the desired materials. In this case the identifiers of the Placements and Bookings with which the previously-sent materials were associated **SHOULD** also be populated if possible, because they may assist the deliverer in retrieving the correct materials.

ResendMaterialsByReference can also be used to request the re-delivery of materials associated with specific Placements or Bookings even though their materials identifiers are not known, by populating only the relevant Placement and Booking identifiers.

If none of these identifiers are known, then the sender of the resend request message should use the alternative approach of requesting materials by specifying a filter. See below.

5.2.2 Requesting materials by providing a filter

The ResendMaterialsByFilter element allows the sender of the resend request message to provide query parameters which, when executed by the recipient of the message, will define the set of ad materials to be resent.

The requester may specify any combination of 0…1 instance of each of the following:

- adsml-bo:PublishedBy – the Party which will publish the materials
- ForPublication – the name of the publication or broadcast in which the materials will appear
- ScheduledFor – the date/time range within which the materials are scheduled to be published or broadcast
- DeliveredBetween – a date/time range within which the original materials were received by the sender of this message
- adsml:MaterialsPreparerParty – the party which prepared the materials

A populated ResendMaterialsByFilter element consists of a request to resend all previously sent materials which match all of the provided parameters. For example, if both a publisher (adsml-bo:PublishedBy) and publication

Copyright © 2010 AdsML Consortium. All rights reserved.

date (ScheduledFor) are included in a resend request, then the recipient of the resend request **MUST** only resend any previously-sent ad materials that were intended to be published by the specified publisher and were scheduled to be published on the specified publication date. If no previously-sent materials match all of the selection criteria, then there is nothing to resend.

The sender of a Resend Request **SHOULD** populate as few of the elements in ResendMaterialsByFilter as are necessary in order to identify the desired materials.

In some cases it may not be possible in a single filter expression to identify all of the materials that need to be resent. For example, only one Publisher can be identified in ResendMaterialsByFilter. Therefore, in order to request the resend of materials intended for more than one publisher, the requester must send multiple Resend Request messages, each containing a ResendMaterialsByFilter that specifies one of the publishers.

5.3 The level of content in a response message

5.3.1 Responses to a Materials Delivery (AM-M)

The AdMaterialResponse and AdMaterialStatus response messages are able to 'mirror' back the entire content of the AdMaterial delivery message to which they are responding. When deciding how much content to mirror back in an AdMaterialResponse or AdMaterialStatus response message, best practice is to,

- Always return any identification elements given in the original request message and populate all optional identification elements in the response message – namely,
  - For AdMaterialResponse the inResponseToMessageID, inResponseToMessageCode, MaterialsIdentifier and AuxiliaryMaterialsReferences
  - For AdMaterialStatus, all of the above plus lastReceivedMessageID
- Always return the status of the materials delivery – namely,
  - For AdMaterialResponse ,
    - If the delivery request is rejected the RequestDenied element
    - If the delivery request is not rejected the Status element child of the AdMaterialResponse element at a minimum. If required to report on the status of each individual Rendering or Delivery, then the AdContent element should be contained with a minimal data set consisting of Rendering and Delivery elements as required. Note that if status reporting is required down to this level that the status reporting **MUST** be complete and so all Rendering and Delivery children **MUST** be contained. Within Rendering, the identification and status elements **MUST** be contained. Within Delivery, the identification and status data **MUST** be contained.

  - For AdMaterialStatus when operating in request-response mode the inResponseToMessageID, inResponseToMessageCode and lastReceivedMessageID

5.3.2 Responses to a Materials Delivery Kill Order (AM-MX)

When responding to an AdMaterialKillOrder message, then the optional AuxiliaryMaterialsReference identifiers **MAY** be returned but are not required to be so. This is because the primary identifiers of a 'kill order' transaction are the AdsML QID format identifiers used for the message ID, materials identifier, and kill order identifier contexts.

5.3.3 Responses to a Delivery Order (AM-DO*)

The AdMaterialDeliveryOrderResponse and AdMaterialDeliveryOrderStatus response messages are able to 'mirror' back the entire content of the AdMaterialDeliveryOrder message to which they are responding. When deciding how much content to mirror back in an AdMaterialDeliveryOrderResponse or AdMaterialDeliveryOrderStatus response message best practice is to,

- Always return any identification elements given in the original request message and to populate all optional identification elements in the response message – namely,
  - For AdMaterialDeliveryOrderResponse the inResponseToMessageID, inResponseToMessageCode, DeliveryOrderIdentifier and AuxiliaryDeliveryOrderReferences
  - For AdMaterialDeliveryOrderStatus, all of the above plus lastReceivedMessageID
- Always return the status of the delivery order – namely,
  - For AdMaterialDeliveryOrderResponse,
    - If the delivery order request is rejected, the RequestDenied element
    - If the delivery request is not rejected, the Status element child of the AdMaterialDeliveryOrderResponse element at a minimum.
      - If required to report on the status of each individual delivery, then the ForDeliveryTo element describing each delivery should be contained with a minimal data set consisting of the mandatory DeliveryIdentifier and MaterialsRecipientParty elements plus a Status code as required.
      - If required to report on the status of the ad content which was conveyed or referenced by the delivery order, then the AdContent or ArchivedContent element should be contained (as appropriate) with a minimal data set consisting of MaterialsIdentifier or

MaterialsReference and Status elements as required.

- If required to report on the status of an individual Rendering or Delivery, then the AdContent element should be contained with a minimal data set consisting of Rendering and Delivery elements as required. Note that if status reporting is required down to this level that the status reporting **MUST** be complete and so all Rendering and Delivery children **MUST** be contained. Within Rendering, the identification and status elements **MUST** be contained. Within Delivery, the identification and status data **MUST** be contained.

5.4 The role of booking information in an AdsML Materials message

Booking information extracted from an AdsMLBookings Ad Order can be embedded in the AdMaterial (AM-M*) and AdMaterialDeliveryOrder (AM-DO*) messages defined by AdsML Materials.

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

Note that AdsMLMaterials makes a specific subset of booking data available in the Materials messages, this subset being only that booking information considered relevant to a materials workflow in order to support materials-booking reconciliation processes.

5.5 How to record and report the Status of a materials delivery in AdsML Materials delivery (AM-M*) messages

5.5.1 Status reporting – technical and business levels

Status is recorded at two levels in AdsML Materials delivery response messages, at the message level and at the ad content level. It is important to note the difference between the status recorded at the message level and at the ad content level because they reflect business-level and technical-level focus. Business-level status reflects the state of the materials transaction and so has business significance; technical-level status reflects the result of automated processing of the delivered ad content, the results of which determine whether the transaction is successfully completed, pending, or has in fact failed.

At the message level, the status is a business-level response that indicates the state of the materials transaction, namely if the transaction has been completed, is pending, or has failed. Status is recorded for the AdMaterialResponse, AdMaterialStatus, and AdMaterialKillOrderResponse response messages.

At the ad content level, the status is a technical-level response that reports the state of the delivered ad content (the rendering) in the recipient's systems and

Copyright © 2010 AdsML Consortium. All rights reserved.

reports on the state of the delivery process. Status is recorded at the Rendering and Delivery structural levels. At this level, the status indicates whether the delivered ad content, the rendering, is suitable for use in the booking it was intended for and indicates the state of the delivery of the rendering. Rendering status values would reflect the result of validations that require some processing of the received ad content, e.g. a lookup of booking information or technical specifications and would result in responses such as material suitable or fails to conform to publisher's technical specifications. Delivery status indicates if the delivery has been completed, is pending, in progress or has failed. Note that at the ad content level status is recorded distinctly for the rendering and for its delivery(ies) to allow for the use case of a rendering having more than one delivery associated with it.

Note also that in the use case of a future delivery it is not required to use the Delivery/adsml:Status with a value of 'pending' as that data is implicitly given by the Delivery/MaterialsAvailableDateTime.

When used the adsml:Status element has the following semantics,

- AdMaterialStatus/adsml:Status. Records the current state of the delivered ad materials in the responder's systems. Indicates if all Rendering(s) and Delivery(s) comprising the AdContent have been received and processed and so records the status of the delivery of the ad content and the delivered rendering(s) as a whole.
- AdMaterialResponse/adsml:Status. Records the current state of the ad materials transaction. Note that an AdMaterialResponse may be sent before the delivered materials have been processed and so may provide only a business–level acknowledgement of receipt of the materials delivery with subsequent reporting on the processed materials given by an AdMaterialStatus message.
- AdMaterialKillOrderResponse/adsml:Status. Records the current state of a 'kill order' transaction. Indicates if the delivered ad materials (all of the Rendering(s) comprising the AdContent delivered by the referenced materials transaction) in the responder's systems have yet been 'killed'.
- Delivery/adsml:Status. Records the status of the delivery. For example, the delivery has failed (e.g. due to damaged digital media or a corrupt file), is complete (i.e. has been 'Accepted'), or is pending.
- Rendering/adsml:Status. Records the status of the rendering. For example, the Rendering is 'Accepted' (i.e. conforms to publisher's technical specifications) or a rendering file is corrupt (i.e. damaged in transmission and so requires resending in a new delivery transaction).

The Status element value is recorded in codified form using the CodeType structure defined in the AdsML Type Library, and can use a controlled vocabulary for validation. The status value can be qualified by optional Status/StatusQualifier elements (also of CodeType) if required to record status at 2 tiers.

Note: when the ad content structure is embedded in an AdsMLBookings message the 'business-level' status of the ad materials is recorded by a specific /adsml-ma:AdContent/adsml:Status element provided for this purpose. This status element is only present in the ad content structure defined for use in Ad Material Delivery Orders or in other AdsML standards where the materials status would not logically be reported at the message level as with AdsMLMaterials but at the ad content level.

5.5.2 Status reporting for materials deliveries – selection and use of values from the AdsML Status & Status Qualifier Code CVs for status reporting

As stated above, status reporting in an AdsML Materials Delivery (AM-M*) message is at the business and technical level. To support this, a simple status model is defined where the status of the materials delivery request is reported at 3 lifecycle stages based on the processing state of the delivered materials: (1) acknowledgement of receipt of information, (2) processing is in progress, (3) outcome of processing.

The lifetime of a materials delivery request goes through these states in the following order as the recipient reports on the state of the delivered materials:

1. Initial acceptance or rejection – the materials delivery request message is accepted or rejected ('Request Denied').
2. '`Pending`' - pending execution; processing of the materials has not yet begun.
3. '`BeingProcessed`' - currently executing; processing of the materials or their delivery is underway.
4. Processing is complete and the outcome is either:
    a. At the business level
        - '`Succeeded`' - message processing has completed with a successful outcome.
        - '`Failed`' - technical processing has failed to complete due to technical errors.

    b. At the technical level
        - '`Accepted`' - processing has completed successfully and the content / delivery is accepted.
        - '`Rejected`' - processing has completed with errors and the content / delivery is rejected.
        - '`Failed`' - technical processing has failed to complete due to technical errors.

In the table below, the use of `AdsMLStatusCodeCV` and `AdsMLStatusCodeQualifierCV` values for reporting status at these levels is shown. Note that:

- 'Possible reasons' in the descriptions below are example `StatusQualifier` values
- No 'partial success' delivery scenarios. There are assumed to be no partial success outcomes and that a materials delivery is either successful or unsuccessful in its entirety. For example, consider the scenario of a delivery with 2 delivery methods and 2 renderings:
    a. In the case that in the response to the delivery one of their status values is '`Failed`', then the whole message will have a '`Failed`' status.
    b. In the case that one is '`Rejected`' and one is '`Accepted`' with no '`Failed`', then the whole message will have '`Failed`' status.

5.6 How to record and report the Status of a materials delivery in AdsML Materials delivery order (AM-DO*) messages

5.6.1 Status reporting – technical and business levels

Status is recorded at either two or three levels in AdsML Materials delivery order response messages: at the message level, at the ForDeliveryTo level, and optionally at the AdContent level. It is important to note the difference between the status recorded at the message level, the ForDeliveryTo level and the AdContent level because they reflect business-level vs. technical-level focus. Business-level status reflects the state of the materials delivery order transaction and so has business significance; technical-level status reflects the result of automated processing of (1) the ad content received by the deliverer and to be delivered to the intended recipients ('for delivery to') parties, and (2) the status of the deliveries to those intended recipients. The results of this technical level processing determine whether the delivery order transaction is successfully completed, pending, or has in fact failed.

At the message level, the status is a business-level response that indicates the state of the delivery order transaction, namely if the transaction has been accepted, rejected, completed, is pending, or has failed. Status is recorded for the AdMaterialDeliveryOrderResponse and AdMaterialDeliveryOrderStatus response messages, in the AdMaterialDeliveryOrderResponse/adsml:Status and AdMaterialDeliveryOrderStatus/adsml:Status elements.

At the ad content level, the status is a technical-level response. It reports on the state of the delivered ad content (the rendering) in the deliverer's systems and reports on the state of the delivery process to the deliverer. Status is recorded at the Rendering and Delivery structural levels. ,

Note that at the ad content level status is recorded distinctly for the rendering and for its delivery(ies) to allow for the use case of a rendering having more than one delivery associated with it.

Note also that in the use case of a future delivery it is not required to use the Delivery/adsml:Status with a value of 'pending' as that data is implicitly given by the Delivery/MaterialsAvailableDateTime.

Status at the 'for delivery to' level reports on the state of the ordered deliveries that the deliverer is making to the intended recipients, the 'for delivery to' parties. This status indicates the state of the ordered materials delivery, that is if the delivery is pending, accepted, rejected, or, in the case of a cancellation, canceled.

When used the adsml:Status element has the following semantics,

- AdMaterialDeliveryOrderStatus/adsml:Status. Records the current state of the entire delivery order transaction in the deliverer's systems. It indicates if all Rendering(s) and Delivery(s) comprising the AdContent have been successfully received by the deliverer for sending on, and the status of the ordered deliveries.
- AdMaterialDeliveryOrderResponse/adsml:Status. Records the current state of the delivery order transaction. Note that an AdMaterialDeliveryOrderResponse may be sent before the

delivered materials have been processed and so may provide only a business–level acknowledgement of receipt of the delivery order with subsequent reporting on the ordered deliveries given by an `AdMaterialDeliveryOrderStatus` message.

- `ForDeliveryTo/adsml:Status`. Records the current state of an ordered delivery, for example that the delivery is "Pending" or "Succeeded", or perhaps was "Rejected" by either the deliverer or the end recipient.delivery.
- `AdContent/adsml:Status`. Records the status of the ad content delivered to the deliverer by the delivery orderer. For example, the AdContent is accepted (i.e. has been 'Accepted').
- `AdContent/Delivery/adsml:Status`. Records the status of the delivery to the deliverer of the materials referenced by this delivery order. For example, the delivery has failed, succeeded, or is pending.
- `AdContent/Rendering/adsml:Status`. Records the status of the rendering in the deliverer's systems. For example, the `Rendering` is 'Accepted' and so the deliverer can use it for sending on, or a rendering file is corrupt (i.e. damaged in transmission and so requires resending in a new delivery transaction).

The `Status` element value is recorded in codified form using the `CodeType` structure defined in the AdsML Type Library, and it can use a controlled vocabulary for validation. The status value can be qualified by optional `Status/StatusQualifier` elements (also of `CodeType`) if required to record status at these tiers.

5.6.2 Status reporting for materials delivery orders – selection and use of values from the AdsML Status & Status Qualifier Code CVs for status reporting

Status reporting in an AdsML Materials Delivery Order (AM-DO*) message is essentially the same as for AdsML Materials Delivery messages, with the status of the delivery order reported at 3 lifecycle stages: (1) acknowledgement of receipt of the order information, (2) processing is in progress, (3) outcome of processing.

The basic AdsML Materials Delivery status reporting model is extended to provide support for cancellation functionality of the Delivery Order.

With regard to cancellation of a delivery order note that,

- No 'partial acceptance' delivery scenarios. A delivery order (or change or cancellation) is either completely accepted or completely rejected.
- No 'partial success' delivery scenarios. There are assumed to be no partial success outcomes and that a materials delivery order is either successful or unsuccessful in its entirety. For example, the scenario of a delivery order with two 'for delivery to' recipients:
    a. In the case that both of their status values is '`Accepted`' then the whole message will have 'Succeeded' and so has an '`Accepted`' status.
    b. In the case that one is '`Failed`' and one is '`Accepted`', then the whole message will have '`Failed`' status.

c. In the case that one is 'Canceled' and one is 'Accepted' with no 'Failed', then the whole message will have 'Failed' status because the deliverer's request to cancel the delivery order was not 100% successful.

5.7 Status reporting using the AdsML Status Code and Status Qualifer CV values

5.7.1 Which of the AdsMLStatusCodeCV values to use for recording status at Status element level

Status CV value	**Description**
Pending	Pending execution; processing has not yet begun.
BeingProcessed	Currently executing; processing is underway.
Accepted	Processing has completed with a successful outcome that the recipient considers acceptable.
Rejected	Processing has completed with an unsuccessful outcome and the recipient rejects the outcome as unacceptable.
Failed	Processing has failed to complete due to technical errors. An unsuccessful outcome - the recipient is unable to complete processing.
Succeeded	Processing of the whole message has been successfully completed. Only for high level statuses.
Canceled	Processing has been canceled with a successful outcome that the deliverer considers acceptable.

Status	**Meaning of Status value in message context of:** • **AdMaterialResponse/Status** • **AdMaterialKillOrderResponse/Status** • **AdMaterialStatus/Status** • **/AdContent/Status** (Note: When reporting status at the ad content level in an AM-* message or when the ad content is used in another AdsML standard) • **AdMaterialDeliveryOrderResponse/Status** • **AdMaterialDeliveryOrderStatus/Status**
Pending	Acknowledges that the message information has been received. The delivery of the content or the processing of the content files / 'kill order' / delivery order has not yet begun (is pending).
BeingProcessed	Processing of at least one of the rendering(s) or delivery (ies) or the 'kill order' is in progress.
Failed	Processing of at least one delivery or rendering or the 'kill order' failed or has been rejected. The whole content /

	message is marked as failed. There are no partial success scenarios.
Succeeded	Processing of all rendering(s) and delivery(ies) or the 'kill order' or the delivery order has completed successfully. The whole content / message is marked as succeeded.

Status	**Meaning of Status value in message context of:** • **/AdContent/Rendering/Status**
Pending	File processing has not yet begun (pending).
BeingProcessed	File processing is in progress.
Accepted	File processing finished successfully and the content was accepted for production.
Rejected	File processing finished and the file has been rejected. Possible reasons include, for example: preflight errors, mismatch to TPA information, booking information conflict. Reasons can be recorded as `StatusQualifier` values.
Failed	Processing of the file has failed. Possible reasons include, for example: media damaged, file corrupted, decoding/ decryption failed. Reasons can be recorded as `StatusQualifier` values.

Status	**Meaning of Status value in message context of:** • **/AdContent/Delivery/Status**
Pending	Delivery has not yet begun (pending).
BeingProcessed	Delivery is in progress.
Accepted	Delivery was completed successfully and the content was accepted by the receiver.
Rejected	The delivery has been rejected. Possible reasons include, for example: delivery method was not in accordance with the TPA. Reasons can be recorded as `StatusQualifier` values.
Failed	Processing of the delivery failed; the files did not reach the receiver. Possible reasons include, for example: transfer error (e.g. protocol error, connection reset), access error (e.g. access denied, the file doesn't exist). Reasons can be recorded as `StatusQualifier` values.

Status	**Meaning of Status value in message context of:** • **AdMaterialDeliveryOrderResponse/ ForDeliveryTo/Status** • **AdMaterialDeliveryOrderStatus/ ForDeliveryTo/Status**

Pending	Delivery has not yet begun (pending).
BeingProcessed	Delivery is in progress.
Accepted	Delivery was completed successfully and the content was accepted by the receiver.
Rejected	The delivery has been rejected. Possible reasons include, for example: delivery method was not in accordance with the TPA. Reasons can be recorded as `StatusQualifier` values.
Failed	Processing of the delivery failed; the files did not reach the receiver. Possible reasons include, for example: transfer error (e.g. protocol error, connection reset), access error (e.g. access denied, the file doesn't exist). Reasons can be recorded as `StatusQualifier` values.
Cancelled	Processing has been cancelled with a successful outcome that the recipient considers acceptable.

5.7.1.1 Which of the AdsMLStatusCodeQualifierCV values to use for recording status at StatusQualifier element level

Status CV value	**Description**
PreflightError	Preflighting has revealed a mismatch between the content file and the technical specifications the file should conform to.
TPAMismatch	Processing has revealed a mismatch between the content file and/or the delivery specifications the materials delivery should conform to as specified in the Trading Partner Agreement (TPA).
BookInfoConflict	The content information does not mirror the information in the booking information associated with that content.
MediaDamaged	The media on which the content file is stored is damaged and so cannot be accepted.
FileCorrupted	The content file is corrupted and so cannot be processed.
FileExtractionError	The content file could not be extracted. For example, due to a decoding or decryption problem.
DeliveryError	A delivery problem has prevented a materials delivery from completing successfully.
RenderingError	A problem with a content file has prevented a materials delivery from successfully completing processing.
PartialDelivery	One or more of the specified delivery(ies) failed to complete successfully.
PartialRenderings	One or more of the rendering(s) comprising the delivery

Copyright © 2010 AdsML Consortium. All rights reserved.

	failed to be successfully processed.
AccessError	Unable to access the content file by the specified delivery channel; unable to initiate transfer. Reasons include factors such as: incorrect authentication information (username and password), broken link (file doesn't exist), etc.
TransferError	Unable to complete the content file delivery due to errors encountered while delivery was in progress. For example, protocol dependent errors (HTTP 500), connection reset, insufficient space on disk to complete delivery.

5.7.1.2 Overview of how to use the AdsML Status & StatusQualifier Code CV values to record state at Status and StatusQualifier level

Status	**Message context of:**		
	AdMaterialResponse/ Status **AdMaterialStatus/Status** **AdMaterialKillOrderResponse /Status** **AdMaterialDeliveryOrderResponse/Status** **AdMaterialDeliveryOrderStatus/Status** **/ForDeliveryTo/Status** **/AdContent/Status**	**/Rendering/ Status**	**/Delivery/Status**
Pending	X	X	X
BeingProcessed	X	X	X
Accepted		X	X
Rejected		X	X
Failed	X	X	X
Succeeded	X		
Cancelled	X (Note: used only in ForDeliveryTo/Status)		

StatusQualifier	**Message context of:**

	AdMaterialResponse/Status/StatusQualifier **AdMaterialStatus/Status/StatusQualifier** **AdMaterialKillOrderResponse /Status/StatusQualifier** **AdMaterialDeliveryOrderResponse /Status/StatusQualifier** **AdMaterialDeliveryOrderStatus/Status/StatusQualifier** **/ForDeliveryTo/Status/StatusQualifier** **AdContent/Status/StatusQualifier**	**Rendering** **/Status/StatusQualifier**	**Delivery** **/Status/StatusQualifier**
PreflightError		X	
TPAMismatch	X	X	X
BookInfoConflict	X	X	
MediaDamaged			X
FileCorrupted		X	
FileExtractionError		X	
DeliveryError	X		
RenderingError	X		
PartialDelivery	X		
PartialRenderings	X		
AccessError			X
TransferError			X

When using qualifier values note that,

- ‘PreflightError’, ‘TPAMismatch’ and ‘BookInfoConflict’ are used as qualifiers in a status context of ‘Rejected’.
- The remaining values are used as qualifiers in a status context of ‘Failed’.

Note that as with all contexts where a code type is specified, the Status element can be restricted to a list of values defined by a controlled vocabulary. Although a user-defined controlled vocabulary can be used, it is **RECOMMENDED** that the AdsMLStatusCodeCV is used for recording status values in an AdsML Materials message.

Note that the status structure allows for the status to be qualified and for both Status and for a description to be added to either Status or StatusQualifier elements to provide human-readable information if desired.

The example below shows a status value of 'Failed' qualified as an 'AccessError', with the details of the access error described as a free-text string. Note in this example the use of xsi:type to specify the controlled vocabulary allows XML Schema validation of the controlled vocabulary values.

```
<adsml:Status>
 <adsml:CodeValue xsi:type="adsml-cv:AdsMLStatusCodeCV">Failed</
adsml:CodeValue>
 <adsml:StatusQualifier>
  <adsml:CodeValue xsi:type="adsml-cv:AdsMLStatusQualifierMaterialsCV"
>AccessError</adsml:CodeValue>
  <adsml:Description>530 Login incorrect.</adsml:Description>
 </adsml:StatusQualifier>
</adsml:Status>
```

5.8 How to report the Status of a soft-proofing test

The status of a soft-proofing test for a Rendering file can be reported by using the AdContent/Rendering/SoftProofTicket/ApprovalStatus structure using status values from the AdsMLStatusCodeCV.

Only the values of 'Accepted' or 'Rejected' are used in the soft proof context to report if the rendering file has been successfully soft-proofed or if it has failed the soft-proofing test.

Note that the ApprovalStatus structure is specified as an adsml:CodeType only, and so it is not possible to record a status qualifier value.

5.8.1 Which AdsMLStatusCodeCV values to use for recording status at Status element level

Status	**Meaning of Status value in message context of:** • **/AdContent/Rendering/SoftProofTicket/ApprovalStatus**
Pending	N/A
BeingProcessed	N/A
Accepted	The file processing finished successfully and the soft-proofing test was concluded with an 'approved' status.
Rejected	File processing finished and the soft-proofing test was concluded with a 'rejected' status.
Failed	N/A

5.9 The use of Party structures in AdsML Materials request and response messages

Within AdsML Materials response and request messages the parties involved in the materials delivery workflow can be identified using a combination of specific and generic structures. The preparer, deliverer, and the intended recipient of materials are to be identified using the `adsml:MaterialsPreparerParty`, `adsml:DeliveringParty`, and `adsml:MaterialsRecipientParty` elements explicitly defined for this purpose. In Delivery Order messages, the ordering party is to be identified using `adsml:DeliveryOrderingParty`. In addition, in all materials messages a generic `adsml:OtherParty` allows other parties involved in the workflow to be recorded should this be required by a specific business workflow. In all cases, the materials preparer, deliverer, and the intended recipient of the materials **MUST** be identified using the dedicated elements supplied for this purpose. The generic `OtherParty` structure **MUST NOT** be used to record the details of the parties for whom explicitly defined structures are available.

5.10 Recording the delivering party in AdsML Materials request and response messages

In AdsML, *the delivering party is always considered to be the party that is sending or will send a materials delivery message (AM-M)*.

The AdsML Materials standard includes message groups for ad material delivery (AM-M* message family) and for ad material delivery order messages (AM-D* message family). To support the message exchanges the standard has structures to describe the following parties,

- `adsml:DeliveryOrderingParty` - the party who is ordering the delivery and is the 'delivery orderer'
- `adsml:DeliveringParty` - the party who is making (AM-M) or is being instructed (AM-DO) to make the delivery and is the 'delivering party'
- `adsml:MaterialsPreparerParty` - the party who has 'created' or 'prepared' the ad materials and so is the 'materials preparer'
- `adsml:MaterialsRecipientParty` - the party to whom the delivering party will deliver the materials and so is the 'intended recipient' of the ad materials
- `adsml:OtherParty` - to identify other parties in the material management workflow that have an interest in the ad material content that is being delivered
- `/Delivery/ThirdPartyServiceProvider` - a party performing a delivery service on behalf of the delivering party
- `/adsml-ma:MaterialsExpectations/adsml-ma:MaterialsProviderParty` - the party that will be making the content available to the publisher once it is ready. The materials provider will be represented as either the `adsml:DeliveringParty` or the `/Delivery/ThirdPartyServiceProvider` in the AM-M that the publisher receives.

The `adsml:DeliveringParty` element identifies the party that is sending an Ad Material message to make a delivery of ad materials. The `ThirdPartyServiceProvider` is used to identify a party providing a delivery service when the delivering party employs a third party to make the delivery on their behalf. Note that the actual use of these three elements may vary between business contexts.

For example, to illustrate using the scenario of a creative agency contracted to supply artwork to a publisher, the party roles would be as follows,

1. When the agency sends the Ad Material (AM-M) message to the publisher and actually delivers the content itself,
    - `adsml:DeliveringParty` - the creative agency
    - `adsml:MaterialsPreparerParty` - the creative agency
    - `adsml:MaterialsRecipientParty` - the publisher
    - `/Delivery/ThirdPartyServiceProvider` - none
2. When the agency sends the Ad Material (AM-M) message to the publisher but uses a third party delivery service to deliver the content on its behalf,
    - `adsml:DeliveringParty` - the creative agency
    - `adsml:MaterialsPreparerParty` - the creative agency
    - `adsml:MaterialsRecipientParty` - the publisher
    - `/Delivery/ThirdPartyServiceProvider` - the delivery service
3. When a delivery service sends the Ad Material (AM-M) message and the content to the publisher,
    - `adsml:DeliveringParty` - the delivery service
    - `adsml:MaterialsPreparerParty` - the creative agency
    - `adsml:MaterialsRecipientParty` - the publisher
    - `/Delivery/ThirdPartyServiceProvider` - none.

6 Use Cases and Recommended Solutions

This section provides a set of sample scenarios and their handling using AdsML Materials. The scenarios are divided into sections for materials delivery, status reporting, delivery order, resend request and 'kill order' transactions.

6.1 Materials delivery

6.1.1 Deliver an artwork file via an FTP upload

Scenario: A materials preparer wishes to initiate digital delivery of a PDF/X file which will be transmitted via FTP.

AdsML handling: The preparer's systems create an `AdMaterial` (AM-M) message. The `AdMaterial` message must contain the full details of the delivery in the `AdMaterialsRequestModule`, and must include the sender's unique materials ID in the `MaterialsIdentifier` element. The ad content artwork file (i.e. the PDF) is described as a single `Rendering` within the `AdContent` element. The delivery instructions for the ad content are specified using a `Delivery` element, in this case specifying the FTP as a digital delivery.

Notes: See response scenarios for response handling using AM-MR.

6.1.2 Deliver materials inline

Scenario: A materials preparer wishes to initiate digital delivery to the publisher with the material contained in-line within the AdsML message.

AdsML handling: The preparer's systems create an Ad Material (AM-M) message and this is then transmitted to the publisher to make the delivery, the content contained inline within the `AdContent` structure. As the content is inline there is no `Delivery` structure in the `AdContent`, only a `Rendering` and associated metadata as provided by the `AdContent` structure.

Notes: See response scenarios for response handling using AM-MR.

6.1.3 Accept a Materials delivery

Scenario: A publisher receives a materials delivery and provides a business level response accepting the materials delivery.

AdsML handling: The delivery recipient receives the `AdMaterial` message containing the content and responds with an `AdMaterialResponse` business message, this response message notifying that the content was successfully received.

Note: Normal response procedure is for the response to contain only identification and status content. A response may, though, mirror back the contents of the original message, if required.

6.1.4 Reject a Materials delivery

Scenario: The delivery recipient receives the `AdMaterial` message containing content that is considered unacceptable. For example, a preflight failure.

AdsML handling: The recipient responds with an `AdMaterialResponse` business message notifying the sender that the content has been rejected. The

AdMaterialResponse contains only an adsml:RequestDenied element, within which one or more ReasonForDenial elements state the reason for the delivery having failed.

Note: Rejection of a materials delivery will often trigger a redelivery of corrected materials according to the business rules of the implementation. However, "redelivery" is a misnomer – it is actually a NEW delivery transaction with a new MaterialsIdentifier.

Note: Rejection of a materials delivery based on results of preflighting performed by the recipient of the AM-M message may be given by an AM-MR or an AM-MS. If the preflighting is performed before an AM-MR response message has been sent, the rejection can be conveyed in the AM-MR message. In this case, use the adsml:RequestDenied element to indicate that the delivery has been rejected, and in the ReasonForDenial, provide the reason for the rejection. If an AM-MR has already been sent, send an AM-MS message to indicate that the status of these materials has now become unacceptable.

6.1.5 Reject part of a delivery

Scenario: A publisher receives a materials delivery containing multiple renderings. Technical processing shows that part of the delivered materials are acceptable and part of the materials is to be rejected for technical reasons.

AdsML handling: Rejection of part of a delivery cannot be done in an AdsML message. The entire delivery would be rejected and a new delivery initiated, the new delivery considered replacement content.

Notes: see the FAQ for delivering replacement content,

6.1.6 Respond to a new Materials delivery with external content, content successfully retrieved

Scenario: A publisher receives an AdMaterial (AM-M) message informing them of the delivery of ad content to an external location on the Web. The publisher collects the content by 'pulling' it. The publisher sends an AdMaterialResponse business message acknowledging successful retrieval of the content.

AdsML handling: The delivery recipient receives an AdMaterial message specifying the (Internet) location at which the content is available for retrieval. The recipient successfully retrieves the content and sends an AdMaterialResponse business message. The response message returns the status and identification information for the materials delivery, the adsml:Status elements (at AdMaterial, Rendering, and Delivery levels) indicating that the delivery has been successful. The AdMaterialResponse message may mirror back a confirmation copy of the original AdMaterial message data, optionally containing the AdContent itself if desired.

Note: The AM-MR may or may not report on the final state of the delivery. Depending on processing requirements, the AM-MR may provide business level acknowledgement of the receipt with an 'in progress' status with the final state reported by an AM-MS message sent in due course. The delivery and processing of a large materials file illustrates such a case.

Note: This example shows the digital delivery made by means of the Internet. Other delivery channels – for example the use of ISDN, delivery direct to a recipient system - would use different structures as appropriate to the case within the `DigitalDelivery` structure.

Note: Delivery using MIME multipart/related packaging. To make a delivery with MIME packaging the delivered content files would be considered external content, the CID URL of the MIME package referenced from within the `DigitalDelivery` structure.

6.1.7 Initiate a new Materials delivery with future physical delivery

Scenario: Ad content on digital media is to be delivered from a preparer to the publisher, the physical delivery made by a third party service provider (XYZCouriers). The preparer notifies the publisher that the content delivery has been initiated, who will be delivering the content and the time at which the delivery will be made.

AdsML handling: The preparer sends an `AdMaterial` (AM-M) message to the publisher, the message specifying what is in effect a future physical delivery – i.e. the content will arrive at your office by 5pm. The publisher's systems return an `AdministrativeResponse`. In this scenario no other response is necessary as the third party service provider will confirm to the sender that the delivery was made.

The `AdMaterial` message contains the full details of the physical delivery. Within the `AdContent` structure the material is described in a single `Rendering`. A `Delivery` element specifies the delivery data: a `PhysicalDelivery` structure describes the physical package and specifies the address to which it is to be delivered, the `MaterialsAvailableDateTime` indicates when the delivery is expected to occur, and the courier service is recorded as the `ThirdPartyServiceProvider`.

Note: The `adsml:MaterialsPreparerParty` and `adsml:DeliveringParty` parties can optionally be specified as children of the `AdMaterial` element in order to explicitly identify the party which prepared the artwork and the party which sends the AM-M message to the publisher. There may be a single party performing both creative and delivery roles, or there could be two different parties. Using these optional elements makes the business relationship transparent and so is a recommended approach for clarity. Note that in this scenario the third party service provider is assumed to be acting as an agent of the delivering party.

6.1.8 Initiate a new Materials delivery of content on multiple media with delivery by multiple methods

Scenario: Content is to be delivered by a production house on multiple media by different delivery methods. Ad content is to be delivered on digital media by a physical delivery (i.e. by courier). A second delivery of the same content is also to be made by digital means, the content sent using a digital delivery service. The production house notifies the publisher that the materials delivery has been initiated.

AdsML handling: The production house sends an `AdMaterial` (AM-M) message to the publisher, the message specifying that (the same set of) ad content is to be delivered by multiple means. A `Rendering` element describes the content that is to be delivered. The multiple delivery methods – digital and physical – are specified using separate Delivery elements. Each Delivery element uses a RenderingReference element to reference the Rendering. That the Rendering is delivered twice by more than one means is made explicit by it having more than one set of delivery data associated with it. The ThirdPartyServiceProvider element is used in the Delivery structures to identify the delivery service providers for the physical and digital deliveries.

6.1.9 Provide advanced notice of content to be delivered in the future, either digitally or physically

Scenario: A deliverer provides advanced warning of a delivery in progress where the materials will arrive at a future date. In the case of a physical delivery this accounts for the physical time lag between the courier collecting the material and actually making the delivery. In the case of a digital delivery the content is to be made available at a location for retrieval and the content up loader notifies the intended recipient that the content will become available.

AdsML handling: The handling is as for digital and physical delivery scenarios. Note that for the digital delivery that the MaterialsAvailableDateTime metadata within the Delivery structure will, as with the physical delivery, specify a date in the future.

Note: None.

6.1.10 Deliver a PDF via a third party digital artwork delivery service

Scenario: Ad content is to be delivered using a third party service provider. In this scenario the party with the business responsibility for making the delivery makes the actual delivery by means of a third party service provider operating a delivery network.

AdsML handling: The handling is as for digital delivery with the exception that the ThirdPartyServiceProvider element within the Delivery structure will identify the delivery service provider.

Note: See 'Note' to FAQ 6.1.8 above.

6.1.11 Initiate a new Materials delivery with delivery of multiple renderings by multiple methods

Scenario: A preparer initiates delivery of a PDF and a hard proof of the same. This is a materials delivery by multiple methods, with delivery by physical and digital means. The PDF is sent digitally and a hard proof of the ad is sent by courier.

AdsML handling: The preparer sends an AdMaterial (AM-M) message to the publisher. The AdMaterial message contains multiple renderings and delivery instructions. One Rendering element describes the digital content, while a second Rendering element describes the hard proof. The RenderingType element clearly identifies one rendering as a digital file and one as the hard proof. Each Rendering has a delivery method associated with it, specified using separate Delivery elements. The digital file is to be delivered by digital delivery, the hard proof by physical delivery using a third party. In this case, there is a natural time delay in the delivery given that the digital delivery is made with the arrival of the AM-M while the physical delivery is still in progress.

Note: Upon receipt of the AM-M the publisher returns an Ad Material Response (AM-MR) message, the digital delivery having been completed and the physical delivery still in progress. Once final processing of the delivery has been completed, that is both digital and physical deliveries have been completed and processed, then the publisher sends an Ad Material Status message reporting the final state of the delivery.

6.1.12 Deliver replacement materials

Scenario: A creative agency initiates a delivery of new ad materials to replace ad materials that were previously delivered.

AdsML handling: To deliver the replacement materials, a new materials delivery is instantiated. The creative agency sends an `AdMaterial` (AM-M) message to the publisher, the message specifying the details of the ad content being delivered. To indicate that this delivery replaces content previously delivered, the `ReplacesMaterialsReference` element explicitly identifies the materials being replaced by the new delivery, making the identification by referencing the `MaterialsIdentifier` of the original delivery transaction. If the reason(s) for replacing the materials are stated, then the repeatable `ReasonForReplacement` element is used for this purpose; the reason(s) are recorded in codified form.

Notes: multiple handling scenarios are possible,

- The delivery of replacement content could be triggered by a rejection of a delivery (by an AM-MR or AM-MS), or could be a spontaneous new delivery by the deliverer.
- If the earlier materials were accepted, an alternative approach would be for the creative agency first to send a 'Kill Order' killing the previously-delivered materials. Whether trading partners choose to explicitly 'kill' materials before replacing them is a Trading Partner Agreement (TPA) matter.
- If the materials in an AM-M are intended to replace materials that either were never accepted in the first place, or have previously been the subject of a kill order, the use of the `ReplacesMaterialsReference` element is optional.
- Replacement content deliveries cannot be partial. Any delivery of replacement content must deliver the entire set of ad content; it is not possible to just deliver a changed version of one of an ad's renderings.
- Changes to an ad's metadata, for example a change to a booking ID in the `BookingInformation` metadata, cannot be done unless the ad materials are redelivered, i.e. a new AM-M is initiated.

6.1.13 Deliver classified ad content along with its booking information

Scenario: Ad content is to be sent to the publisher in the same file as its actionable ad booking information.

AdsML handling: This should not be done in AdsMLMaterials, because an AM-M message may only contain an *informational* (rather than actionable) copy of the booking information. Correct handling is to use the AdsMLBookings `AdOrder` (AD-O) message, which also permits the transmission of inline content.

Notes: none

Copyright © 2010 AdsML Consortium. All rights reserved.

6.1.14 Deliver a set of classified ads to another party

Scenario: A bulk send of classified ads is to be sent to a single publisher.

AdsML handling: Each classified ad delivery constitutes a discrete materials delivery and is represented as one using the `AdMaterial` (AM-M) message. To make the bulk delivery, a single AdsMLMaterials message is sent, that message containing the multiple `AdMaterial` message elements within it.

Notes: none.

6.1.15 Deliver only the structured description of an advertisement

Scenario: A sender wishes to deliver just a structured description of the contents of an advertisement, without providing a visual representation of it. The recipient will render the published advertisement using its own design templates. (This is a common scenario when, for example, republishing a classified ad in a web environment.)

AdsML handling: Within the `AdContent` element, the sender populates just the `StructuredDescriptions` element (and optionally also `adsml:DescriptionLine`, `AdContentSearchText` and `AdContentText`). It is not necessary to send a `Rendering` in this scenario, therefore there will also be no `Delivery` element.

Notes: If supplemental materials such as photographs need to be sent along with the structured description, then it will be necessary to package and deliver them as if they were a `Component`. Trading partners will need to agree on values for `ComponentType`, `adsml:Usage`, and `ContentHandlingInstructions` to support the necessary processing.

6.1.16 Deliver artwork components for make-up by a Repro House

Scenario: A sender wishes to deliver artwork components for make-up by a creative house, providing the raw ad text and their company logo as an EPS image contained inline within the AdsML message.

AdsML handling:
The sender's systems create an Ad Material (AM-M) message and this is then transmitted to the repro house to make the delivery, the ad components contained inline within the `AdContent` structure. Within `AdContent`, the sender populates the `AdContentText` element with the raw text of the ad. The component artwork file (i.e. the EPS image file) is contained inline as a single `Component` within the `AdContent` element. The image is encoded in an XML friendly fashion and embedded. Using values taken from controlled vocabularies the trading partners have agreed to use, the `ComponentType` is recorded as '`ImageFile`' and its use as the logo is specified by assigning the `adsml:Usage` element a value of '`Logo`'; instructions describing how to make-up the ad are recorded as `ContentHandlingInstructions`. The `ContentProperties` structure records technical metadata about the image file, specifying the encoding applied, MIME type and other characteristics needed to support application level processing. As the content is inline there is no `Delivery` structure in the `AdContent`.

Notes: None.

6.1.17 Deliver encoded or encrypted content

Scenario: A materials preparer wishes to initiate digital delivery to the publisher with the material contained in-line in the AdsML message. The digital file has been encoded and encrypted.

AdsML handling: The preparer's systems create an Ad Material (AM-M) message, the content contained inline within the `AdContent` structure, as a `Rendering`. As the content is inline there is no `Delivery` structure in the `AdContent`, only a `Rendering` and associated metadata as provided by the `AdContent` structure.

The artwork – a TIFF file – is first encrypted and then encoded using the binhex encoding method and encryption method agreed in their AdsMLMaterials 'configuration checklist'. The encoding and encryption applied are recorded using the `ContentDataEncoding` and `EncryptionMethod` content metadata properties. The other content metadata properties are used to record metadata about the TIFF file prior to its encoding and/or encryption (MIME type, format, format profile, size in bytes, and file name).

The message is then transmitted to the publisher to make the delivery

The delivery is successful and the ad content is accepted.

Notes: See response scenarios for response handling using AM-MR.

Note that AdsML assumes a processing model in which encoding and encryption are reversed first. That is, the recipient of a file first decodes then decrypts the received file, and the result then conforms to the stated MIME type, file size and other metadata recorded in the `ContentProperties`. For example, in the case of a TIFF file that had been binhexed, the MIME type would not record 'BinHEX' but rather 'TIFF'.

6.1.18 Deliver content with associated soft-proof metadata

Scenario: A materials preparer delivers ad materials to a publisher, including metadata describing the outcome of a soft-proof approval test.

AdsML handling: An `AdMaterial` business message is created. The soft-proof metadata is recorded at the `Rendering` level using the `SoftProofTicket` element.

Notes: The content model of the soft-proof ticket is optional and there is no constraint specified on what or which elements must be used. The user, therefore, is free to choose the level of soft-proof metadata appropriate to their usage scenario and workflow.

6.1.19 Deliver artwork for a specific region

Scenario: A materials deliverer provides regionalized artwork to a publisher, explicitly identifying the artwork is for the 'Northern' regional edition of the publication.

AdsML handling: An `AdMaterial` business message is created. Within the `AdContent` structure, metadata describing the ad materials and their delivery are recorded. Specifically, the `adsml:UsageLabel` is used to record the regionalization: the `CodeList` is given a value of '`Region`' to specify that regionalization is being recorded; the regional value – 'Northern' – is recorded as the CodeValue.

Notes: In a scenario where multiple regional specific versions of an ad are delivered this is done using individual AdMaterial messages. A bulk delivery could be made by sending a single AdsMLMaterials message containing the multiple AdMaterial message elements within it.

6.1.20 Deliver multiple renderings for an interactive advertisement

Scenario: A creative house provides high and low res versions of an interactive ad, supporting a scenario where either the low or high is served depending on the user's connection speed. The finished renderings are delivered digitally.

AdsML handling: Each version of the ad content is conveyed as a separate rendering. Within the AdContent element two Rendering elements are populated. Use the RenderingType code to identify them as, respectively, "HighRes" and "LowRes" renderings. These values are taken from the AdsML Rendering Type CV.

The same delivery method is used to deliver both renderings. Delivery is specified using a single Delivery element, which is referenced by each Rendering.

Note: It would be possible to convey the renderings in two separate AdsML Ad Material messages (AM-M). 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.

6.1.21 Deliver ad materials with textual metadata describing the materials in more than one human language

Scenario: A deliverer provides ad materials with human-readable metadata describing the ad content and how to handle it in three languages - English, French and German.

AdsML handling: Handling is the same as digital and physical delivery scenarios. Multilingual metadata is provided for the materials in English, French and German using the AdContent and the Rendering's DescriptionLine elements and the ContentHandlingInstructions' Text elements. The DescriptionLine and Text elements are repeated to record the different language versions of the text. 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 'content handling instructions' might contain three Text elements, one each in English, French and German and identified as such by their *xml:lang* attributes. If the French version is the original text from which the English and German translations are derived, then the French language Text element would have a *source* attribute with the value of 'true'.

Notes:

- Multilingual metadata can be provided in all element contexts where textual elements are repeatable and the AdsML i18n attributes are available. For example, elements taking codified values or user-defined properties have repeatable Description labels to support multilingual metadata.

Copyright © 2010 AdsML Consortium. All rights reserved.

- Multilingual metadata should not be confused with multilingual versions of advertisement content. Multilingual versions of advertisements will always be treated as different ad content and delivered in separate `AdMaterial` messages. Note that:
    - The same logical constraint applies to `Rendering`(s) provided within the `AdContent` message: a rendering sheet will always be for one set of ad content (i.e. an ad in a single language) but the `Rendering` element may itself contain multilingual metadata.
    - This constraint does not apply to ad materials `Component`(s) as components are not finished ad content. For example, an ad production scenario may use an `AdMaterial` message to deliver ad components for the make-up of French and German versions of an ad in a single AdMaterial message. Once the materials have been made-up, though, the French and German versions would each be treated as separate sets of ad content.
  - The *`i18nAttributes`* group provides three optional attributes:
    - *`xml:lang`* - to specify the human language (e.g. 'en' for English)
    - *`dir`* – to specify the reading direction of the text
    - *`source`* – to specify if the text is the original source text from which other translations of the text have been derived.

  In a simple usage scenario only the *`xml:lang`* attribute would be used when content is provided in more than one human language. See the *AdsML Type Library Specification & Schema* and *E-Commerce Usage Rules & Guidelines* documents for more information about the *`i18nAttributes`*.

6.2 Materials status

6.2.1 Request the status of a Materials delivery

Scenario: The sending party of a materials delivery requests the status of a materials delivery in progress (e.g. was it received ok, is the content fit for purpose). The AM-MR already received has indicated that the materials have yet to be processed to determine technical satisfactoriness. The sender requests status of the delivery by sending a Material Status Enquiry message (AM-MSE).

AdsML handling: An AM-MSE is sent, the message identifying the materials delivery whose status is sought by the `MaterialsIdentifer` element of the AM-MSE.

Note: The AM-MSE could be sent before receipt of the AM-MR.

6.2.2 Report the status of a Materials delivery

Scenario: A recipient of a materials delivery reports on the status of a materials delivery in the scenario that the AM-MR sent on receipt of the delivery acknowledged initial receipt of the materials delivery but did not report on the delivery's final state. The recipient reports the final status of the delivery by sending a Material Status message (AM-MS).

AdsML handling: An AM-MS is sent, the message identifying the state of the delivery.

Note: Multiple AM-MS may be sent if required, for example in the case of responding to an AM-MSE before the materials have been technically processed. However, in the simple case a single AM-MS would be sent by the recipient to report the final state of the delivery.

6.3 Delivery Order

6.3.1 Order delivery of a single set of materials to one publisher

Scenario: A creative agency has prepared an advertisement that they want to send to a publisher. The publisher requires that all incoming ad materials be sent by a specific deliverer with which the creative agency has established a business relationship. The agency now needs to send the materials to the deliverer, along with instructions as to where and when the deliverer should deliver them.

AdsML handling: The creative agency sends an `AdMaterialDeliveryOrder` (AM-DO) message to the deliverer containing a single `ForDeliveryTo` element in which the intended recipient of the materials (the Publisher) is specified. The `ForDeliveryTo` element also contains the address to which the materials should be sent, labeling and delivery instructions, the due date/time, and booking information which describes the intended publication, publication date, etc.

The creative agency populates the `AdContent` element with metadata describing the ad materials that are to be delivered. The materials themselves are either contained in-line inside the message, or information is provided which tells the delivery agency where and how to retrieve the materials. (All of the capabilities and metadata described in this paragraph are the same as in an AM-M Materials Delivery message.)

Note: In this scenario, the agency could alternatively provide the materials to the deliverer by use of an AM-M message rather than an AM-DO message. However, it is **RECOMMENDED** that an AM-DO message always be used when transmitting ad materials and/or delivery instructions to a delivery agency.

6.3.2 Order delivery of a single set of materials to multiple publishers

Scenario: A creative agency has prepared an advertisement that it wants to send to a deliverer for transmission to multiple publishers.

AdsML handling: The creative agency sends an AM-DO message to the deliverer containing multiple `ForDeliveryTo` elements, one for each intended recipient of the materials (e.g. the publishers). Each `ForDeliveryTo` element contains the address to which the materials for that publisher should be sent, labeling and delivery instructions, the due date/time, and booking information which describes the intended publication, publication date, etc.

The creative agency populates the `AdContent` element with metadata describing the ad materials that are to be delivered. The materials themselves are either contained in-line inside the message, or information is provided which tells the delivery agency where and how to retrieve the materials. (All of the capabilities and metadata described in this paragraph are the same as in an AM-M materials delivery message.)

Copyright © 2010 AdsML Consortium. All rights reserved.

6.3.3 Order delivery of two sets of ad materials to the same list of publishers

Scenario: A creative agency has prepared two advertisements that it wants to send to a deliverer for transmission to a single set of publishers. (That is, both advertisements should be transmitted to the same list of publishers.)

AdsML handling: It is not possible to transmit two different advertisements in a single AdMaterialsDeliveryOrder. The creative agency must send two AdMaterialsDeliveryOrders, each containing one set of artwork along with the list of publishers to which that artwork should be sent. These may be transmitted in a single AdsMLMaterials XML message instance, or in two different XML messages.

6.3.4 Order additional deliveries of materials that were the subject of a previous delivery order

Scenario: A creative agency wishes to send a delivery order which references ad materials that have already been delivered to the deliverer, usually because they were the subject of a previous delivery order. The creative agency has reason to believe that a copy of the materials is still in the deliverer's possession, and thus available for retransmission.

AdsML handling: The creative agency sends an AM-DO message as usual, but instead of populating the AdContent element, it conveys the ID of the materials that are to be delivered in the ArchivedContent/MaterialsReference element. (Additional identifiers for the materials can also be provided in the sibling AuxiliaryMaterialsReferences element.) This indicates to the deliverer that they are to re-send the referenced materials.

Note: If the deliverer does not possess archived copies of the referenced ad materials, it cannot accept the Delivery Order as presented. In this case, it would reject the delivery order stating this as the reason for denying the request.

6.3.5 Accept a Delivery Order

Scenario: A delivery agency wishes to signify acceptance of a delivery order it has just received.

AdsML handling: The deliverer sends a Delivery Order Response (AM-DOR) message which contains the same DeliveryOrderIdentifier as the order that is being accepted and does not contain a RequestDenied element. This combination constitutes acceptance of the order. The deliverer may use the optional Status element to convey further information about its processing of the order.

Note: It is recommended that the response message be fully populated with the details of the delivery order and the identifier of the ad materials that are to be delivered, although ad materials themselves should not be included in a delivery order response message.

6.3.6 Reject a Delivery Order

Scenario: A delivery agency wishes to reject a delivery order it has just received.

AdsML handling: The deliverer sends a Delivery Order Response (AM-DOR) message which contains the same DeliveryOrderIdentifier as the order that is being rejected, and populates RequestDenied/ReasonForDenial with an agreed code (or text) explaining the reason for the rejection. The rest of the message is left empty.

6.3.7 Accept part of a Delivery Order

Scenario: A delivery agency wishes to accept just part of a delivery order that it has received, for example because it is unable to deliver to one of the intended recipients.

AdsML handling: It is not possible to accept or reject just part of a delivery order. The deliverer should send an AM-DOR message rejecting the entire order, using `RequestDenied/ReasonForDenial` to explain the reason. The ordering party can then send another (replacement) delivery order which fixes the problem.

Note: An alternative is for the deliverer to send an AM-DOR which accepts the order, and then use non-AdsML mechanisms (e.g. a phone call) to communicate that actually only part of the order can be executed. The ordering party can then send a Delivery Order Change which removes those parts of the original order that cannot be executed by this deliverer. See the next scenario.

6.3.8 Change a Delivery Order

Scenario: An ordering party has previously sent a delivery order that was accepted by the deliverer, but now wishes to change the contents of that order, for example by adding or removing an intended recipient.

AdsML handling: The orderer sends a Delivery Order Change (AM-DOC) message using the same `DeliveryOrderIdentifier` as the original order. The change message completely specifies all the details of the changed order. Optionally, the orderer uses the `adsml:ChangeSpecification` element to identify the sections of the order that have changed.

Unless datagram communications are being used, the recipient of the Delivery Order Change message now sends a Delivery Order Response (AM-DOR) to indicate whether it is able to accept the changes to the order.

Note: Rejection of a Delivery Order Change message means that the originally accepted delivery order is still in effect – for example, because the deliveries have already occurred or are being processed and can no longer be cancelled. In this case, after the deliverer has rejected the change order the orderer has the option of cancelling the remainder of the delivery by sending a cancellation. However, there is no AdsML mechanism for partially accepting a change.

6.3.9 Cancel a Delivery Order

Scenario: An ordering party wishes to cancel a previously-accepted delivery order.

AdsML handling: The ordering party sends a Delivery Order Cancellation (AM-DOX) message using the same `DeliveryOrderIdentifier` as the original order, and including `adsml:ReasonForCancellation` to explain the cancellation. The rest of the cancellation message may optionally specify all the details of the cancelled order, or may be left empty.

Unless datagram communications are being used, the recipient of the Delivery Order Change message now sends a Delivery Order Response (AM-DOR) to indicate whether or not it is able to accept the cancellation of the order.

Note: Rejection of a Delivery Order Cancellation message means that the originally accepted delivery order is still in effect – typically because the deliveries have already occurred.

6.3.10 Reject a Delivery Order Cancellation because some deliveries are underway

Scenario: A delivering party is unable to accept a cancellation because some or all of the ordered deliveries have been initiated.

AdsML handling: The delivering party sends a Delivery Order Response (AM-DOR) message using the same `DeliveryOrderIdentifier` as the original order. Note that in order to indicate that some of the deliveries are already in process, a simple reply using `adsml:RequestDenied` to state the reason for rejecting the cancellation is not sent. Rather, the delivering party **SHOULD** populate the message fully using the `ForDeliveryTo/adsml:Status` to report the status of the individual deliveries in the order and so unambiguously identify those deliveries that have been cancelled and those deliveries which could not be cancelled.

Note: In the case where a Delivery Order Cancellation message is received after all of the deliveries have been executed then the Delivery Order Response message does not need to report the status of each individual `ForDeliveryTo` and can instead be a simple 'request denied' response message.

6.3.11 Respond to a delivery order that specifies archived content that does not exist

Scenario: A creative agency sends an AM-DO message to a deliverer that requests delivery of archived content which the deliverer does not have in its possession and therefore cannot deliver.

AdsML handling: The deliverer cannot accept the Delivery Order as presented. It **SHOULD** send a Delivery Order Response message (AM-DOR) which denies acceptance of the delivery order and indicates in the `ReasonForDenial` element that the referenced archived content does not exist or is not available. Upon receipt of this message, the agency can send a new, replacement Delivery Order (AM-DO) which uses the `AdContent` element to transmit a copy of the necessary materials to the deliverer.

Note: An alternative approach would be for the deliverer to pick up the phone and inform the agency that the materials are not available, at which point the agency could provide the missing materials either by non-AdsML means, or by sending a Materials Delivery message (AM-M) to the deliverer which contains them. Note that AdsML does not provide metadata in the AM-DOR and AM-M messages which directly support this alternative approach.

6.3.12 Order a delivery to multiple recipients specifying the intended usage of the content by each recipient

Scenario: An Agency orders the delivery of a set of artwork consisting of two renderings of the ad to three publishers, and wishes to specify to each recipient which rendering they should use.

AdsML handling: AdsML does not provide a mechanism by which a sender of materials can explicitly direct the recipient to use one rendering rather than another. In this scenario, the sender must populate the `RenderingType` and `Usage` codes in such a way that the recipient publishers will be able to tell which renderings are relevant to their situation.

Note: It is also possible to split the delivery into two Delivery Orders, each of which contains just one of the renderings, in order to ensure that the correct rendering is processed by each recipient.

6.4 Delivery order status

6.4.1 Request the status of a Delivery Order

Scenario: The sending party of a Delivery Order requests the status of the order (i.e. was the order received ok, is the content available, has it been executed). The AM-DOR already received has indicated that the order has yet to be processed to determine technical satisfactoriness.

AdsML handling: The sender requests the status of the delivery order by sending a Delivery Order Status Enquiry message (AM-DOSE), the message identifying the delivery order whose status is sought by the `DeliveryOrderIdentifer` element of the AM-DOSE.

Note: The AM-DOSE could be sent before receipt of the AM-DOR.

6.4.2 Report the status of a Delivery Order

Scenario: A recipient of a delivery order reports on the status of the processing of that delivery order, in the scenario that the AM-DOR sent on receipt of the delivery order did not report on its final state.

AdsML handling: The recipient reports the final status of its handling of the delivery order by sending a Delivery Order Status message (AM-DOS), the message identifying whether the order was executed successfully.

The value of the highest-level `adsml:Status` element in the status message **MUST** indicate the status for the delivery order as a whole. If any part of the order has not yet been successfully executed, the top level `adsml:Status` element (i.e. `DeliveryOrderStatus/adsml:Status`) must reflect that fact. The status of each requested delivery within the order can be conveyed via the `adsml:Status` element inside each instance of `ForDeliveryTo`.

Note: Multiple AM-DOS messages may be sent over time if required, for example in the case of responding to an AM-DOSE before the order has been fully processed. However, in the simple case a single AM-DOS would be sent by the recipient to report the final state of the delivery order.

6.5 Resend Request

6.5.1 Request another copy of the materials for a specific appearance of an advertisement

Scenario: A publisher has misplaced or damaged the materials for a specific insertion or appearance of an advertisement. It needs to request another copy of those materials.

AdsML handling: The publisher sends a Resend Request message (AM-RES) to the original provider of the materials. In the message, the publisher populates the

Copyright © 2010 AdsML Consortium. All rights reserved.

ResendMaterialsByReference/BookingReferences element with the placement and booking identifiers for the insertion in question. If the publisher knows the materials identifiers of the missing ad content, it also populates the ResendMaterialsByReference/AdContentReferences element with as many of those materials identifiers as it has available.

The recipient of the message should then respond to it by re-sending the referenced materials.

Note: The publisher may wish to send two copies of this AM-RES message: one to the preparer of the original artwork, and one to its deliverer, if a separate delivery agency was used. The two messages will be identical except for header metadata such as MessageTo. There is no limit to the number of parties to which a publisher may send copies of the same AM-RES message if the publisher is not sure which party is best able to resend the missing materials.

6.5.2 Request resend of all materials sent on a particular day

Scenario: A publisher's systems were down for a day, so it is concerned that ad materials may have been sent to it during that day which it did not properly receive.

AdsML handling: The publisher sends a Resend Request message (AM-RES) to each delivery agency from which it normally receives artwork, in which it requests that the agencies resend all materials that were originally sent on the day in question. In the message, the publisher populates the ResendMaterialsByFilter/DeliveredBetween element with the starting date/time and ending date/time of the period in question.

Each recipient of the message should then respond to it by re-sending all materials which were previously delivered to that publisher during the specified time window.

Note: The publisher may narrow the request by specifying, for example, a particular publication and publication/issue date for which the materials were intended.

6.5.3 Request resend of all materials intended for two specific publications and publication dates

Scenario: A publisher needs to request the resend of all materials that were intended for the next issue of two of its publications.

AdsML handling: It is not possible to define a filter in a resend request that combines two or more sets of criteria whose results should be added together (in this case, the two different publication names and dates). The publisher should split the request into two AdMaterialResendRequest messages, one for each of the publications and publication/issue dates in question. Together the two messages should result in the resending of all missing materials.

Note: If a complex additive filter is required, it is probably better to use non-AdsML mechanisms (e.g. a phone call) to convey the request.

6.6 Materials ‘Kill’ order

6.6.1 Request the ‘kill’ of materials delivered by a Materials delivery

Scenario: Delivered materials are to be ‘killed’, business reasons requiring confirmation that the materials have been suppressed and so there is no risk of their being further processed. A materials ‘kill’ order is issued (AM-MX).

AdsML handling: An AM-MX transaction is initiated, the message identifying the materials to be ‘killed’ by the `MaterialsIdentifer` element of the AM-MX.

Note: The AM-MX **MUST** be responded to by an AM-MXR.

6.6.2 Respond to a Materials ’kill’ order

Scenario: A recipient of a materials ‘kill’ order responds, reporting the state of the ‘kill’ order transaction by sending a Material Kill Order Response message (AM-MXR).

AdsML handling: An AM-MXR is sent, the message reporting the final state of the delivery.

Note: Only one AM-MXR can be sent and **MUST** report the final state of the material ‘kill’ order transaction.

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 Message exchange mode

There are two defined message exchange modes:

1. Datagram model only
2. Full Request-Response

Trading partners must select one of these two exchange modes.

Note that the ability to send and receive Administrative Responses is a fundamental feature of AdsML messaging and is required in both modes.

7.2 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.

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

7.3 Use of encoding and encryption of ad content payload

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

- Encoding of inline content

Copyright © 2010 AdsML Consortium. All rights reserved.

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

7.4 Replacement content

Trading partners must agree on whether or not they choose to solely identify replaced content by using the Ad Material (AM-M) message's replaces materials reference alone or if they additionally issue a 'kill' order to explicitly suppress replaced content.

- ReplacesMaterialsReference flag
- Materials 'kill' order request-response

Note that it is always possible to use non-AdsML mechanisms to communicate this information.

7.5 Use of status reporting

Trading partners must agree on whether or not they support the use of AdsML status messages to report on the status of delivered materials and of delivery orders.

- Materials delivery status reporting (broadcast model)
- Materials delivery status request-response
- Materials delivery order status reporting (broadcast model)
- Materials delivery order status request-response

Note that it is always possible to use non-AdsML mechanisms to communicate this information.

7.6 Use of a Delivery Order Change message

Trading partners must agree on whether or not they support an AdsML message flow in which a delivery order can be changed using the AdsML Delivery Order Change (AM-DOC) message.

Note that it is always possible to use non-AdsML mechanisms to communicate this information.

7.7 Use of a Delivery Order Cancellation message

Trading partners must agree on whether or not they support an AdsML message flow in which a delivery order can be cancelled using the AdsML Delivery Order Cancellation (AM-DOX) message.

Note that it is always possible to use non-AdsML mechanisms to communicate this information.

Copyright © 2010 AdsML Consortium. All rights reserved.

7.8 Use of a Delivery Order message to communicate only ‘materials delivery expectations’

Trading partners must agree on whether or not they support the use of the AdsML Delivery Order (AM-DO) message only to communicate material delivery expectations.

Note that it is always possible to use non-AdsML mechanisms to communicate this information.

7.9 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.

Copyright © 2010 AdsML Consortium. All rights reserved.

8 References – (Non-Normative)

- [CIP4 Organization] CIP4™ Organization. *JDF Specification, Release 1.2*. International Cooperation for Integration of Processes in Prepress, Press and Postpress (CIP4), 2004. (http://www.cip4.org/)
- [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)

Copyright © 2010 AdsML Consortium. All rights reserved.

9 Appendix A: Acknowledgement for contributions to this document

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

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:

- Israel Viente (Vio Worldwide Ltd.) israel_viente@il.vio.com
- John Iobst (Vice President Newspaper Operations & Research, NAA) iobst@naa.org
- The AdsML Technical Working Group

Copyright © 2010 AdsML Consortium. All rights reserved.