AdsML® Framework for E-Commerce Business Standards for Advertising

AdsMLBookings 2.5.0
Part 1
Usage Rules & Guidelines

Document Authors: AdsML Technical Working Group

Document ID: AdsMLBookings-2.5.0-SpecP1Usage-AS-2

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

Document Status: Approved Specification

Document Date: 15 May 2011

Draft Number: 2

Table of Contents

1 AdsMLBookings Standard Documentation

1.1 Document status and copyright

This is the Approved Specification of the *AdsMLBookings 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 Acknowledgements: The AdsML Non-Exclusive License Agreement is based in part 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 AdsMLBookings-2.5.0-SpecP1Usage-AS-2, is freely available. It can 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 AdsMLBookings standard. AdsMLBookings is an XML-based language used for encoding and routing advertisement booking transaction messages.

1.6 Audience

The intended audience for this document is primarily user and vendor organizations who seek to implement the AdsMLBookings 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 provides rules and guidelines for using AdsMLBookings messages to address specific business requirements. A companion document, *AdsMLBookings – Part 2 - Specification & Schema*, serves as the reference guide to the AdsMLBookings 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. AdsMLBookings 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.(S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. Internet Engineering Task Force (IETF), Request for Comments: 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt)

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, ‘AdsMLEnvelope’, ‘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 some cases, element names mentioned in usage guidelines and narrative text in this document do not include their namespace prefix. For example, the element adsml-ma:Rendering is sometimes referred to as simply 'Rendering'. 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, AdOrder.

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 AS-2	15 May 2011	Minor changes. Added new use case descriptions	UW, JC, TS
2.5 AS-1	15 April 2010	First Approved version for AdsMLBookings 2.5 Previous change history removed	TS
2.0 AS-1	May 30 2008	First Approved version for AdsMLBookings 2.0 Previous change history removed	TS
1.1.1-AS-3	1 Oct 2006	Changed to Controlled Vocabulary 3.0. No other changes.	UW
1.0.1 AS-2	1 October 2006	Added data overview diagram and message exchange diagrams AdsML references updated to reflect Registered Trademark status	TS
1.0 AS -1	1 June 2006	First approved version	TS, UW

1.9.1 Changes in version 2.5

This version is a major upgrade including new functionality and changes to previously used structures. It is not backwards compatible, i.e. document instances valid to the 2.0.x schemas are not valid for 2.5 schemas.

Version 2.5 of AdsML Bookings provides dedicated support for interactive media and a number of smaller improvements. Much of the functionality that was originally designed in order to support Interactive bookings has ended up being made available to all media types.

Booking interactive advertisements

Dedicated support has been added for the booking of interactive advertisements through the availability of a new media-specific `Placement.Interactive` element. This new type of placement is intended to be used for booking of banner, rich media and other interactive advertisement

The media-specific extensions for `Placement.Interactive` include:

- Ability to define the type of thing being charged for (e.g. Impressions vs. Clicks) and the target count of them
- Ability to specific capping and throttling information
- Ability to identify the ad server that will be used, the type of rich media (if any), and other similar details

Structures at the media-agnostic levels have been reviewed and validated against requirements from the interactive context. Some existing structures such as `SchedulingType` and `ProofOfPublication` have been updated for usage across all media types, supporting interactive requirements as well as requirements from other media types.

Distribution and Targeting

- Using the new `PlacementTarget` structure, it is now possible to define the type of thing being purchased and the target count of them, e.g. "50,000 impressions" or "1,000 website registrations".
- Distribution has been renamed to `DistributionTarget` in order to clarify the difference between the intended (targeted) distribution, as shown in a Bookings message, and the eventual actual result (`DistributionResult`, which is an element in Proof of Publication and Financials messages). Also, `DistributionTarget` is now a sibling rather than a child of `Publication`. This allows for a single set of distribution instructions in a placement to apply to multiple publications.
    - This is a backwards-incompatible change.

Materials and Production Details

- `ProductionDetail.Interactive` includes the following new structures to support interactive orders:
    - `CappingSpecification`
    - `ShareOfVoice`
    - `AdServingSystem`
    - `TechnicalAdFormat`
- A `Duration` element has been added to `ProductionDetail.Interactive` and `ProductionDetail.Generic` to support descriptions of broadcast and interactive advertisements.
- The `Size` structure is now identical for three of the media types: Inserts, Interactive and Generic. In practice the primary change is that `Size` now always includes an optional `Area` element, which can be ignored in contexts where it is not required.

  - Size in NewspaperMagazine is based on the same structure, but with the additional Gutter element.
- It is now possible to convey multiple versions of the ordered Size of an ad, for example to show the same size in different units of measure.
- When delivering ad materials, tearsheets or any other type of rendering, it is now possible to convey the size of the rendering.
- It is now possible to convey in MaterialsExpectations the TechSpecDetails that should be used for the advertisement that is being booked. This information can take the form of either the URL for an external location or an in-line copy of the relevant document.

Guarantees

- Using the new Guarantees structure, it is now possible to indicate that an entire placement, placement group or order has been "guaranteed", and to convey the nature of that guarantee as a set of codes or text strings as appropriate.
  - This new element takes the role of the Guaranteed flag that used to be available in Positioning. For systems which have implemented guarantees this is a non-backwards-compatible change.

Scheduling

- The Scheduling and InsertionPeriod structures have been enhanced so as to better support the scheduling complexities of non-print media, and their structures have been aligned so that the same capabilities (and structure) are available for all media types.
  - This is a non-backwards-compatible change.
- In order to support the concept of "throttling", an optional EventCount element has been added to Scheduling and InsertionPeriod. This means that a target event count can be associated with each sub period during the lifespan of the order. For NewspaperMagazines, EventCount takes the role of the previous NumberOfOccurences element.

Pricing, Payments and Billing

- It is possible to identify the ProvenanceParty that will provide third-party performance verification, to be used for Proof of Performance and Invoicing purposes later in the workflow.
- It is now possible to convey textual AdditionalPaymentInstructions regarding how payment should be made.
- It is now possible to indicate that a placement, placement group or entire order is CostExempt, along with the associated reason. This supports workflows in which "free" ads are handled differently than paid ones.
- It is now possible to transmit a price component consisting of just the rate that will be used to calculate that component (e.g. an agency discount or commission percentage), without needing to reference a specific amount.
- It is now possible when booking a long-running ad or campaign (e.g. one that will span several months) to convey BillingInstructions governing how often

invoices should be generated and how the total booking should be allocated across those invoices.

- There is now an optional `Currency` element associated with each price in the order. Therefore, an order may (if desired) contain prices in more than one currency.
- It is now possible to specify the `ExchangeRate` that was used to calculate any price in the order, to support situations where one or more of the prices were converted from a rate card in a different currency.
- The deprecated attribute `scheduleEntryReference` has been removed from `NamedPriceType` (which is used by `PriceComponents`). This functionality has been replaced by `ScheduleEntryReference`, a repeatable element in the same structure.

Terms and Conditions

- All of the information relating to Terms and Conditions in a booking has now been consolidated in a single `TermsAndConditions` structure, with optional sub-structures to specifically identify the terms relating to `Cancellations` and `Claims` as well as the previously existing `adsml:DisclaimerText`.
    - This is a backwards-incompatible change.
- It is now possible to reference or convey the full `adsml:TermsAndConditionsDetails` that apply to a booking, either as an embedded PDF or as a reference (e.g. URL) to the web page where they can be found.

General

- It is now possible to convey information that is targeted for human (rather than machine) consumption, e.g. `adsml:Description` and `adsml:DescriptionLine`, in more than one language, e.g. English and French. All such elements have been made repeatable, and the new *i18n* attribute group can be used to identify the language of each instance of the repeated element.
- It is now possible to convey the `MediaType` for any type of ordered ad, not just `Placement.Generic` where it was previously available.
    - As a result of this change, the element `MediaType` has moved from its previous location in `Placement.Generic` and `BookingInformation.Generic`, which is a backwards-incompatible change.
- `ProductionDetail.Generic` has been significantly enhanced.
- A `CampaignDescription` structure has been added to the top level of the message. This can convey a description of the advertiser's campaign such as its start and end dates, goals, description and constraints.
- The `ProofOfPublication` structure has been expanded and made media-agnostic so that it fully supports interactive media scenarios, for example, by identifying the party that will certify the advertisement's performance.

- The deprecated element `AdditionalServices` has been removed. (It had been replaced in Bookings 2.0 with the more expressive structure `adsml:AdditionalService`.)
- It is now possible to convey user defined properties at many more locations within an AdsMLBookings message than previously. This allows for the transmission of more context-specific properties.

1.10 Acknowledgments

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

Primary authorship and editing was performed by,

- Tony Stewart (RivCom) – tony.stewart@rivcom.com
- Ulf Wingstedt (CNet Svenska AB) ulf.wingstedt@cnet.se

Additional editing was performed by,

- Howard Robson (5 fifteen Solutions, Inc)
- Matthew Leete (Associated Newspapers Ltd)

Acknowledgments and thanks to other contributors for additional input to this document are listed in Appendix A: Acknowledgment 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 AdsMLBookings standard is developed by the AdsML Consortium to be a global standard for the exchange of advertising bookings. It relies on earlier experience and standards that have been embraced and extended in order to support current advertising business requirements. In addition, AdsMLBookings 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.

The current release of AdsMLBookings fully supports bookings for display, classified and insert advertisements in print publications (including both newspapers and magazines) and for interactive online media, and contains initial, lightweight support for bookings in any other media. The standard is designed to handle the full lifecycle of a booking, from quotations through firm bookings, including any changes or cancellations along the way.

AdsMLBookings is flexible and expressive and able to support complex package bookings of multiple placements in different media, as well as simple single placement, single date type bookings. The standard is also designed to support the later addition of other media types, including broadcast and outdoor advertising.

An AdsMLBookings message consists of three levels:

1) Booking level - A media agnostic top level that includes basic information about involved organizations such as buyer and seller, unique identifiers of the booking as well as a total price.

2) Placement Group level - An optional mid level supporting bookings with “packages of packages” including a sub package price.

3) Placement level - A media specific bottom level where all details about the actual advertisement (placement) is given, in XML structures that have been specifically designed to support the requirements for different media types. Price information can be recorded also at this level.

AdsMLBookings message exchanges are expected to normally be performed in Request/Response pairs, i.e. an advertisement booking is requested by a buyer from a seller of advertising who responds with a confirmation or denial message. But the standard also handles other models such as sending a formal XML quotation based on a request that came in as a phone call. At the other extreme, the message choreography can support sophisticated exchange patterns that allow senders and receivers to ensure that messages are received in the expected sequence.

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 AdsMLBookings standard has been developed by the AdsML Consortium as the preferred approach to handle ad bookings and packages of ad bookings for multiple media.

For AdsMLBookings, the AdsML Framework provides a messaging infrastructure for delivery of booking messages.

An important issue in enabling automatic business message flows is the use of common well-defined message choreography. One of the main components in the AdsML Framework is a set of business process models and related documentation that includes a definition of common process models for the workflows of selected advertising classes (*AdsML Advertising Component Interactions Analysis).* All business messages from the ad bookings group are supported in AdsMLBookings.

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 common technical components and a common approach to advertising e-commerce that makes them "AdsML".

2.2 Use of the AdsML Envelope is optional, but recommended

AdsMLBookings 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 AdsMLBookings does not require use of, nor support for, the AdsML Envelope standard. The actual transfer of AdsMLBookings messages can be performed by arbitrary method and software application, with or without the use of the AdsML Envelope. For instance, an AdsMLBookings 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 AdsMLBookings 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.

2.3 Relationship to other advertising standards

AdsMLBookings focuses on the ad booking processes, and is intended to be used in conjunction with other XML vocabularies covering other areas in the advertising work flow.

- AdsML Structured Descriptions. An XML standard developed by the AdsML Consortium for expressing structured descriptions of objects appearing in an advertisement.
- JDF. Developed by CIP4™[1], the Job Definition Format (JDF) is an XML-based job ticket format used to create end-to-end job tickets for a print run. JDF facilitates information exchange and facilitates integration and interoperability among workflow systems.

  AdsMLBookings slightly overlaps with JDF in that it includes some information about the print publishing process. However, the rationale behind this information in the booking is to be able to calculate a price as well as prepare the publishing process (e.g. reserve space, use of colors) and not to perform actual publishing. JDF includes much more detail and it is assumed that the bookings publishing data will eventually be an input to a JDF controlled publishing process.

Within the booking area, AdsMLBookings is designed to extend and embrace functionality previously covered, partly, by older standards. The following targets have principally been targeted:

- CREST. Developed by the Classified Advertising Standards Task Force of NAA[2], CREST 2.0 is an XML-based media independent format for electronically exchanging and sharing classified advertising data. CREST focuses on the three main areas of classified advertising - real estate, transportation, and

[1] CIP4™ is an acronym for 'Cooperation for the Integration of Processes in Prepress, Press and Postpress'. (http://www.cip4.org/)

[2] NAA is an acronym for 'Newspaper Association of America'. (http://www.naa.org)

employment categories, and provides a generic extension mechanism to record advertising data that falls outside these categories. CREST 2.0 supersedes the earlier '*CREST® NAA Guidelines for Classified Advertising Remote Markup and Transmission, Version 1.0, May 1995*'.

CREST includes parts that cover bookings of classified ads.

- IfraAdConnexion. Developed by Ifra[3], IfraAdConnexion is an XML-based vocabulary for the newspaper industry, the vocabulary concentrating on the ad booking and ordering processes. IfraAdConnexion supersedes the earlier 'Edifra' EDI messages for advertising.

  IfraAdConnexion version 2 is a proper subset of the AdsMLBookings and it is possible to translate any IfraAdConnexion message into an equivalent AdsMLBookings message. The reverse is also possible, provided that the appropriate subset of AdsMLBookings is used.

- SPACE/XML. Developed by IDEAlliance[4] and NAA, the XML-based Specification for Publisher and Agency Communication Exchange (SPACE/XML) is a standard for sending and acknowledging advertising space reservations, insertion and change orders, invoicing, and advertising copy data files between advertising agencies, prepress or prepress media services, printers, and publishers. SPACE/XML is an XML version of the earlier SPACE/X12 EDI transactions.

[3] Ifra (http://www.ifra.com/) is an acronym for 'INCA-FIEJ Research Association'. "INCA" stands for "International Newspaper Colour Association". "FIEJ" stands for "Fédération Internationale des Editeurs de Journaux".

[4] The committee developing the standard included representatives from International Digital Enterprise Alliance, (IDEAlliance) (http://www.idealliance.org), Newspaper Association of America (NAA), Digital Distribution of Advertising for Publications (DDAP) Association, and Magazine Publishers of America (MPA).

3 Business Messages Overview

3.1 Supported messages

AdsMLBookings supports the business process model and message flow as proposed in the *AdsML Advertising Component Interactions Analysis*, a part of the AdsML Framework. In particular, AdsMLBookings defines the set of business messages that belongs to the ad booking group (AD).

There are three main classes of business messages defined in the schema, specifying bookings in different states:

- Quotations
- Reservations (preliminary bookings)
- Orders (confirmed bookings)

A reservation or an order might be based on a previous quotation. A reservation might be turned into an order.

In all three of these message classes, there is a request-response pair of transactions. For instance, a customer can request a quotation and a seller can respond with a quotation. For reservations and orders, there are also messages for changing or canceling an existing booking. A change or a cancellation message might be issued by either the buyer or the seller.

Finally, all three message classes have status request and response messages defined. A status request might be issued by the buyer that would like to know the current state of the booking in the seller's system. A status message might also be sent spontaneously by the seller to inform either the buyer or other relevant parties about the status of a particular booking.

The complete list of supported business messages is:

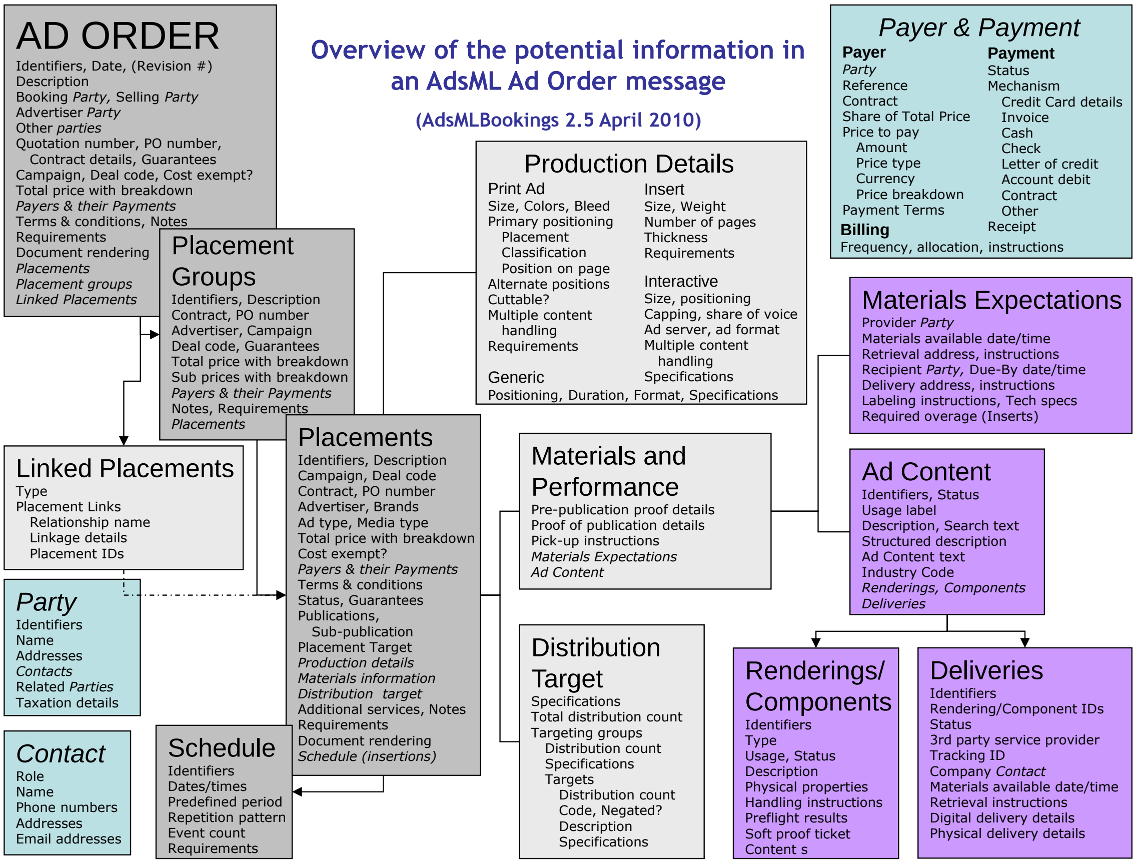
Message Code	Message Name
AD-RFQ	Ad Quotation Request
AD-Q	Ad Quotation
AD-QSE	Ad Quotation Status Enquiry
AD-QS	Ad Quotation Status
AD-R	Ad Reservation
AD-RC	Ad Reservation Change
AD-RX	Ad Reservation Cancellation
AD-RR	Ad Reservation Response
AD-RSE	Ad Reservation Status Enquiry
AD-RS	Ad Reservation Status
AD-O	Ad Order
AD-OC	Ad Order Change

Message Code	Message Name
AD-OX	Ad Order Cancellation
AD-OR	Ad Order Response
AD-OSE	Ad Order Status Enquiry
AD-OS	Ad Order Status

3.2 Data overview

The figure below provides a diagrammatic overview of the potential information in a Bookings request message describing a new Quotation, Reservation or Order.

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



Notes:

- The Revision identifier is not available in an initial Reservation or Order. It becomes available in Change and Cancellation messages.
- Change messages also include a Change Specification structure to describe the nature of the changes.

3.3 Content model

For each business message type supported by AdsMLBookings, there is a corresponding content model in AdsMLBookings.

Different messages have different content models. However, there are also many similarities and common content models are reused in several message types.

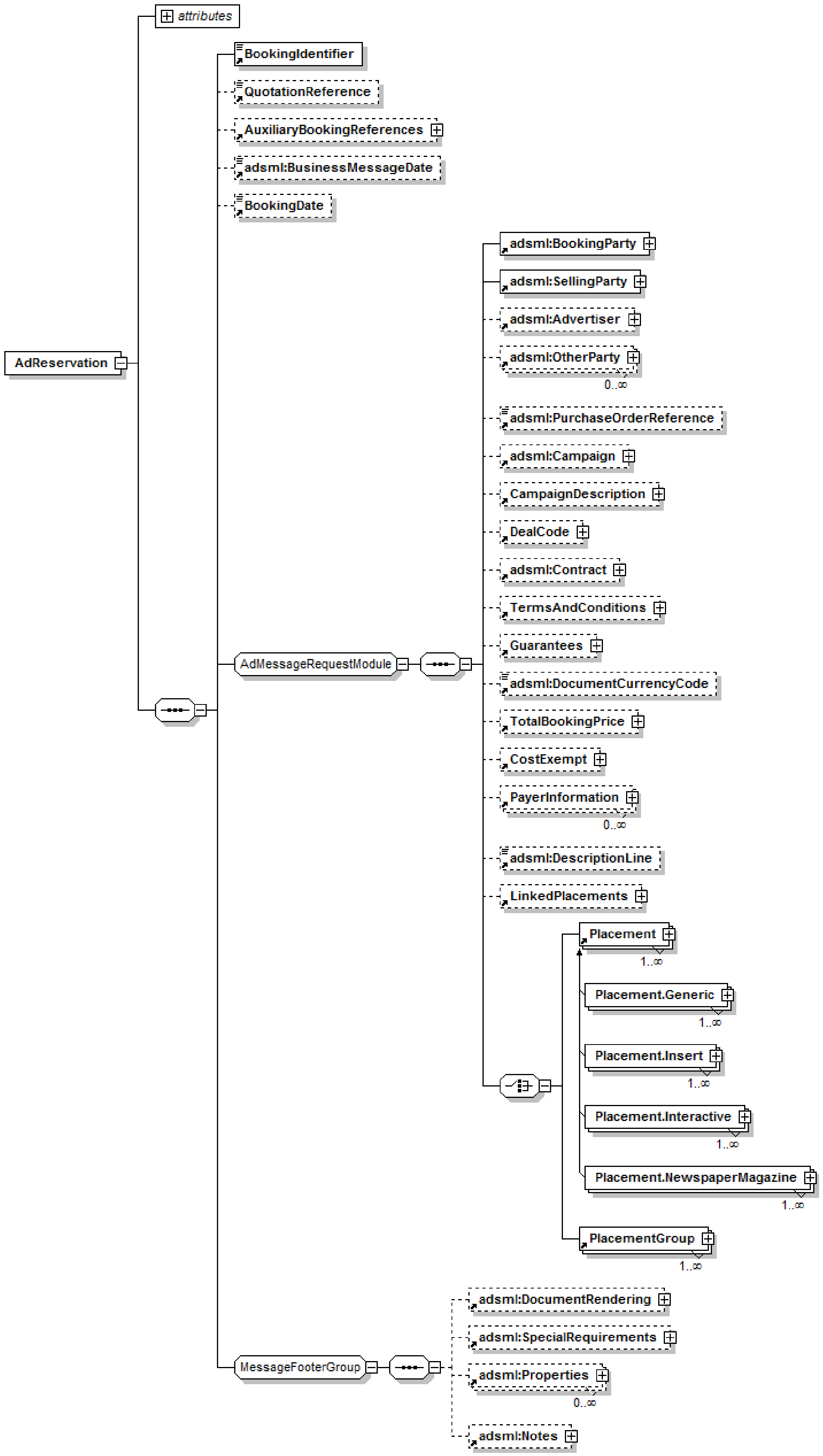
The figure below shows the content model for the AdReservation message type. It is very similar to, for example, the AdOrder or AdReservationResponse.

A new reservation must include a reference key (the BookingIdentifier). The general rule for reference keys is that the originating party provides a reference key with the message and that this key is then used in the response and any further transaction regarding this reservation.

In addition to the reference key for the reservation, the buyer might also include a reference to a previous quotation.

All booking messages can include one or more placements of advertising. This approach provides the possibility of transferring information about booking packages, where each component in the package can be separately defined with respect to media, ad content descriptions, insertion dates etc. Moreover, each component in the package (or Placement as it is called here) can be separately priced while the package as a whole also has a price. This allows specification of package "deals" where the price of a complete booking package might be less than the sum of its components.

A booking can in addition include a PlacementGroup with a separate pricing level, allowing "packages of packages" to be defined.


attributes
BookingIdentifier
QuotationReference
AuxiliaryBookingReferences
adsml:BusinessMessageDate
BookingDate
AdReservation
adsml:BookingParty
adsml:SellingParty
adsml:Advertiser
adsml:OtherParty
0..∞
adsml:PurchaseOrderReference
adsml:Campaign
CampaignDescription
DealCode
adsml:Contract
TermsAndConditions
Guarantees
AdMessageRequestModule
adsml:DocumentCurrencyCode
TotalBookingPrice
CostExempt
PayerInformation
0..∞
adsml:DescriptionLine
LinkedPlacements
Placement
1..∞
Placement.Generic
1..∞
Placement.Insert
1..∞
Placement.Interactive
1..∞
Placement.NewspaperMagazine
1..∞
PlacementGroup
1..∞
adsml:DocumentRendering
adsml:SpecialRequirements
MessageFooterGroup
adsml:Properties
0..∞
adsml:Notes

3.3.1 Media-specific booking information

A booking message is media agnostic in that the top (package) level is identical regardless of whether the ad will run in a newspaper or broadcast media. All media specific information is instead pushed down into the components of the package, in the `Placement`. AdsMLBookings provides specific content models targeted for specifying ad placements in particular media where media specific differences have been accounted for, as well as a generic placement structure for media not particularly covered.

At the time of this writing, the following media-specific content models are available:

- **.NewspaperMagazine** (`Placement.NewspaperMagazine, ProductionDetail.NewspaperMagazine, InsertionPeriod`): used for an advertisement that will be printed on one or more pages of a newspaper, magazine or similar publication. Includes all varieties of such advertisements: Display, Classified, Classified Display, etc.
- **.Insert** (`Placement.Insert, ProductionDetail.Insert, PickUp.Insert, MaterialsExpectations.Insert`): used for pre-printed material which is inserted into, wrapped around, attached to or bound into a printed publication. It can also be used for samples of goods which are included within a publication.
- **.Interactive** (`Placement.Interactive, ProductionDetail.Interactive`): used for an advertisement that will be distributed by an electronic medium which allows for (but does not necessarily require) user interaction with the advertisement. This includes, in particular, online advertisements, and it is also well suited for digital signage. However, the .Interactive content model is not optimized for ordering search engine ads, or similar situations where the price and positioning of the ad are determined in an auction process.
- **.Generic** (`Placement.Generic, ProductionDetail.Generic`): used for all other media, as well as for any products or services that are not directly related to a specific publication or broadcast (e.g. sponsorships).

Information regarding the publication of an advertisement in a medium for which a media-specific content model is provided by AdsMLBookings **MUST** be conveyed using the appropriate media-specific content model.

Information regarding the publication of an advertisement in a medium for which no media-specific content model is provided by AdsMLBookings **SHOULD** be conveyed using the `.Generic` content model.

Information regarding the provision of a product or service that is not directly related to the publication of an advertisement (e.g. sponsorships) **SHOULD** be conveyed using the `.Generic` content model.

4 Message Choreography

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

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

AdsMLBookings includes two main categories of messages:

- Business messages, i.e. messages such as new bookings, changes, cancellations and status requests that are part of the parties' advertising business.
- Administrative messages, i.e. house-keeping messages for the systems involved in exchange of business messages. Examples are error messages and receipts of received AdsMLBookings 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 AdsMLBookings messages. For a description of administrative messages and error handling, please see *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 AdsMLBookings messages. For a description of test messages, please see *AdsML E-commerce Usage Rules & Guidelines*.

4.3 Response Modes

The preferred messaging model is the Request-Response model as specified in the *AdsML E-commerce Usage Rules & Guidelines*.

However, since legacy applications may have limited ability to provide appropriate responses, it is also possible to use a model where only requests and administrative responses are transmitted, assuming an acceptance on the receiver's side. If a problem occurs when a booking message cannot be accepted, it has to be solved manually. This kind of model is called a datagram model. For more information about datagram messaging, see the *AdsML E-commerce Usage Rules & Guidelines.*

As a summary:

1) Implementations of AdsMLBookings **SHOULD** apply the full Request-Response model
2) If agreed by communication parties, implementations **MAY** use a datagram model (no business level responses required), and if so, they must agree on which direction(s) of datagram messaging they will support.

4.4 Business Messages

Each business message type is identified by a message code that specifies if the message is, for instance, an ad order, an ad order change, a cancellation, or responses to these messages. AdsMLBookings supports a subset of business messages as defined in *AdsML Advertising Component Interactions Analysis*, namely messages from the ad bookings group (AD).

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

The sections below give a summary of the messages in each subgroup “Quotations” “Reservations” and “Orders”. For more information, see also the reference section for each message element (named as the message name in CamelCase).

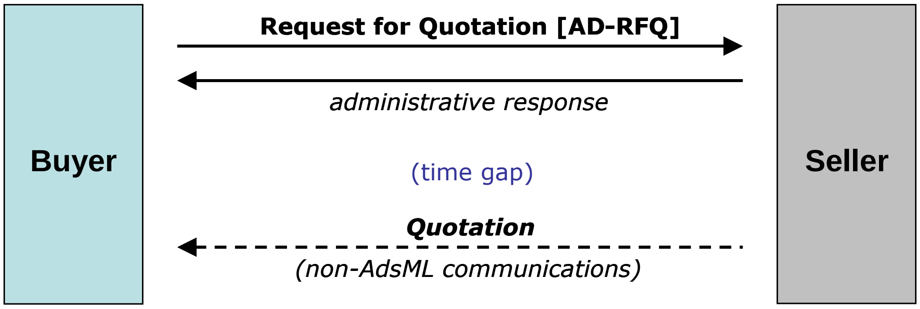
4.4.1 Ad Quotation Messages

A Quotation is requested by the Buyer and provided by the Seller. However, provision of the quotation is merely an offer; it does not constitute a business agreement between buyer and seller.

AdsML does not provide a Quotation message by which the buyer “accepts” a quotation. Such acceptance is indicated when the buyer later sends a Reservation or Order message which references the seller’s previous quotation.

4.4.1.1 Datagram messaging from buyer to seller

1) The buyer requests a quotation by sending an Ad Quotation Request message (AD-RFQ), after which the seller provides the requested quotation information by non-AdsML means (i.e. phone or fax). This ends the message transaction.



Ad quotation datagram message from buyer to seller

4.4.1.2 Datagram messaging from seller to buyer

1) The buyer requests a quotation by a non-AdsML means such as a fax or phone call. The seller sends an Ad Quotation (AD-Q) message in response to that request. This ends the primary message transaction.

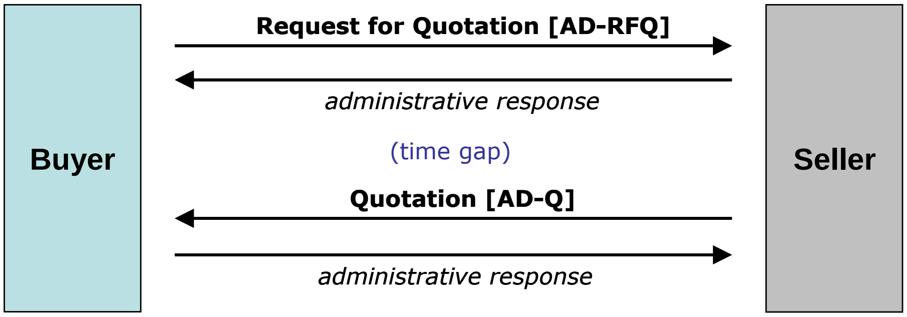
2) A seller **MAY** send an Ad Quotation Status (AD-QS) message spontaneously at any time during the quotation's lifecycle.

4.4.1.3 Request-Response messaging model

1) An Ad Quotation Request (AD-RFQ) can be issued by a buyer and **MUST** result in an Ad Quotation (AD-Q) that either confirms or denies the request.

2) If an Ad Quotation is a response to an Ad Quotation Request, it **MUST** reference the Ad Quotation Request's message identifier and quotation identifier.

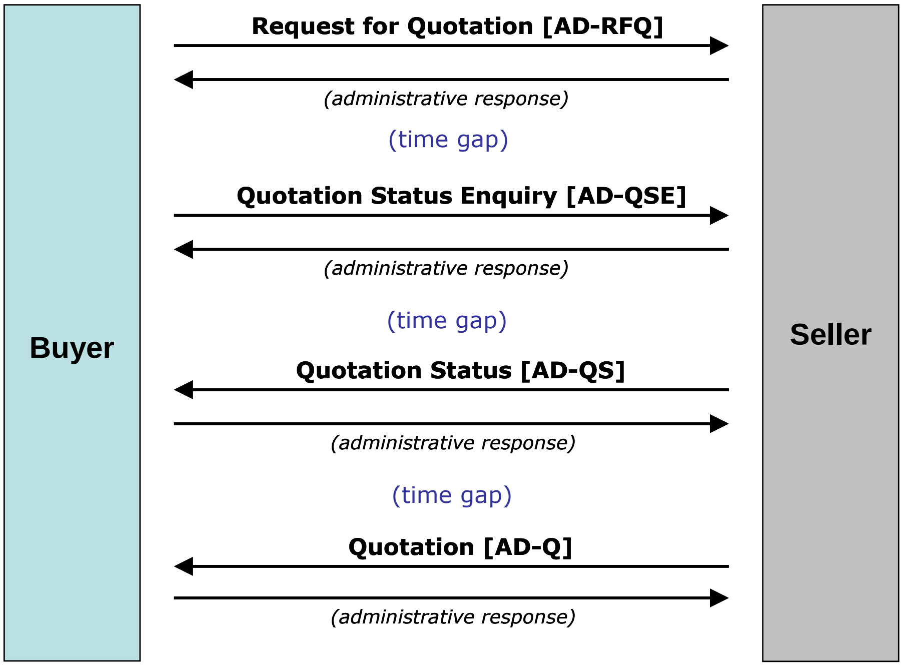
3) If a quotation cannot be given, an Ad Quotation message **MUST** anyway be issued by the seller that explains the reason for not accepting the request using the `adsml:RequestDenied` element.



Ad Quotation request-response messages

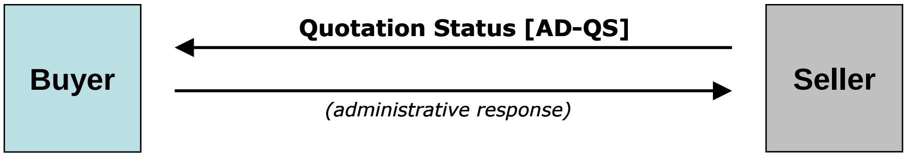
4) An Ad Quotation Status Enquiry (AD-QSE) **MAY** be issued by the buyer while waiting for the Ad Quotation (AD-Q) to arrive, but **SHOULD NOT** be issued after it has arrived. An Ad Quotation Status Enquiry **SHOULD** result in an Ad Quotation Status (AD-QS) message from the seller that provides information about the current status of the quotation.

5) If an Ad Quotation Status message is a response to an Ad Quotation Status Enquiry, it **MUST** reference the Ad Quotation Status Enquiry's message identifier and quotation identifier.


Buyer
Seller
Request for Quotation [AD-RFQ]
(administrative response)
(time gap)
Quotation Status Enquiry [AD-QSE]
(administrative response)
(time gap)
Quotation Status [AD-QS]
(administrative response)
(time gap)
Quotation [AD-Q]
(administrative response)

Ad quotation status enquiry and response in between a quotation request and response

6) An Ad Quotation Status (AD-QS) **MAY** be issued by a seller without having received an Ad Quotation Status Enquiry (AD-QSE).



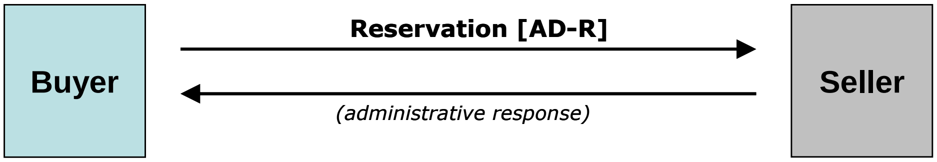
Ad quotation status message spontaneously sent by the seller.

4.4.2 Ad Reservation Messages

A Reservation is requested by the Buyer and confirmed or denied by the Seller. Acceptance of a Reservation constitutes a business agreement between buyer and seller according to the terms of their TPA.

4.4.2.1 Datagram messaging from buyer to seller

1) The buyer sends an Ad Reservation (AD-R) message to the Seller. Once the buyer has received an Administrative Response from the seller (indicating that the message was received), the seller is assumed to have accepted the reservation.



Ad reservation datagram messaging. This pattern is also used for changes and cancellations.

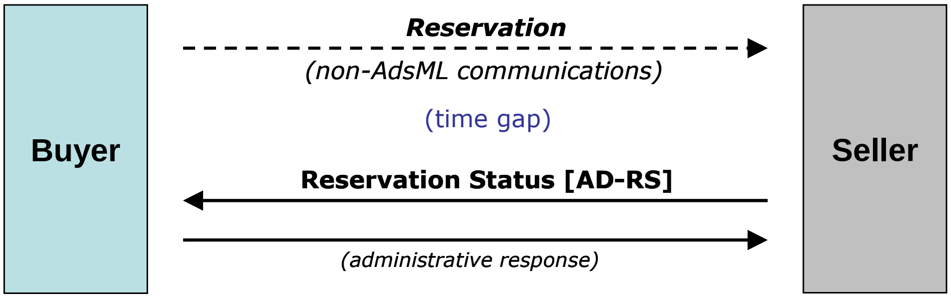
2) To change a previously transmitted reservation, the buyer sends an Ad Reservation Change (AD-RC) message to the seller. Once the buyer has received an Administrative Response from the seller (indicating that the message was received), the seller is assumed to have accepted the changes to the reservation.

3) To cancel a previously transmitted reservation, the buyer sends an Ad Reservation Cancel (AD-RX) message to the seller. Once the buyer has received an Administrative Response from the seller (indicating that the message was received), the seller is assumed to have cancelled the reservation.

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

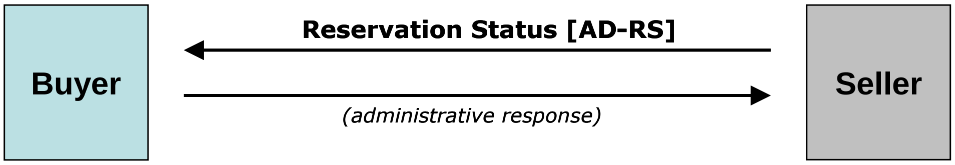
4.4.2.2 Datagram messaging from seller to buyer

1) The buyer initiates, changes or cancels a reservation by a non-AdsML means such as a fax or phone call. The seller then sends an Ad Reservation Status (AD-RS) message which confirms the current status of the reservation. This ends the primary message transaction.


Buyer
Reservation
(non-AdsML communications)
(time gap)
Reservation Status [AD-RS]
(administrative response)
Seller

Reservation request using datagram messaging from seller to buyer. This pattern is also used for changes and cancellations.

2) A seller **MAY** send an Ad Reservation Status (AD-RS) message spontaneously at any time during the reservation's lifecycle.



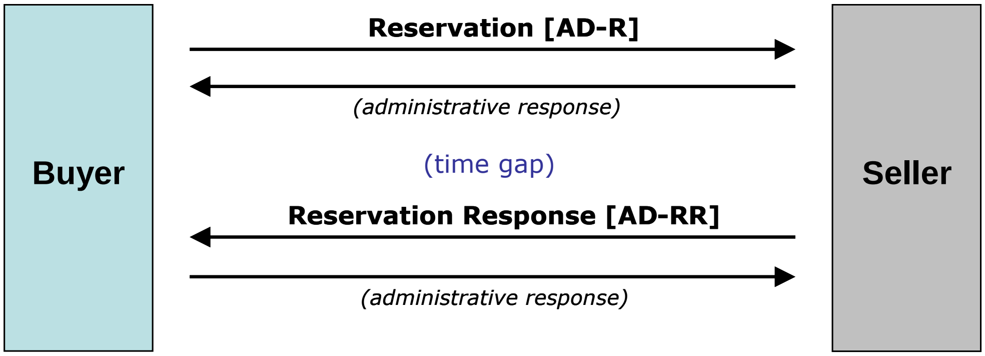
Reservation status message

4.4.2.3 Request-Response messaging model

1) An Ad Reservation (AD-R) **MUST** result in an Ad Reservation Response (AD-RR) that either confirms or denies the reservation.

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

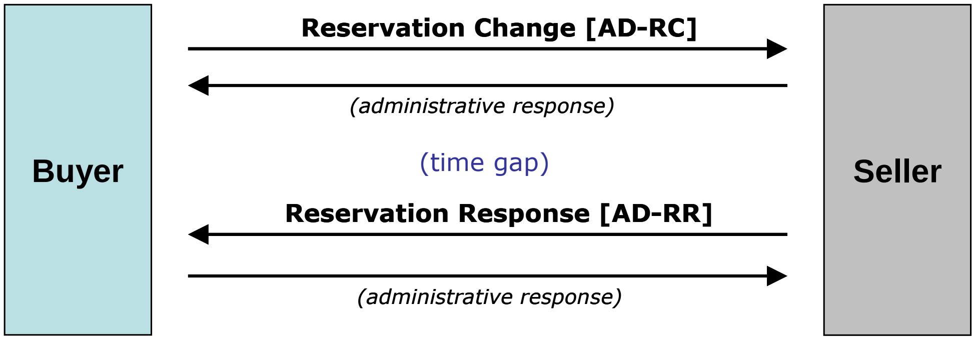
3) If a reservation is confirmed, the response message **MUST** contain a adsml:NatureOfResponse code indicating acceptance and **MUST** echo back to the buyer the details of the accepted reservation as it exists in the seller's system (to the level of detail agreed by the parties in their TPA).


Buyer
Reservation [AD-R]
(administrative response)
(time gap)
Reservation Response [AD-RR]
(administrative response)
Seller

Ad reservation request-response, including administrative responses.

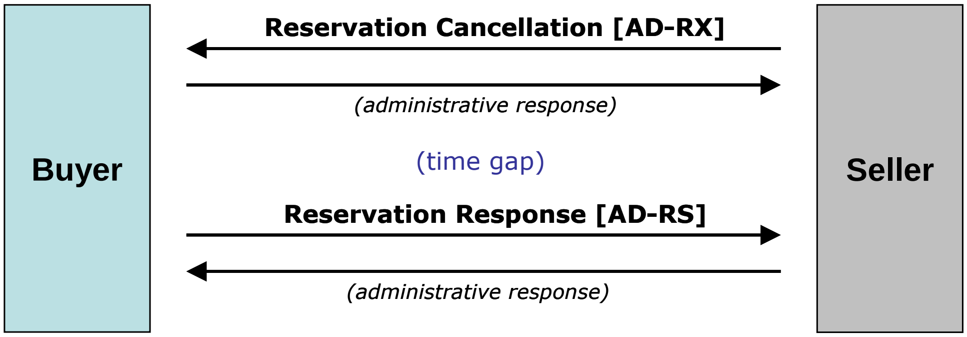
4) A reservation can be changed by either the buyer or the seller using the Ad Reservation Change (AD-RC) message.

5) All change requests **MUST** be fully specified reservations and must completely replace a previous reservation. Partial updates, of e.g. only an insertion date, are not allowed, and if received by the seller **MUST** trigger a "Request Denied" response message.


Buyer
Reservation Change [AD-RC]
(administrative response)
(time gap)
Reservation Response [AD-RR]
(administrative response)
Seller

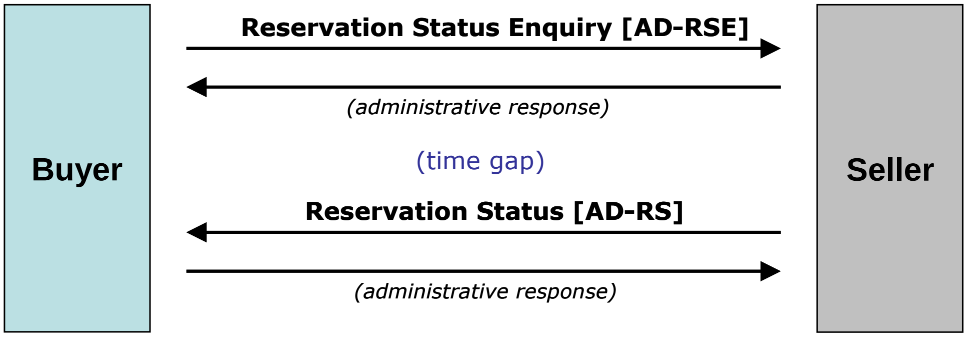
Buyer-initiated reservation change, followed by the seller's response.

6) Either buyer or seller can cancel an entire reservation using the Ad Reservation Cancellation (AD-RX) message. The receiver of the cancellation message **MUST** confirm with an Ad Reservation Response (AD-RS). (Note: cancellation of just part of a Reservation, for example a single Placement, is done using a Change message rather than a Cancellation message.)


Buyer
Reservation Cancellation [AD-RX]
(administrative response)
(time gap)
Reservation Response [AD-RS]
(administrative response)
Seller

Seller-initiated cancellation, followed by the buyer's response. (This message exchange can also be initiated by the buyer.)

7) An Ad Reservation Status Enquiry (AD-RSE) **MUST** result in an Ad Reservation Status message response.
8) If an Ad Reservation Status message is a response to an Ad Reservation Status Enquiry, it **MUST** reference the Ad Reservation Status Enquiry's message identifier and reservation identifier.


Buyer
Reservation Status Enquiry [AD-RSE]
(administrative response)
(time gap)
Reservation Status [AD-RS]
(administrative response)
Seller

Reservation status enquiry and response.

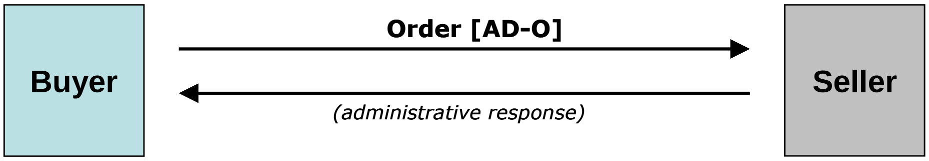
9) An Ad Reservation Status (AD-RS) **MAY** be issued by the seller without a previous Ad Reservation Status Enquiry (AD-RSE) having been received.

4.4.3 Ad Order Messages

An Order is requested by the buyer and confirmed or denied by the seller. Acceptance of an Order constitutes a business agreement between buyer and seller according to the terms of their TPA.

4.4.3.1 Datagram messaging from buyer to seller

1) The buyer sends an Ad Order (AD-O) message to the seller. Once the buyer has received an Administrative Response from the seller (indicating that the message was received), the seller is assumed to have accepted the order.



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

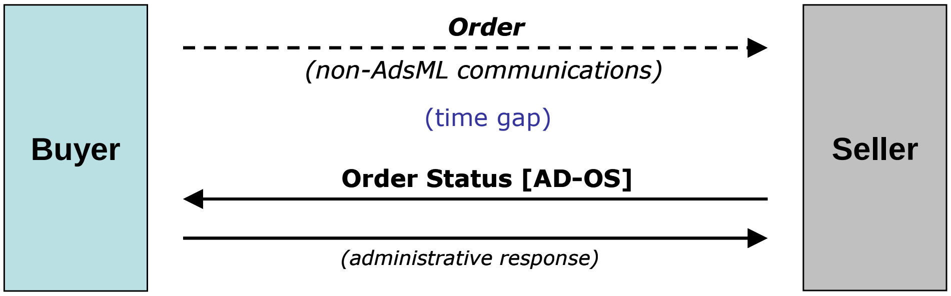
2) To change a previously transmitted order, the buyer sends an Ad Order Change (AD-OC) message to the seller. Once the buyer has received an Administrative Response from the seller (indicating that the message was received), the seller is assumed to have accepted the changes to the order.

3) To cancel a previously transmitted order, the buyer sends an Ad Order Cancel (AD-OX) message to the seller. Once the buyer has received an Administrative Response from the seller (indicating that the message was received), the seller is assumed to have cancelled the order.

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

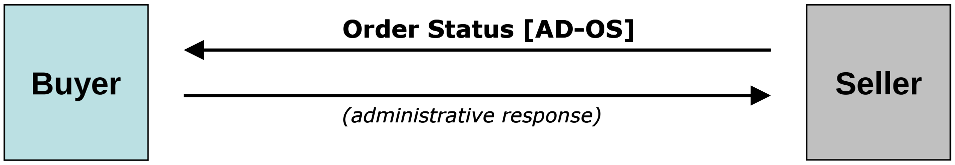
4.4.3.2 Datagram messaging from seller to buyer

1) The buyer initiates, changes or cancels an order by a non-AdsML means such as a fax or phone call. The seller then sends an Ad Order Status (AD-OS) message which confirms the current status of the order. This ends the primary message transaction.



Order request using datagram messaging from seller to buyer. This pattern is also used for changes and cancellations.

2) A seller **MAY** send an Ad Order Status message spontaneously at any time during the order's lifecycle.



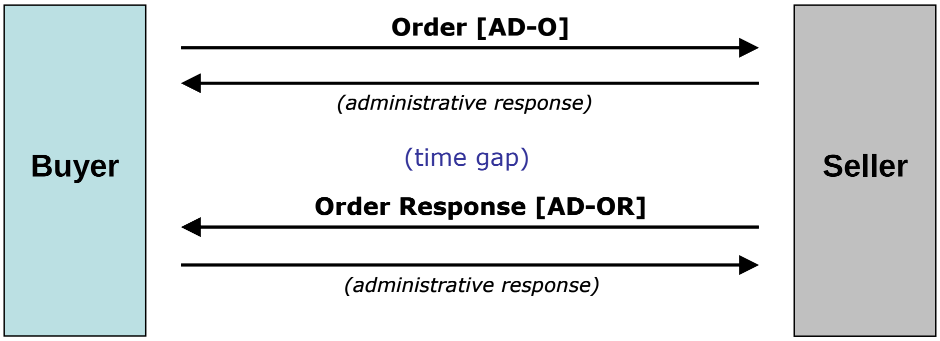
Order status message

4.4.3.3 Request-Response messaging model

1) An Ad Order (AD-O) **MUST** result in an Ad Order Response (AD-OR) 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.

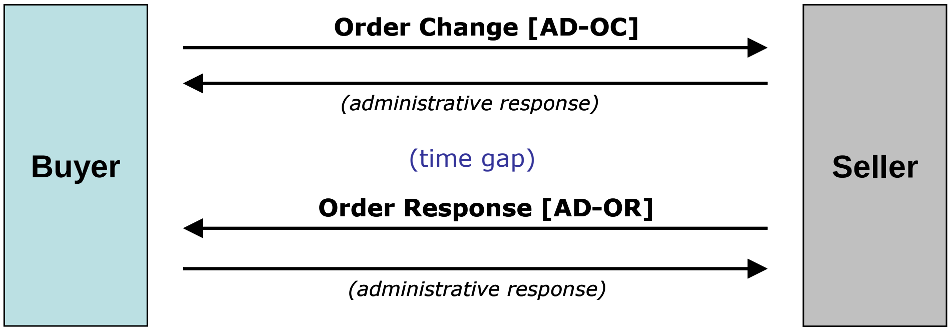
3) If an order is confirmed, the response message **MUST** contain a `adsml:NatureOfResponse` code indicating acceptance and **MUST** echo back to the buyer the details of the accepted order as it exists in the seller's system (to the level of detail agreed by the parties in their TPA).



Ad order request-response, including administrative responses.

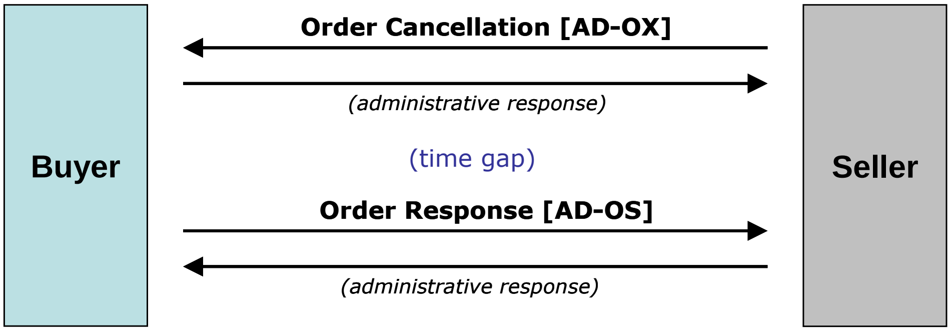
4) An order can be changed by either the buyer or the seller using the Ad Order Change (AD-OC) message.

5) All change requests **MUST** be fully specified orders and must completely replace a previous order. Partial updates, of e.g. only an insertion date, are not allowed, and if received by the seller **MUST** trigger a "Request Denied" response message.



Buyer-initiated order change, followed by the seller's response.

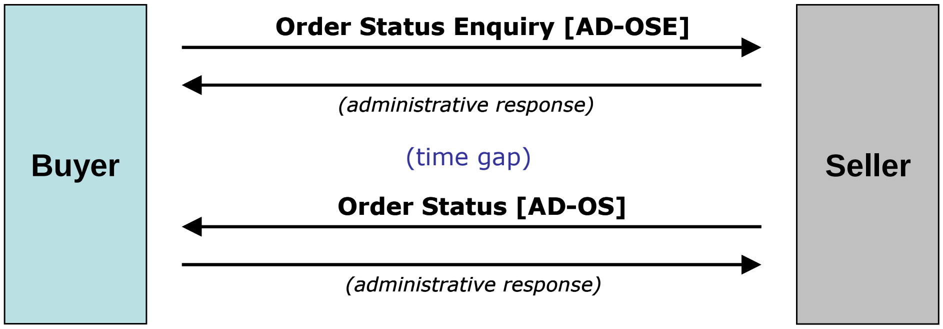
6) Either buyer or seller can cancel an entire order using the Ad Order Cancellation (AD-OX) message. The receiver of the cancellation message **MUST** respond with an Ad Order Response (AD-OR). (Note: cancellation of just part of an order, for example a single Placement, is done using a Change message rather than a Cancellation message.)


Buyer
Order Cancellation [AD-OX]
(administrative response)
(time gap)
Order Response [AD-OS]
(administrative response)
Seller

Seller-initiated cancellation, followed by the buyer's response. (This message exchange can also be be initiated by the buyer.)

7) An Ad Order Status Enquiry (AD-OSE) **MUST** result in an Ad Order Status (AD-OS) message.

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


Buyer
Order Status Enquiry [AD-OSE]
(administrative response)
(time gap)
Order Status [AD-OS]
(administrative response)
Seller

Order status enquiry and response.

9) An Ad Order Status **MAY** be issued by the seller without a previous Ad Order Status Enquiry having been received.

4.4.4 Message References – Booking and Quotation Identifiers

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

Both ad reservation (AD-R) and ad order (AD-O) messages normally initiate a **booking** in the seller's booking system. During the life cycle of a booking, a reservation might change and later be confirmed as an order that, in turn, can be changed. In order to maintain the relationship between request-response message pairs, as well as between a series of change messages regarding the same "booking", a stable identifier **MUST** be used during the complete suite of possible messages, i.e. the *booking identifier* below. Change and Status messages (AD-RC, AD-OC, AD-RS, AD-OS, AD-RSE, AD-OSE) **MUST** use the same identifier as was used in the initiating message.

An implication of the above is that a buying or booking system **MUST** be able to store this identifier with the booking in its internal data storage.

In general, each message has the following identifiers:

- *The booking identifier*. An identifier first issued by the party that initiates the reservation or ordering process. The booking identifier is the primary identifier for a Reservation or Order and **MUST NOT** change during the life of that Reservation or Order. Its structure **MUST** conform to the AdsML QID format, and it **MUST** be included in any reservation or order message transmitted, both responses and requests. The booking identifier is called `BookingIdentifier`.
- *The buyer's reference*. A reference identifier issued by the buying party in a quotation, reservation or ordering transaction. The buyer's reference conveys the internal identifier that the buyer uses to reference this Quotation, Reservation or Order. The buyer's reference is optional; however, once provided, its value **MUST** remain stable throughout the life of the Quotation, Reservation or Order. The buyer's reference is called `adsml:BuyersReference`.
- *The seller's reference*. A reference identifier issued by the selling party in a quotation, reservation or ordering transaction. The seller's reference conveys the internal identifier that the seller uses to reference this Quotation, Reservation or Order. The seller's reference is optional; however, once provided, its value **MUST** remain stable throughout the life of the Quotation, Reservation or Order. The seller's reference is called `adsml:SellersReference`.
- *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`*.

Response messages need to identify the message they respond to as well as the message class of that message:

- *The "in response to" message ID*. A mandatory ID referencing the message that a response is about.

As a set of multiple updates, say, may be issued before a response is received for the first message, it must be possible to distinguish responses from one another. The ID appears in responses as the attribute *`inResponseToMessageID`*. Update and cancellation messages also include:

- *The last received message ID.* All update and cancellation messages **MAY** include the message ID of the last message received about the particular booking.

This ID can be used to detect "dirty updates", i.e. since both seller and buyer can issue updates there is a risk that both do so simultaneously. When receiving an update request, it is possible to check that the value of the last received message ID in the update request corresponds to the last message sent for the particular booking.

Quotation messages (AD-RFQ, AD-Q, AD-QSE, AD-QS) use a mandatory identifier equivalent to the BookingIdentifier, but named `QuotationIdentifier`:

- *The quotation identifier*. An identifier first issued by the party that initiates a quotation process. The quotation identifier is the primary identifier for a Quotation and **MUST NOT** change during the life of that Quotation. Its structure

MUST conform to the AdsML QID format, and it **MUST** be included in any Quotation message transmitted, both responses and requests. A reference to this value can also be conveyed in Reservation or Order messages that relate to the quotation. The quotation identifier is called `QuotationIdentifier`.

Compared to Orders and Reservations, Quotations have a shorter life cycle with a single pair of request/response. Each new request for quotation **MUST** use a new `QuotationIdentifier`. The quotation identifier **SHOULD** be stored by the booking system.

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

4.5 Sample Choreography Use Cases

These technical use cases illustrate the relationship between message flows and the ID values that are used to synchronize the messages.

4.5.1 Ad order and later update

A new order followed by an update in due time.

	Buyer	Seller
1	-> AdOrder messageID=”X1” BookingIdentifier=”1”	
2		-> AdOrder messageID=”X1” BookingIdentifier=”1”
3		Create new booking in the booking system based on the order.
4		<- AdOrderResponse messageID=”Y1” BookingIdentifier=”1” inResponseToMessageID=”X1” status=”OK”
5	<- AdOrderResponse messageID=”Y1” BookingIdentifier =”1” inResponseToMessageID=”X1” status=”OK”	
6	Check that the response is correct compared to order.	
7	OK – transaction is complete.	
8	The order must be changed. Create a change message that includes the message ID of the last received message.	

	Buyer	Seller

9	-> AdOrderChange
	messageID=”X2” BookingIdentifier =”1”
	lastReceivedMessageID=”Y1”	

10		-> AdOrderChange messageID=”X2” BookingIdentifier =”1” lastReceivedMessageID=”Y1”
11		Verify that the change is based on the last message sent by the seller. It is. The change is acknowledged, send response.
12		<- AdOrderResponse messageID=”Y2” BookingIdentifier =”1” inResponseToMessageID=”X2” status=”OK”
13	<- AdOrderResponse messageID=”Y2” BookingIdentifier =”1” inResponseToMessageID=”X2” status=”OK”	
14	Check that the response is correct compared to change order.	
15	OK – Transaction is complete.	

4.5.2 Ad order and change before response

A new order followed by an update from the buyer before the seller’s response to the new order was received by the buyer.

	Buyer	Seller
1	-> AdOrder messageID=”X1” BookingIdentifier =”1”	
2		-> AdOrder messageID=”X1” BookingIdentifier =”1”
3		Create new booking in the booking system based on the order.
4		The booking requires manual handling and cannot be accepted immediately.
5	The order must be changed. Create a change message, but it cannot include the message ID of the last received message since no response message has been received yet.	

	Buyer	Seller
6	-> AdOrderChange messageID=”X2” BookingIdentifier =”1”	
7		-> AdOrderChange messageID=”X2” BookingIdentifier =”1”
8		The new order and order change are handled and both are acknowledged with Administrative Response messages. A single business-level response can be issued with the resulting booking (after applying new order and change order)
9		<- AdOrderResponse messageID=”Y1” BookingIdentifier =”1” inResponseToMessageID=”X2” status=”OK”
10	<- AdOrderResponse messageID=”Y1” BookingIdentifier =”1” inResponseToMessageID=”X2” status=”OK”	
11	Check that the response is correct compared to order plus changed order.	
12	OK – transaction complete.	

4.5.3 Ad order and simultaneous updates

A new order is sent and accepted, after which both buyer and seller issue simultaneous changes. The conflict is detected, and both changes are denied.

	Buyer	Seller
1	-> AdOrder messageID=”X1” BookingIdentifier =”1”	
2		-> AdOrder messageID=”X1” BookingIdentifier =”1”
3		Create new booking in the booking system based on the order.
4		<- AdOrderResponse messageID=”Y1” BookingIdentifier =”1” inResponseToMessageID=”X1” status=”OK”

	Buyer	Seller
5	<- AdOrderResponse messageID=”Y1” BookingIdentifier =”1” inResponseToMessageID=”X1” status=”OK”	
6	Check that the response is correct compared to order.	
7	OK – transaction is complete.	
8	The booking must be changed. Create a change message that includes the message ID of the last received message.	
9	-> AdOrderChange messageID=”X2” BookingIdentifier =”1” lastReceivedMessageID=”Y1”	Before the buyer’s change request arrives, the seller also wants to change the booking. Create a change message that includes the message ID of the last received message.
10		<- AdOrderChange messageID=”Y2” BookingIdentifier =”1” lastReceivedMessageID=”X1”
11	<- AdOrderChange messageID=”Y2” BookingIdentifier =”1” lastReceivedMessageID=”X1”	-> AdOrderChange messageID=”X2” BookingIdentifier =”1” lastReceivedMessageID=”Y1”
12	The buyer can see that the seller’s order change request is based on message X1. However, the last message sent by the buyer was X2, so a conflict is detected. The seller’s change request is not accepted.	The seller can see that the buyer’s order change request is based on message Y1. However, the last message sent by the seller was Y2, so a conflict is detected. The buyer’s change request is not accepted.
13	-> AdOrderResponse messageID=”X3” BookingIdentifier =”1” inResponseToMessageID=”Y2” status=”ERROR”	<- AdOrderResponse messageID=”Y3” BookingIdentifier =”1” inResponseToMessageID=”X2” status=”ERROR”
14	<- AdOrderResponse messageID=”Y3” BookingIdentifier =”1” inResponseToMessageID=”X2” status=”ERROR”	-> AdOrderResponse messageID=”X3” BookingIdentifier =”1” inResponseToMessageID=”Y2” status=”ERROR”
15	Message transaction completed with errors, no changes made	Message transaction completed with errors, no changes made

At this point, in theory either party may attempt to re-initiate AdsML communications by re-sending their requested changes. More likely, however, they should first speak to each other by phone in order to agree on which changes should be made and who should initiate them.

5 Usage of Business Messages

5.1 Acceptance

An acceptance of an order or reservation by a seller indicates that the seller accepts what it believes were the significant terms requested by the buyer, not necessarily the precise text by which those terms were expressed. The amount of latitude available to the seller in interpreting the buyer's instructions, and the degree to which the seller's response message should echo back the precise terms used by the buyer, **MUST** be agreed in advance between buyer and seller. The available options in this regard are described in the Usage scenarios: "Accept as-is", "Accept with conditions" and "Accept with changes" in *Bookings Usage*.

5.1.1 Accept an order with changes

AdsML provides a mechanism by which a seller can accept an order "with changes". This is primarily meant to support scenarios such as partial acceptance (for example, when some but not all of the requested dates can be accommodated), or secondarily, relatively small changes that are thought by the seller to accommodate the spirit, if not the full details, of the buyer's order.

Acceptance "with changes" is accomplished by sending an Ad Order Response (or Ad Reservation Response) message in which the details of the order have been changed to reflect the conditions the seller is willing to accept. The seller uses the `adsml:NatureOfResponse` element to indicate that the acceptance is "with changes", and optionally the `adsml:ChangeSpecification` structure to identify the specific Placements or Insertions in which the changes have been made.

The "accept with changes" capability **SHOULD NOT** be used in cases where the seller is unable to accept a substantial part of the order and the seller-initiated changes constitute more than either a simple filtering of the order (as in the example above), or a reasonably small date or time change, or a reasonably small change of the ad's position within the specified publication. This capability **MUST NOT** be used to add new placements, insertion dates, publications or targeting instructions to a booking.

Trading partners wishing to use this capability **MUST** agree in advance on the types of information that may be changed by the seller in an acceptance, and on the value that will be placed in the `adsml:NatureOfResponse` element in order to signify that such changes have been made.

5.1.2 Synchronization of terminology and content in acceptance messages

An AdsML acceptance message provides the capability for the seller to fully specify the terms of the acceptance, at a high level by entering an appropriate code in the `adsml:NatureOfResponse` element, and at a lower level by populating the response message with the details of the accepted order as it exists in the seller's system. In doing so, the seller has the ability to change some aspects of the order (including both cosmetic, e.g. terminology, and business-significant values) from those that were requested. But this capability should be used very carefully.

Before exchanging AdsML messages, the buyer and seller **SHOULD** synchronize the controlled vocabularies and other terminology used in their systems, so that the seller's response and status messages will naturally echo back to the buyer the same code values and terminology that the buyer used. However, it is often the case that buyers and sellers are unable to fully synchronize their vocabularies, in which case there may be a mismatch between some of the terms in the request and response messages.

If vocabularies cannot be synchronized, senders of messages **SHOULD** use values originating in their respective systems.

In addition to terminology issues, there are more substantive reasons why the parties might want the seller to accept an order while changing some of the buyer's terms. For example, the buyer may have made a relatively harmless error such as providing both a name and code for the target publication, where the code is correct but the name is misspelled. Or perhaps the buyer provides several classification codes from which the seller is expected to select the most suitable one. Or the seller may wish to add or correct information that was not available to the buyer, such as providing the seller's order ID or updating the name of the sales executive handling this order.

For reasons like these, the trading partners **MAY** agree in advance that the seller is allowed to make certain limited types of cosmetic and/or business-significant changes in an acceptance. But as a general rule, a seller's response message **SHOULD NOT** change any of the business-significant information that was expressed in the buyer's request. And if the parties agree to permit such business-significant changes, changes **MUST NOT** be made to types of information that the trading partners have not previously agreed may be changed in these circumstances, and the presence of such changes in the message **MUST** be identified by a suitable `adsml:NatureOfResponse` value that has been agreed between the trading partners.

5.2 Changes and cancellations

5.2.1 Change a previously-accepted Booking

Request for changes of a booking (order or reservation) can be specified using the `AdOrderChange` and `AdReservationChange` business messages.

Change messages **MUST** be used to request business-significant changes. Updates to supplemental or supporting metadata, for example a change in the booking's status or the addition of a Note, should be communicated via status messages (`AdOrderStatus` or `AdReservationStatus`) and not via a change messages.

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

In addition to specifying the booking, the `AdOrderChange` and `AdReservationChange` business messages also allow 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 PlacementIdentifier in a changed Placement and the ScheduleEntryIdentifier in a changed InsertionPeriod. If a section of the message has been deleted (for example, an entire Placement was removed), the ChangeLocationReference can contain the identifier of the deleted information.

The change specification is intended to be used as a facilitator for the receiving applications to either automatically or with manual support update the relevant parts of a complex booking.

See also section on changes in "Use Cases and Recommended Solutions".

5.2.2 Cancel a previously-accepted Booking

Only in the case where a complete booking is cancelled should the cancellation messages be used. A partial cancellation where a single placement or insertion in a booking that also has additional placements or insertions is cancelled **MUST** be handled using a change message and **MUST NOT** be handled using a cancellation message.

Depending on the agreement between trading partners (the TPA), a change involving a cancellation of a single booking part **MAY** either be implicitly defined by the absence of the part, or explicitly defined in the change specification.

It is **RECOMMENDED** to use the explicit model when transmitting a partial cancellation. In this case, the change message includes the part that is being canceled, and an adsml:ChangeSpecification structure containing a ChangeLocationReference that points to the primary identifier of the cancelled section, and a CodeValue that indicates that the nature of the change is a cancellation, e.g. a "CANCEL" code.

See also section on changes in "Use Cases and Recommended Solutions".

5.3 The use of Status in response messages

Status is recorded at all major levels in AdsML Bookings response messages, at the booking, placement and scheduling (insertion) levels. In addition, status can be recorded at the ad content level as part of a placement (please see the *AdsMLMaterials Usage - Rules & Guidelines* for information about use of status values inside the adsml-ma:AdContent context).

Status information is conveyed in the optional adsml:Status element, which contains a code and an optional, repeatable StatusQualifier code that can provide more information about the status. Please see the *E-Commerce Usage Rules & Guidelines* for a general discussion about use of status values.

The status is recorded in code form using the CodeType defined in the AdsML Type Library, and can use a controlled vocabulary for validation.

5.4 Pricing and Payments

5.4.1 Overview

Prices can be specified at three levels: for the complete booking; for each placement group; and for each placement. On all three levels, there are two types of information

1) A Total Price for the level (e.g. `PlacementPrice`), accompanied by an optional list of the `PriceComponents` which make up that price, and

2) A `PayerInformation` structure, which indicates the party or parties who will pay for this booking, placement group or placement. The payer information includes payment data and instructions, and allows for each payer to be allocated a share of the total price. Each payer's share can be specified by either amount or percentage, so that complete payer information can be provided even when the total price was not specified.

Both types of information are optional on all levels due to the fact that a price may not be available at all times during the life span of a booking, and also that trading partners might agree that prices must not be transmitted in a booking message (price and payment may be settled using other channels). The format also has to accommodate booking of free ads, where no price will be provided.

Within the pricing area, it is possible to include positive or negative price adjustments, i.e. discounts or surcharges. These adjustments can be done to the total price, and individually to each payer's share.

The price contained in a booking is often merely an expected or "ballpark" price, which reflects the price that would be charged if the ad ran exactly as specified and no contractual agreements or other external factors caused its price to be adjusted. In practice, it is quite common for ads to run somewhat differently than ordered, and additional discounts or surcharges may be triggered by factors that are external to the booking message (for example, volume discounts).

Using these structures, it is possible to include apparently contradictory pricing information in a given Bookings message. This possibility is a necessary side effect of the flexibility which is built into the message structure in order to accommodate different regional, media and ad-class usage scenarios. It is therefore important that each set of trading partners agree in advance on the types of pricing information that they will exchange, the structures in which they will convey it, and the interpretation which should be applied to it.

See also the section on pricing in "Use Cases and Recommended Solutions".

5.4.2 Pricing semantics

It is possible for either a buyer or seller to include pricing information in most AdsMLBookings messages. But what is the intended meaning in each such case? For example, if pricing information is included in an order confirmation message, does it reflect a guaranteed price, a "not to exceed" cap, or a general guideline that might be plus or minus 20% of the final price?

It is up to trading partners or regional industry groups to agree on the intended meaning of pricing information that is included in a given type of message. In general, the expectations of the AdsML technical working group are:

- If a buyer includes pricing information in a request for a reservation or order, it reflects the buyer's expectation of the general pricing range and/or applicable contract, if any. Buyers wishing to convey other semantics, for example "price not to exceed", should define this meaning in advance via their Trading Partner Agreement, and/or explicitly state the desired meaning by using an appropriate `adsml:PriceType` code.
- Similarly, when a seller includes pricing information in a quotation, a response to an order or reservation, or a status message, the price reflects an assertion which could loosely be phrased as: "If the ad(s) are published according to the placement criteria defined in this message, the price will be as shown here. However, actual price may differ depending on circumstances."
- When a seller includes pricing information in an order status message, the expected accuracy of the price depends on whether (and *to what extent*, if it is a multi-placement order) the total order has actually been published.

Note: These issues and guidelines apply to the `adsml:TotalPrice` at each level of the booking message, which is the agreed price for the booking, placement or placement group. By contrast, information in the `PayerInformation` structures, including each payer's amount or price share, is usually provided by the buyer and, if accepted by the seller, is used to generate invoices to the specified parties.

5.4.3 Payers and their payments

A Booking message may contain information about how and by whom the booking is expected to be paid (or has already been paid). This is conveyed in the `PayerInformation` structure, which is available at every level of the message where pricing can be expressed. Each instance of `PayerInformation` identifies a single payer party and conveys details about the price that will be paid by that payer and the mechanism by which it will be paid.

It is recommended that when pricing information is contained in an order, one or more corresponding `PayerInformation` elements **SHOULD** always be provided. If an order contains pricing but `PayerInformation` is not provided, trading partners **SHOULD** assume that the entire order will be paid by the `BookingParty` using a payment mechanism and payment terms that have been agreed elsewhere.

Each instance of `PayerInformation` describes the combination of a single payer and payment method. Therefore, if more than one paying party will be related to this booking, or if a single paying party will pay by multiple payment methods, then multiple `PayerInformation` elements will be needed: at least one for each combination of payer and payment method.

5.4.3.1 Relationship to future invoices and payments

Note that there is not necessarily a one-to-one correspondence between the number of `PayerInformation` elements in an AdsMLBookings message and the number of invoices or payments that will result from that booking. It is common for a booking that involves just one paying party and one payment method to generate multiple invoices or payments: for example, if the booking covers a span of time, the seller may choose to issue invoices on a daily, weekly or monthly basis, and the payer may pay each of those invoices individually or group them into a smaller number of payments.

In order to accommodate these and other similar situations, it is strongly **RECOMMENDED** that trading partners agree in their Trading Partner Agreement on the invoicing and payment patterns that they will normally use (e.g. how often will invoices be issued and when are they expected to be paid), rather than attempting to convey that information in AdsMLBookings messages.

5.4.3.2 Relationship to payments already received

When describing payments that have already been received, for example in the case of a pre-paid classified booking, it is **RECOMMENDED** that if multiple payments have been received there should be one `PayerInformation` structure be provided for each individual pre-payment.

5.4.3.3 Pricing details

Within `PayerInformation`, it is possible to provide pricing details using two different structures: `PayersPriceDetails` and `ShareOfTotalPrice`. `PayersPriceDetails` conveys the price to be paid by this payer in complete, self-contained detail. It consists of a stack of `PriceComponents` which can be structured to represent the payer's pricing information as it would appear on an invoice. `ShareOfTotalPrice`, on the other hand, indicates the relationship between this payer's pricing information and one or more of the price components in the body of the order. For example, it can be used to indicate that this payer will pay 40% of the total color charge.

`PayersPriceDetails` is the primary method for indicating the price to be paid by a given payer, while `ShareOfTotalPrice` provides additional, supplemental information if desired. Therefore, when conveying pricing information for a given payer, `PayersPriceDetails` **SHOULD** always be populated, while `ShareOfTotalPrice` **MAY** also be populated if appropriate.

Within `PayersPriceDetails` only `adsml:TotalPrice` is mandatory. The degree to which additional pricing details are provided is up to the user.

5.4.3.4 Payment type and mechanism

The `Payment` structure provides information about the type of payment associated with this payment: for example, whether or not the payment has already been made (`Payment/adsml:Status`), and the mechanism by which the payment either has already been or will be made (e.g. by cash, credit card, in response to an invoice, etc.).

The `Payment` element does not contain pricing information. The price associated with a `Payment` is conveyed in its sibling `PayersPriceDetails/TotalPrice`.

Only one set of `Payment` information can be included in a given instance of `PayerInformation`. Therefore, if a payer wishes to use more than one payment method for an order, then multiple `PayerInformation` elements will need to be provided.

5.4.3.5 Billing Instructions

`BillingInstructions` are primarily used to describe the method that should be used for allocating the price of a long-running advertisement across multiple invoices. (For example, if the ad will run for five months, should the publisher send monthly invoices, and if so, should each invoice convey one-fifth of the total price?) However, it can also be used to convey any other type of instructions describing how the invoice or invoices should be generated.

Each instance of `BillingInstructions` is relevant only to that particular `PayerParty`. If a booking contains multiple `PayerPartys` and the same billing instructions apply to all of them, those instructions **SHOULD** be copied into each of the `PayerParty` structures.

5.4.4 Currencies

AdsMLBookings contains an optional `adsml:DocumentCurrencyCode` at the top level of the message. This element specifies the default currency for all of the pricing information contained in the document. The element is optional in order to support situations in which no pricing information is conveyed in the booking message. If any pricing information is contained in the message, however, the `adsml:DocumentCurrencyCode` **MUST** be populated.

The pricing structures contained at each lower level of a Bookings message (e.g. `PlacementGroupPrice` and `PlacementPrice`) contain an additional, optional `adsml:CurrencyCode`. These elements, if populated, override the `adsml:DocumentCurrencyCode` from the top level with respect to the pricing information found in that placement or placement group.

Due to the use of common schema components, `TotalBookingPrice` also contains a local instance of `adsml:CurrencyCode`; however since the total booking price is the highest level pricing information in the booking it **MUST** always either be omitted or populated with a copy of the `adsml:DocumentCurrencyCode`, both of which actions convey the same meaning.

As noted above, a Booking message may also contain information about how and by whom the booking is expected to be paid (`PayerInformation`). Any monetary amounts contained in `PayerInformation` are accompanied by an instance of `adsml:CurrencyCode`, which specifies the currency for that particular payment amount. These may or may not be the same as the currency for the booking as a whole.

5.4.5 Exchange rates

At each location where an `adsml:CurrencyCode` may be specified, it is also possible to indicate the `adsml:ExchangeRate` that was used to convert the prices shown in the message from another currency.

Exchange rates are purely optional. They provide supplemental information about the prices in the message, so that (for example) a seller may indicate that one or more of the prices were derived from a published rate card using a specific exchange rate as of a particular date.

5.4.6 Rules and guidelines

This section provides rules and guidelines for the inclusion of pricing and payment information.

NB: These rules apply ***only*** in those cases when trading partners have chosen to include pricing information in a message. They do not apply if there is no pricing information in the message.

5.4.6.1 Pricing and currencies

1. Pricing information contained in an AdsML Booking message is intended to support various processes such as invoice reconciliation, but **SHOULD NOT** be considered a replacement for the invoice itself.
2. If any pricing information appears anywhere in a Bookings message, a `TotalBookingPrice` for the complete booking **MUST** be provided.
3. The value of the highest-level total price in the booking **MUST** override the sum of all sub-totals that may appear lower down. In particular, a value given in `TotalBookingPrice` must override the sum of all values given in `PlacementGroupPrice elements`, which value in turn must override the sum of all `PlacementPrice` elements in that group.
4. If any pricing information appears anywhere in a Booking message, then either the trading partners **MUST** have agreed in advance on the currency that will be used in all of their transactions, or `adsml:DocumentCurrencyCode` **MUST** be populated with the currency code which applies to the document as a whole.
5. If an `adsml:CurrencyCode` is provided for the pricing information in a placement or placement group, it overrides the booking's document currency code with respect to the pricing information conveyed in that placement or placement group.
6. If an `adsml:CurrencyCode` is provided within the `TotalBookingPrice`, it **MUST** contain the same value as the `adsml:DocumentCurrencyCode` for the booking as a whole.

5.4.6.2 Payments and payer information

1. Trading partners **SHOULD** agree in advance on the invoicing and payment patterns that they will normally use (e.g. how often will invoices be issued and when are they expected to be paid).
2. At least one `PayerInformation` structure **SHOULD** be placed at the level(s) at which the booking will be invoiced, and **SHOULD NOT** be placed at any other level(s). For example, if there will be one or more invoices per placement, a `PayerInformation` structure should be included at the placement level; if the invoice(s) will be per placement group, then `PayerInformation` should be provided at the placement group level; and if there will be a single invoice for the complete booking, `PayerInformation` should be provided at the bookings level.
3. If `PayerInformation` appears at a given level, it **SHOULD** be accompanied by a total price for that level. For example, if a placement contains `PayerInformation`, then it should also contain a `PlacementPrice`.
4. When describing payments that have not yet occurred, it is **RECOMMENDED** that if payment information appears at a given level of the message, there should be one `PayerInformation` element for each unique combination of

PayerParty and Payment/Mechanism. If a payer will use more than one method of payment, then multiple PayerInformation elements **SHOULD** be provided for that payer, one for each method of payment.

5. Each instance of PayerInformation **SHOULD** contain a populated instance of PayersPriceDetails. ShareOfTotalPrice **MAY** also be populated if desired.

5.4.7 Relationship of price structures to those used in AdsMLFinancials

Although users of AdsMLBookings are not required to use AdsMLFinancials, those who do will find many of the same pricing structures in both of them, although with different names. The re-use of pricing structures across various AdsML standards is meant to make it easier for developers to implement them in software.

This structural similarity can be deceptive, however, because the way that pricing information is meant to be conveyed in AdsMLBookings is quite different from the equivalent cases in AdsMLFinancials.

In AdsMLBookings, all of the pricing information for a given section of the booking, including charges, allowances, discounts and taxes, is meant to be conveyed in a single pricing structure in that section of the booking. In AdsMLFinancials, however, each line item contains separate structures for the calculated price, allowances and taxes. The same information which, in AdsMLBookings, would be conveyed in PlacementPrice is allocated to three structures in an AdsMLFinancials line item.

This difference in approach and complexity reflects the fact that an AdsML financial document is primarily about a financial transaction and only secondarily about the other booking-related information that it may contain. It is critical that a financial document be able to describe the financial transaction in sufficient detail to satisfy both the directly involved trading partners and other financial stakeholders such as auditors and tax authorities.

A booking message, conversely, is primarily a document about the nature, timing and desired distribution for the publication of an advertisement, which only secondarily, and optionally, discusses its price. Only in the classified ad workflow, where many bookings have been pre-paid by credit card, and in a few specific countries, is it common to see pricing information in a booking that is complete to the level of detail that one would find on an invoice. The payment and pricing structures in AdsMLBookings are designed to support these flexible requirements.

5.5 Taxation information

AdsMLBookings supports the transmission of taxation information in two contexts:

1) Each Party in a Booking includes an optional adsml:PartyTaxScheme specifying taxation information that is associated with that particular party.

2) Each price component may have an associated adsml:TaxCategory that plays the same role as PartyTaxScheme above by recording tax information associated with a particular price.

Rules and guidelines for populating the taxation structures in any AdsML message can be found in the *AdsML Type Library* specification.

5.6 Targeting and Distribution

5.6.1 Placement Target

AdsMLBookings provides a "placement target" structure which allows the buyer to define the type of thing being ordered or bought in this placement, and the target number of such things that are expected to result from the placement. In effect, this structure defines "what the buyer is paying for". For example, in a web ad it is possible to specify that the placement is intended to deliver "50,000 impressions".

Use of `PlacementTarget` is strongly **RECOMMENDED** when booking interactive, inserts or generic advertisements. However, use of `PlacementTarget` is optional when booking print ads (e.g. newspapers and magazines), because in that case the type of thing being bought is understood to be "insertions" and the number of instances is defined by the scheduling instructions elsewhere in the placement.

5.6.2 Distribution (Editions, Regions, Zones, Demographics, etc.)

AdsMLBookings also provides a generic "distribution target" mechanism which allows the buyer to restrict the distribution of the advertisement to a subset of the audience that the publication is capable of reaching. This includes, but is not limited to, temporal (e.g. editions, etc.), geographic (e.g. regions, zones, micro-zones, etc.), delivery classes (e.g. single copy, newsstand, etc.) and demographic (e.g. gender, age, income, etc.) targets.

> Note: `PlacementTarget` defines the thing that is being paid for, while `DistributionTarget` describes the people/places/mechanisms to which the advertisement should be distributed in order to achieve that target.

The distribution mechanism is completely generic, and can convey any form of targeting codes that the buyer and seller have agreed to use. These can be very simple instructions (e.g. a list of editions or zones that should be included), or as complex as necessary (e.g. a precise combination of editions, zones and demographics, some of which should be included and some of which should be excluded from the distribution of the advertisement).

The distribution instructions can be conveyed using any combination of machine-processable `Code` elements and/or textual instructions conveyed in `adsml:Specifications`. As always when communicating via codes and code lists, it is important that the trading partners agree in advance on the types and meaning of the targeting codes and code values that may be conveyed in a booking message.

It is strongly **RECOMMENDED** that if codes are used to convey distribution information, the <u>names</u> of the code lists (e.g. "Edition", "Region", "Zone", "Demographic" and "Delivery Class") should be as provided in `AdsMLTargetingCodeCV`, and their usage should be consistent with the descriptions of those codes in `AdsMLTargetingCodeCV`.

> Note: If distribution information is provided in a Placement that includes multiple `Publications`, then the information in `DistributionTarget` must apply to all of those publications.

5.6.3 Region, Zone and Micro-zone codes

There are many different ways to define concepts like "region", "zone", "micro-zone" and the relationships between them. For example, some publications consider a region to be larger than a zone, while elsewhere the concepts are reversed or even orthogonal to each other.

Trading partners that wish to specify localization information precisely should agree on a suitable coding scheme to represent the regions, zones and micro-zones in which the publisher is able to deliver the advertisement. It is **RECOMMENDED** that these codes be "flat" and self-contained. For example, the code for a zone within a region might specify both the region and zone, e.g. "RegionA.Zone23" or "Northwest/1".

5.6.4 Distribution count

It is possible to express a `TotalDistributionCount` for the distribution as a whole, and/or a `DistributionCount` at each significant level within it. The meaning of a `TotalDistributionCount` or `DistributionCount` in an Order message (i.e. whether it represents a requested distribution count, a required distribution count, or the publisher's expectations of the distribution that will be achieved based on this booking) will vary according to the type of booking, the party that populated the element, and the Trading Partner Agreement between the Buyer and Seller.

NOTE: The distribution structure also appears as `DistributionResult` in `AppearanceInformation` in the AdsMLProofOfPublication and AdsMLFinancials standards. In that context it represents the publisher's assertion of the distribution that was actually achieved.

5.6.5 Processing Rules

The following rules **MUST** be applied when creating and interpreting a set of machine-processable distribution information. Because these rules are context-independent it may be hard to understand them in the abstract. It is recommended that readers carefully examine the Examples which are provided in the following section and which illustrate common applications of these rules.

1. If no distribution information is provided for a placement (i.e. it does not contain a `DistributionTarget` element), then the advertisement should be distributed to all available target points within the scope defined by the publication, sub-publication and insertion or appearance date(s) for that placement.
2. If the only distribution information provided for a given type of code list has been negated (for example, the only instance of an Edition code anywhere in the targeting information is specified as negated), then the advertisement should be distributed to all available target points of that kind except those that have been negated. This provides for the simple case of e.g. "distribute to all editions except the Late City Edition, and all delivery areas except the Northwest." This filter is combined with any other distribution information provided, according to the rules below.
3. If a `Target` element contains one or more non-negated `Codes` from a given `CodeList`, then the advertisement should be distributed only to the specified

non-negated target points of the type defined by that code list. Negated and non-negated Codes **MUST NOT** be used simultaneously inside a Target element. This building-block rule is combined with the following three rules to define the target distribution.

a. If more than one non-negated Code element is provided within a given Target element, distribution should go to the union of the different target points represented by the two or more Code elements. For example, if the Codes within a single Target element specify "Northeast delivery area" and "Southeast delivery areas", and neither of their *negated* attributes is set to "true", then the resulting target points would be "Northeast and Southeast delivery areas."

b. If more than one sibling Target set is provided within a given Targeting element, then distribution should go to the intersection of the target points specified by those two or more sibling Target sets. For example, if the Codes within the first Target element specify "Northeast and Southeast delivery areas", and the Codes within a second Target element specify "White females aged 35-45", the intersection of those two sets would be "White females aged 35-45, but only in the Northeast and Southwest delivery areas".

c. If more than one Targeting element is provided, then distribution should go to the union of the target points specified by the Targets within those Targeting elements. For example, if the information in the first Targeting element specifies "Northeast and Southeast delivery areas" and the information in the second Targeting element specifies "White females aged 35-45", the union of those two sets would be "All white females aged 35-45 no matter where they live, and <u>also</u> all target points in the Northeast and Southwest delivery areas".

Note that the combined power of the above rules allows for specification of very complex distribution scenarios in a machine-processable fashion. Such cases may however be difficult for a receiver of a bookings message to interpret correctly with the risk of misunderstanding. It is thus recommended to avoid very complex machine-processable distribution descriptions and make them as simple as possible along the lines of the examples in the following section. Additional or alternative distribution instructions can be transmitted as either text or generic codes via the adsml:Specifications element, which is available at every level of the DistributionTarget hierarchy.

5.6.6 Examples

5.6.6.1 Rule 1: No distribution information provided

Sample: N/A. (The Placement does not contain a DistributionTarget element.)

Explanation: Distribute the advertisement to all available consumers, in all available regions, zones and editions, by all available distribution methods.

5.6.6.2 Rule 2: Only negated information is provided – distribute to all targets except the negated ones

Sample:

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code adsml:negated="true">
    <adsml:CodeList xsi:type="adsml-
cv:AdsMLTargetingTypeCodeCV">DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>Northern</adsml:CodeValue>
   </Code>
   <Code adsml:negated="true">
    <adsml:CodeList xsi:type="adsml-
cv:AdsMLTargetingTypeCodeCV">Edition</adsml:CodeList>
    <adsml:CodeValue>Late City</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute the advertisement to all regions except the "North" delivery area, in all editions except the "Late City" edition.

Note that in this example the *`xsi:type`* attribute is used in order to indicate that the names of the code lists (`DeliveryArea` and `Edition`) and their intended usage are as defined in the `AdsMLTargetingTypeCodeCV`. The code values (`North` and `Late City`) are not defined by AdsML but rather must be agreed between the trading partners.

5.6.6.3 Rule 3: Only non-negated information is provided for a given type of code – distribute only to the specified targets

Sample:

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>North</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute the ad only in the "North" delivery area – in all available editions, by all available delivery classes, etc.

5.6.6.4 Rule 4: Multiple non-negated codes in a single Target element – distribute to the Union of all specified targets

Sample 1 – simple case, all codes are of the same type:

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>North</adsml:CodeValue>
   </Code>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>South</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute the ad to both the “North” and “South” delivery areas – in all available editions, by all available delivery classes, etc.

Sample 2 – similar to Sample 1, but showing the use of more precise zone or micro-zone targeting. (As recommended above, a “flat” coding scheme is used in which each zone also specifies its parent region.)

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code>
     <adsml:CodeList>Zone</adsml:CodeList>
     <adsml:CodeValue>Northern.1</adsml:CodeValue>
   </Code>
   <Code>
     <adsml:CodeList>Zone</adsml:CodeList>
     <adsml:CodeValue>Northern.33</adsml:CodeValue>
   </Code>
   <Code>
     <adsml:CodeList>Zone</adsml:CodeList>
     <adsml:CodeValue>Northern.52</adsml:CodeValue>
   </Code>
   <Code>
     <adsml:CodeList>Zone</adsml:CodeList>
     <adsml:CodeValue>Southern.22</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute the ad to all of the specified zones – in all available editions, by all available delivery classes, etc.

Sample 3 – more complex case, a mixture of codes of different types in the same `Target` element:

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code>
     <adsml:CodeList>DeliveryArea</adsml:CodeList>
     <adsml:CodeValue>North</adsml:CodeValue>
   </Code>
   <Code>
     <adsml:CodeList>DeliveryArea</adsml:CodeList>
     <adsml:CodeValue>South</adsml:CodeValue>
   </Code>
   <Code>
     <adsml:CodeList>Edition</adsml:CodeList>
     <adsml:CodeValue>Late City</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute this ad to the entire North and South delivery areas (including all editions), and also to the entire distribution of the Late City edition (including all areas in which this edition is normally distributed), but do not distribute to any other delivery areas than those specified or implied by these instructions.

Note: In this case what appears simple at first glance is actually a complex instruction to run the ad in almost all delivery areas. This may not be what the buyer intended. See the samples relating to rule 5, below, for an alternative arrangement of the same code values which results in more precise targeting and is more likely to be what the buyer intended.

5.6.6.5 Rule 5: Multiple Target elements – distribute to the Intersection of the specified target sets

Sample 1:

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>North</adsml:CodeValue>
   </Code>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>South</adsml:CodeValue>
   </Code>
  </Target>
  <Target>
   <Code>
    <adsml:CodeList>Edition</adsml:CodeList>
    <adsml:CodeValue>Late City</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute only to those targets that appear in both of the provided `Target` structures. The resulting meaning is: Distribute the ad only to the “North” and “South” delivery areas, and within those areas, only in the “Late City” edition.

Note: The sequence of the elements in the explanation is not important. The explanation could equally have been phrased as: Distribute the ad only in the “Late City” edition and only to the “North” and “South” delivery areas. This should result in exactly the same distribution of the advertisement.

Sample 2:

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>North</adsml:CodeValue>
   </Code>
   <Code>
     <adsml:CodeList>DeliveryArea</adsml:CodeList>
     <adsml:CodeValue>South</adsml:CodeValue>
   </Code>
  </Target>
  <Target>
   <Code adsml:negated=”true”>
     <adsml:CodeList>Edition</adsml:CodeList>
     <adsml:CodeValue>Late City</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute only to those targets that appear in the first but not the second of the provided `Target` structures. The resulting meaning is: Distribute the ad <u>only</u> to the “North” and “South” delivery areas, and within those delivery areas, to all editions <u>except</u> the “Late City” edition.

Sample 3:

```
<DistributionTarget>
 <Targeting>
  <Target>
   <Code adsml:negated =”true”>
     <adsml:CodeList>DeliveryArea</adsml:CodeList>
     <adsml:CodeValue>North</adsml:CodeValue>
   </Code>
   <Code adsml:negated=”true”>
     <adsml:CodeList>DeliveryArea</adsml:CodeList>
     <adsml:CodeValue>South</adsml:CodeValue>
   </Code>
  </Target>
  <Target>
   <Code adsml:negated=”true”>
     <adsml:CodeList>Edition</adsml:CodeList>
     <adsml:CodeValue>Late City</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: Distribute only to those targets that do not appear in the combined targets identified by both provided `Target` structures. The resulting meaning is: Distribute the ad to all target points of the publication, except to the “Late City” edition in the “North” and “South” delivery areas.

5.6.6.6 Rule 6: Multiple Targeting elements – distribute to the Union of the target sets defined in the Targeting elements

Sample:

```
<DistributionTarget>
 <TotalDistributionCount>55000</TotalDistributionCount>
 <Targeting>
  <Target>
   <Code>
     <adsml:CodeList>DeliveryArea</adsml:CodeList>
     <adsml:CodeValue>North</adsml:CodeValue>
   </Code>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>South</adsml:CodeValue>
   </Code>
  </Target>
  <Target>
   <Code>
    <adsml:CodeList>DeliveryClass</adsml:CodeList>
    <adsml:CodeValue>Newsstand</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
 <Targeting>
  <Target>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>East</adsml:CodeValue>
   </Code>
   <Code>
    <adsml:CodeList>DeliveryArea</adsml:CodeList>
    <adsml:CodeValue>West</adsml:CodeValue>
   </Code>
  </Target>
  <Target>
   <Code>
    <adsml:CodeList>DeliveryClass</adsml:CodeList>
    <adsml:CodeValue>Subscribers</adsml:CodeValue>
   </Code>
  </Target>
 </Targeting>
</DistributionTarget>
```

Explanation: The first Targeting element defines a target set of "Newsstands in the North and South delivery areas". The second Targeting element defines a target set of "Subscribers in the East and West delivery areas". (Each of these is constructed according to Rule 5 as described above.) The union of these two target sets results in: "Distribute to Newsstands in the North and South delivery areas, and also to Subscribers in the East and West delivery areas."

In this example a TotalDistributionCount is provided. It is also possible to express target distribution counts at every lower level of the Distribution structure, i.e. for each individual Targeting or Target.

Note: beginning with AdsMLBookings 2.5, TotalDistributionCount **SHOULD NOT** be used in situations where the EventType specified in PlacementTarget is the same as DistributionTarget and where the EventCount and TotalDistributionCount would be identical values. It is **RECOMMENDED** to use the new PlacementTarget structure instead to convey the total target count for the placement as a whole.

5.7 Linked placements

AdsMLBookings provides two mechanisms for indicating that two or more placements are "linked", or related, to each other for production and/or pricing purposes. The most common reasons why placements may be linked include:

- They must be published or broadcast together, often in a specified physical or temporal relationship (above/below on the same page, in successive slots within a single broadcast, on sequential pages in a publication, etc.)
- They are part of a Co-op or "umbrella" advertisement, in which a master space booking is then filled in with smaller ads that may be placed and/or paid for by parties other than the one which made the original booking.

Linked placements may be in the same booking, or may span several bookings.

5.7.1 Human-readable linkage information

It is often not possible (or even desirable) to convey all of the details about the relationship between two linked placements in a machine-processable fashion. Human-readable instructions for linking a placement with one or more other placements should be conveyed in `ProductionDetail/adsml:SpecialRequirements` for each of the affected placements. (For example, "This is the first of three placements that must appear on sequential pages.") This element can also, of course, convey machine-processable instructions that are specific to each placement.

5.7.2 Machine-processable linkage information

5.7.2.1 Overview

The `LinkedPlacements` structure is designed to convey machine-processable information about any linkages between the placements in a particular Bookings message and other placements. The structure is designed purely for machine processing; human-readable (text) instructions should not be conveyed in `LinkedPlacements`.

`LinkedPlacements` contains an optional `adsml:Type` code to characterize all of the linkages contained in the structure, and a stack of 1 or more `PlacementLink`s, each of which defines a specific temporal or physical relationship between two or more placements.

Each `PlacementLink` contains a mandatory `adsml:RelationshipName` code to define the nature of the relationship, and pointers to two or more specific placements (which may or may not be in this booking message). There are also optional *`anchor`* and *`adsml:sequenceNo`* attributes to further define the relationship, and an optional stack of `LinkageDetails` codes in case further machine-processable information needs to be conveyed.

When two or more placements are linked, one of them can be defined as the "anchor". The meaning of "anchor" depends on the nature of the linkage. Only some types of relationships require the specification of an anchor; for example, when specifying a relationship such as "`VerticallyAbove`" or "`PartOf`", the anchor attribute is used to indicate which placement goes above the other, or which placement is part of the other. In general, the "anchor" is the placement that is <u>not</u> referenced by the name of the relationship. Therefore if the `adsml:RelationshipName` is "`VerticallyAbove`" then the anchor is the placement that is below the others, as if the description of the code read "vertically above the anchor".

The “anchor” concept is not relevant to all types of linkage relationships. For example, in a “Group Repel” linkage there is no anchor.

When two or more placements are linked in a “sequential” relationship, explicit sequence numbers can be provided. The meaning of the sequence will depend on the medium (pages, time slots, etc.) and the nature of the ads. Explanatory information can be provided, if necessary, in `LinkageDetails`.

A placement may participate in multiple `PlacementLinks` in order to specify multiple relationships – or types of relationships - between it and other placements.

5.7.2.2 Usage rules and guidelines

In general, all of the placements belonging to a set of linked placements **SHOULD** be explicitly identified; a placement should only be omitted if its identifier is not known to the sender of the message (for example because it has not yet been booked). If all of the placements belonging to a set of linked placements are contained in the current booking, then by definition their identifiers are known, so each of them **MUST** be represented by at least one `PlacementReference` element.

If any `PlacementReferences` in a `PlacementLink` make use of the *`anchor`* attribute then all of them **MUST** explicitly include an *`anchor`* attribute, and one (but only one) of them **MUST** specify *`anchor`*`=“true”`.

If any `PlacementReferences` in a `PlacementLink` make use of the *`adsml:sequenceNo`* attribute, then all of them **MUST** explicitly include an *`adsml:sequenceNo`* attribute, each of whose values **MUST** be unique within that set. The sequence numbers **SHOULD** consist of consecutive integers beginning with “1”.

The anchor and sequence capabilities are intended to be mutually exclusive and **SHOULD NOT** be used together:

- If a set of `PlacementReferences` in a `PlacementLink` uses *`adsml:sequenceNo`*, then *`anchor`* **SHOULD** be omitted for all participants. (This is semantically identical to specifying *`anchor`*`=“false”`.)
- If a set of `PlacementReferences` in a `PlacementLink` uses the *`anchor`* attribute, then *`adsml:sequenceNo`* **SHOULD** be omitted for all participants.
- If it is necessary to define a sequence of relationships to an anchor ad, such that it would be conceivable to use both anchor and sequence capabilities simultaneously, it is **RECOMMENDED** that these be broken down into individual binary relationships to the anchor ad, each contained in its own `PlacementLink` and each using either the anchor or sequence attributes but not both of them.

If a `PlacementReference` points to a placement in another booking, then its *`bookingReference`* attribute **MUST** be present and **MUST** contain the value of the `BookingIdentifier` of that booking.

The use of *`bookingReference`* when pointing to a placement in the current booking is optional; the absence of the attribute indicates that the placement is in this booking. However it is **RECOMMENDED** that if any of the `PlacementReferences` in a `PlacementLink` use the *`bookingReference`* attribute, then all of them **SHOULD** explicitly include *`bookingReferences`*.

5.7.2.3 Example: two placements that must be published together

Sample:

```
<LinkedPlacements>
 <PlacementLink>
  <adsml:RelationshipName>
   <adsml:CodeList>AdsMLPlacementRelationshipCV</adsml:CodeList>
   <adsml:CodeValue>PublishTogether</adsml:CodeValue>
  </adsml:RelationshipName>
  <PlacementReference>aaa.com:2007-10-01:M-05THC0015</PlacementReference>
  <PlacementReference>aaa.com:2007-10-01:M-05THC0016</PlacementReference>
 </PlacementLink>
</LinkedPlacements>
```

Explanation: This is the simplest possible use of the `LinkedPlacements` structure. It identifies a linkage between two placements that must be published together but provides no details about how exactly that should be done. Additional information, if any, will be found in `ProductionDetails/adsml:SpecialRequirements` for each of the placements.

5.7.2.4 Example: two placements above/below each other on the same page

Sample:

```
<LinkedPlacements>
 <PlacementLink>
  <adsml:RelationshipName>
   <adsml:CodeList>AdsMLPlacementRelationshipCV</adsml:CodeList>
   <adsml:CodeValue>VerticallyAbove</adsml:CodeValue>
  </adsml:RelationshipName>
   <PlacementReference anchor=”true”>aaa.com:2007-10-01:M-
05THC0015</PlacementReference>
   <PlacementReference anchor=”false”>aaa.com:2007-10-01:M-
05THC0016</PlacementReference>
 </PlacementLink>
</LinkedPlacements>
```

Explanation: This example asserts a linkage between two placements and also characterizes it. The second placement (*`anchor`* = “`false`”) must appear above the first one.

5.7.2.5 Example: three placements on successive pages

Sample:

```
<LinkedPlacements>
 <adsml:Type>
  <adsml:CodeList>AdsMLPlacementLinkCV</adsml:CodeList>
  <adsml:CodeValue>Sequence</adsml:CodeValue>
 </adsml:Type>
 <PlacementLink>
  <adsml:RelationshipName>
   <adsml:CodeList>AdsMLPlacementRelationshipCV</adsml:CodeList>
   <adsml:CodeValue>SequenceAd</adsml:CodeValue>
  </adsml:RelationshipName>
```

```
   <PlacementReference adsml:sequenceNo=”1” >aaa.com:2007-10-01:M-
05THC0015</PlacementReference>
   <PlacementReference adsml:sequenceNo=”2”>aaa.com:2007-10-01:M-
05THC0016</PlacementReference>
   <PlacementReference adsml:sequenceNo=”3”>aaa.com:2007-10-01:M-
05THC0017</PlacementReference>
   <LinkageDetails>
    <adsml:CodeList>LinkageDetailsTypeCV</adsml:CodeList>
    <adsml:CodeValue>EveryPage</adsml:CodeValue>
   </LinkageDetails>
 </PlacementLink>
</LinkedPlacements>
```

Explanation: This structure uses many of the capabilities of the `LinkedPlacements` structure in order to describe a sequential linkage between three placements. The optional `adsml:Type` code at the top characterizes the overall situation as a "Sequence", and the `PlacementLink` structure provides the details: a single Placement Link with a Relationship Name of "`SequenceAd`" that points to three placements and assigns them sequence numbers "`1`" through "`3`". The `LinkageDetails` code tells us that they should be printed on "`EveryPage`", which is defined in the LinkageDetailsTypeCV as meaning sequential pages.

5.7.2.6 Example: a placement that is part of an Umbrella ad

Sample:

```
<LinkedPlacements>
 <adsml:Type>
  <adsml:CodeValue>UmbrellaAd</adsml:CodeValue>
 </adsml:Type>
 <PlacementLink>
  <adsml:RelationshipName>
   <adsml:CodeValue>PartOf</adsml:CodeValue>
  </adsml:RelationshipName>
  <PlacementReference anchor=”false”>aaa.com:2007-10-01:M-
05THC0015</PlacementReference>
  <PlacementReference bookingReference=”aaa.com:2007-10-01:L-05THC001”
anchor=”true”>aaa.com:2007-10-01:L-05THC0016</PlacementReference>
 </PlacementLink>
</LinkedPlacements>
```

Explanation: The first placement referenced by the `PlacementLink` is part of an umbrella ad; the second placement contains the master booking for that umbrella ad. The first placement has no *`bookingReference`*, implying that it is contained in the current Booking. The second placement has an explicit *`bookingReference`* which identifies the booking in which that placement can be found. (It would also be possible to include *`bookingReference`* in the first `PlacementReference`, in order to indicate explicitly that it points to the current booking.)

In this example the Code Lists are not specified; presumably they were specified in the Trading Partner Agreement.

5.8 Colors (print)

5.8.1 Overview

The `Colors` structure in `ProductionDetails.NewspaperMagazine` specifies the color(s) and finishes for a printed advertisement, and provides space for additional processing instructions if necessary. The structure contains four elements, only the first of which is mandatory:

- `ColorType`: A code that describes the overall color situation for this placement and, in conjunction with `NumberOfColors,` provides its pricing basis.
- `ColorName`: An optional stack of codes (or text strings) which describe each individual color that will be used, for example, a Pantone number or an advertiser-specific name such as "Coca-Cola Red".
- `NumberOfColors`: An optional count of the total number of colors to be used, including black.
- `adsml:SpecialRequirements`: For additional color-specific requirements, if necessary.

5.8.2 Usage

In most newspaper ads, the only information that needs to be communicated is whether or not the ad will be in color. This is accomplished by populating `ColorType` with either "`Black`" or "`ProcessColor`".

In most magazine ads (and some newspaper ads), additional information is necessary, for example, to specify the spot colors that will be used, whether a metallic finish is needed, etc. This is accomplished by placing a more detailed code in `ColorType`, and then populating `ColorName` and `NumberOfColors` as appropriate.

It is strongly **RECOMMENDED** that `ColorType` be populated with a value from the `AdsMLColorTypeCV`.

5.8.3 Examples

5.8.3.1 Example: black and white

Sample:

```
<Colors>
 <ColorType>
  <adsml:CodeValue>Black</adsml:CodeValue>
 </ColorType>
</Colors>
```

Explanation: This is the simplest possible use of the `Color` structure. The ad will be published using only black ink.

5.8.3.2 Example: full color

Sample:

```
<Colors>
 <ColorType>
  <adsml:CodeList>AdsMLColorTypeCV</adsml:CodeList>
  <adsml:CodeValue>ProcessColor</adsml:CodeValue>
 </ColorType>
</Colors>
```

Explanation: The ad will be published using process colors. Since no further details are provided it is assumed that the ad will use all available standard colors.

This is the most typical format for newspaper bookings of color ads.

5.8.3.3 Example: 3 unspecified metallic spot colors

Sample:

```
<Colors>
 <ColorType>
  <adsml:CodeList>AdsMLColorTypeCV</adsml:CodeList>
  <adsml:CodeValue>BlackMetallicSpot</adsml:CodeValue>
 </ColorType>
 <NumberOfColors>4</NumberOfColors>
</Colors>
```

Explanation: The ad will use black plus three unspecified spot colors, one or more of which will be metallic. This may be sufficient information to calculate the pricing, though it will be of limited use to the production department. Presumably more details will arrive later.

5.8.3.4 Example: 3 spot colors specified using codes

Sample:

```
<Colors>
 <ColorType>
  <adsml:CodeList>AdsMLColorTypeCV</adsml:CodeList>
  <adsml:CodeValue>BlackSpot</adsml:CodeValue>
 </ColorType>
 <ColorName>
  <adsml:CodeList>Pantone</adsml:CodeList>
  <adsml:CodeValue>123</adsml:CodeValue>
 </ColorName>
 <ColorName>
  <adsml:CodeList>Pantone</adsml:CodeList>
  <adsml:CodeValue>246</adsml:CodeValue>
 </ColorName>
 <ColorName>
  <adsml:CodeList>Custom</adsml:CodeList>
  <adsml:CodeValue>CocaColaRed</adsml:CodeValue>
 </ColorName>
 <NumberOfColors>4</NumberOfColors>
</Colors>
```

Explanation: The ad should be published using the two specified Pantone colors plus “`CocaColaRed`”, a custom color whose composition is presumed to be known by both buyer and seller.

Variations on this approach can be used for most magazine ads, by placing an appropriate value in ColorType and describing each specific color in an instance of ColorName.

5.8.3.5 Example: 3 spot colors specified using text

Sample:

```
<Colors>
 <ColorType>
  <adsml:CodeList>AdsMLColorTypeCV</adsml:CodeList>
  <adsml:CodeValue>BlackSpot</adsml:CodeValue>
 </ColorType>
 <adsml:SpecialRequirements>
   <adsml:Text>Pantone 123 and 456 plus Cocoa-Cola Red</adsml:Text>
 </adsml:SpecialRequirements>
</Colors>
```

Explanation: Same information as in the previous example, but the details have been typed by hand in a single text string rather than using machine-processable ColorName codes.

5.9 Physical Size Information

Size information can be conveyed in three different elements in an AdsMLBookings message:

1. A repeatable Size element in ProductionDetails. This is the "booked size". It is often expressed as a logical concept (e.g. "half page") rather than a specific physical size. (This structure is optional for most media types but mandatory in ProductionDetail.NewspaperMagazine.)
2. An optional Size element in MaterialsExpectations. This is the "expected size" of the ad materials that will be delivered. It should always be expressed as an actual physical size.
3. An optional Size element in each Rendering of ad content that is included in the message. This is the "delivered size". It must always be expressed as an actual physical size.

The following sections discuss the usage of each of these elements in more detail.

Note: These structures for size are available in all media types, but these usage issues are particularly relevant in the print workflow.

5.9.1 Booked Size

The Size element in ProductionDetails conveys the "booked" size of the ad, that is, the size that the buyer is willing to pay for and the publisher has agreed to publish. This can be expressed in any combination of code values (e.g. "half page" or "IABSkyscraper") and/or literal height, width and area measurements.

The booked size is a logical concept that allows the trading partners to agree on a price and on the amount of space, time or inventory that should be reserved in the publication, and it also serves as input to the process of creating and delivering the advertisement so that an appropriately sized set of materials will be delivered. It does not necessarily convey the actual size of the artwork that will be delivered to the publisher, or the actual size of the ad that will eventually appear.

5.9.1.1 Usage Rule

`ProductionDetail*/Size` is repeatable in order to allow the booked size to be conveyed in alternative units of measure: for example, in both inches and millimeters.

- When more than one `Size` element is provided in an instance of `ProductionDetail`, all of these `Size` elements **MUST** be functionally equivalent to each other, so that the recipient of the message **MAY** base its processing on any one of the `Size` elements while ignoring all of the others.

5.9.2 Expected Materials Size

The optional `Size` element in `MaterialsExpectations` contains the physical size of the ad materials that should be delivered to the publisher. Often this “expected” size is slightly different from the booked size. For example, a booked size of “half page” may be converted into the precise width and height values that the expected artwork must have in order to fulfill that booking.

There can be at most one instance of `MaterialsExpectations/Size` in each placement. Either the buyer or the seller may populate this element.

5.9.2.1 Usage Guidelines

- If a `Size` element is included in `MaterialsExpectations`, the size **SHOULD** be expressed in terms of the expected physical dimensions (width, height and/or area) of the artwork rather than merely as a logical size code.
- When the expected size is provided by the buyer, e.g. in an Ad Order message, it describes the size of the ad materials that the buyer expects to deliver. When the expected size is provided by the publisher, e.g. in an Ad Order Response message, it describes the size of the ad materials that the publisher wishes to receive. It is up to the trading partners to resolve any discrepancies that may exist between these respective expectations.

5.9.3 Rendering size

A Placement may optionally include one or more `Renderings` of the advertisement. Each `Rendering` contains metadata about a set of ad materials that are either digitally included in the Ad Order message or have been sent by other means in parallel with the booking. This feature supports workflows in which ad materials are delivered to the seller at the same time as their Ad Order.

When a `Rendering` is provided as part of a placement, an optional `Size` element in it can be used to convey the actual physical size of the ad materials that are being delivered.

5.9.3.1 Usage Guidelines

- If a size element is included in a rendering, the size **MUST** be expressed in terms of the actual physical dimensions (width, height and/or area) of that rendering rather than merely as a logical size code.

5.10 Business message dates

The message types defined in AdsMLBookings contain two business-significant dates at the top of the message:

- adsml:BusinessMessageDate
- BookingDate (in Orders and Reservations) or QuotationDate (in Quotations)

adsml:BusinessMessageDate specifies the date on which the business information in the current message was considered valid by its sender. It is available in every AdsML message. (Often it will be the same as *messageAssembledTime* in the document header, though if there was a delay in assembling the message their values may differ.)

BookingDate and QuotationDate specify the original date of an Order, Reservation or Quotation. They are defined when the order, reservation or quotation is first transmitted, and copied unchanged into any subsequent change, cancellation or status messages that reference the same Order, Reservation or Quotation.

In the initial transaction request message (e.g. AD-O, AD-R or AD-Q) adsml:BusinessMessageDate will be the same as BookingDate or QuotationDate. In subsequent messages relating to that transaction, adsml:BusinessMessageDate will be updated to reflect the date of the current message while the BookingDate or QuotationDate will retain its original value.

5.10.1 Usage rules and guidelines

adsml:BusinessMessageDate **MUST** specify a date on which the business information in the current message was considered valid by its sender. In the case of a transaction request or response message, it **SHOULD** be the date of the decision to make that request or response. In the case of a status message, it **MUST** be a date on which the status information contained in the message was valid.

BookingDate and QuotationDate **MUST** specify the original transaction request date of the current Order, Reservation or Quotation, and **MUST NOT** change in subsequent messages which relate to that Order, Reservation or Quotation.

5.11 Placements containing multiple publications

It is possible in AdsMLBookings to transmit a Placement that specifies more than one Publication. This ability is provided in order to support workflows in which a media buyer transmits a “package” order to a middleman (agency or publisher) which may then clone that order into multiple derivatives of the original order, one for each of the target publications.

5.11.1 Rules and Guidelines

In order to facilitate further processing by the receiver of the message, and to avoid some of the complexities described below, the **RECOMMENDED** approach is for the buyer to transmit multiple placements, each of them identical except for having a different `Publication`, and therefore not to use the functionality described in this section.

When a Placement in an AdsMLBookings message specifies more than one Publication, then:

1. All of the information in that Placement **MUST** be valid for all of the specified publications.
2. Subsequent messages in the order thread (including response, change and cancellation messages) **MAY** split the original placement into multiple placements, each of which references a subset of the original list of publications. Trading partners wishing to transmit multiple publications in a single placement **SHOULD** agree in advance on how they will use this functionality.
    a. For example, if the requested dates or positions are not available for all of the publications, then the response message will need to split the order into multiple Placements in order to indicate which Publications are able to accept the order. Similarly, if some of the production details will vary from one publication to another, then subsequent messages will need to move those publications into separate placements in order to transmit the varying details.

NOTE that a "publication" in an AdsMLBookings message can represent either an individual publication or a package of publications that is sold by the seller via a single publication code.

5.12 Defining an online placement: Placement Target, Distribution, Capping, Throttling and Share of Voice

This section discusses several common types of information which, between them, define the primary goals and constraints of an interactive order. There are five primary concepts involved, each represented by a different structure in AdsMLBookings:

- Overall goal of the order (Placement Target)
- Distribution instructions
- Throttling instructions
- Capping instructions
- Share of Voice

Each of these concepts is discussed below.

5.12.1 Placement Target

Every interactive order has an explicit or implicit "Placement Target" which describes the type of event, and the count of such events, that the buyer wishes to achieve; for example: "10,000 impressions" or "500,000 clicks". This is the primary goal of the order, the result for which the buyer is willing to pay a fee to the seller.

5.12.1.1 AdsML Handling

The Placement Target is expressed in `PlacementTarget` at the root level of the Placement structure. The type of event is given as an `EventType` code, and the target count is conveyed in `EventCount`. Optionally, the order may indicate a target range by populating the `MinimumEventCount` and `MaximumEventCount` elements.

The concept of a Placement Target applies equally well to orders in all media, not just interactive bookings, but in some media it is implied rather than explicitly provided. For example, a NewspaperMagazine order has an implied event type of "Insertions" and an event count of 1 insertion per published issue during the life of the placement.

5.12.2 Distribution

While the Placement Target defines the thing that is being paid for, distribution information describes the people/places/mechanisms by which or to whom the advertisement should be distributed in order to achieve the placement target. For example, demographic and regional constraints ("only show the ad to college educated users who live in New York City") would be expressed as a set of distribution constraints.

5.12.2.1 AdsML Handling

Distribution information is conveyed in the `DistributionTarget` structure. Detailed instructions for using `DistributionTarget` can be found earlier in this chapter.

5.12.3 Throttling

Throttling is the process of allocating the appearances of an advertisement to different time periods during the life of the placement. For example, if a placement specifies that an ad should appear 120,000 times over six months, it might be "throttled" so as to appear only 15,000 times per month during the first three months, but then 25,000 times per month in the latter three months.

In effect, throttling consists of allocating the target `EventCount` (defined in `PlacementTarget`) to different sub-periods within the placement.

The concept of Throttling can be applied to longer-running placements in any media, not just interactive.

5.12.3.1 AdsML Handling

Throttling is supported in AdsMLBookings by creating multiple `Scheduling` structures for the placement in question, each of which conveys the target for that sub-period in `Scheduling/EventCount`. For example, if a long-running ad is to be throttled on a monthly basis, the buyer would create a separate `Scheduling` structure for each month, each of which would convey the target `EventCount` for that month.

Throttling is often adjusted during the life of the placement. This can be done by sending Order Change messages with updated throttling information.

5.12.3.2 Rules and Guidelines

If one or more of the `Scheduling` structures in a placement contain `EventCounts` for the purpose of conveying throttling information, then:

- The placement **MUST** contain a `PlacementTarget` which defines the `EventType` and `EventTarget` for the placement as a whole.
- The instances of `Scheduling/EventCount` **MUST** refer to the same type of event as the one defined in `PlacementTarget/EventType`.
- All of the `Scheduling` structures within the placement **SHOULD** contain `EventCounts.`
- The sum of the individual `Scheduling/EventCounts` in the placement **SHOULD** equal the `PlacementTarget/EventCount` for the placement as a whole.

5.12.4 Capping

Capping is a set of constraints which limit the number of times an advertisement may be published within a defined scope. For example, an ad may be capped to be shown "no more than 5 times to the same visitor" during the run of the ad, or more commonly, "no more than twice to the same visitor on the same day." Often several caps are provided which interact with each other, for example: "no more than twice to the same visitor per session or per hour, whichever comes first". Or even: "2 impressions per visitor per session per hour, but no more than 1000 total impressions (across all visitors) per day".

5.12.4.1 Capping compared to Throttling

As can be seen from the example of "no more than 1000 total impressions per day", some caps look a lot like throttling instructions. There are several significant differences between capping and throttling, however.

Unlike throttling instructions, capping constraints can be based on any type of event, not just the one defined in `PlacementTarget/EventType`. For example, a placement whose target event is a specified number of "clicks" might well contain capping instructions that limit the number of "impressions" or "appearances" within a given time frame.

Also unlike throttling instructions, each capping instruction applies for the entire duration of the placement. If a time period is specified as part of the cap, for example "per day", then it is deemed to apply equally to each day during the run of the placement.

Capping and throttling can coexist in a placement, provided that the largest time period expressed as part of a cap is shorter than the length of the scheduling period(s) used in the placement. For example, a placement can contain a total cap "per day" if the scheduling period(s) in the placement are defined as weeks or longer, while a cap "per month" makes sense only if the scheduling period(s) each last longer than a month.

5.12.4.2 AdsML Handling

Capping instructions are conveyed in `ProductionDetail.Interactive/CappingSpecifications.`

Capping instructions can be expressed either as purely textual instructions in an `adsml:Description` element, or as a stack of one or more machine-processable `Caps`.

Each *`Cap`* defines an `EventType`, a `MaximumEventCount` and one or more constraining `Scope` codes. If two or more scopes are provided within a given *`Cap`*, they are ANDed together. For example, a common cap is a specific number of impressions "per visitor per session" or "per visitor per day". In this case two *`Scopes`* would be included, one to indicate "per visitor" and the other to indicate either "per session" or "per day".

Two or more caps can run simultaneously. For example, a cap of "1000 impressions per day" could coexist with a cap of "5 impressions per visitor".

Each cap remains in effect for the entire duration of the placement, unless it is cancelled by the buyer via an Order Change message or a non-AdsML communication.

5.12.4.3 Example: textual instructions only

```
<CappingSpecification>
  <adsml:Description>2000 impressions per day, but no more than 5 impressions
per visitor</adsml:Description>
</CappingSpecification>
```

5.12.4.4 Example: simple cap

Sample:

```
<CappingSpecification>
  <Cap>
    <EventType>
      <adsml:CodeValue>IMPRESSION</adsml:CodeValue>
    </EventType>
    <MaximumEventCount>2</MaximumEventCount>
    <Scope>
      <adsml:CodeValue>SESSION</adsml:CodeValue>
    </Scope>
  </Cap>
</CappingSpecification>
```

Explanation: A simple cap containing the bare minimum components: the event type ("impression"), count (2) and one scope ("session"). Thus: "2 impressions per session."

5.12.4.5 Example: a cap with a complex scope

Sample:

```
<CappingSpecification >
  <Cap>
    <EventType>
      <adsml:CodeValue>IMPRESSION</adsml:CodeValue>
    </EventType>
    <MaximumEventCount>5</MaximumEventCount>
    <Scope>
      <adsml:CodeValue>HOUR</adsml:CodeValue>
    </Scope>
    <Scope>
```

```
      <adsml:CodeValue>VISITOR</adsml:CodeValue>
    </Scope>
  </Cap>
</CappingSpecification>
```

Explanation: When multiple scope statements are contained in a single cap they are ANDed together, thus yielding: “5 impressions per visitor per hour”.

5.12.4.6 Example: 2 caps running simultaneously

Sample:

```
<CappingSpecification>
  <Cap>
    <EventType>
      <adsml:CodeValue>IMPRESSION</adsml:CodeValue>
    </EventType>
    <MaximumEventCount>2000</MaximumEventCount>
    <Scope>
      <adsml:CodeValue>DAY</adsml:CodeValue>
    </Scope>
  </Cap>
  <Cap>
    <EventType>
      <adsml:CodeValue>IMPRESSION</adsml:CodeValue>
    </EventType>
    <MaximumEventCount>5</MaximumEventCount>
    <Scope>
      <adsml:CodeValue>VISITOR</adsml:CodeValue>
    </Scope>
  </Cap>
</CappingSpecification>
```

Explanation: When two or more caps are provided, they all apply equally and simultaneously. Thus: “2000 impressions per day, but no more than 5 impressions per visitor”.

5.12.5 Share of Voice

It is possible in an interactive order to purchase a “share of voice” of the target publication.

Often this corresponds to a percentage of the impressions that are theoretically available at the target location. For example, a 100% “share of voice” of the home page of a website means that the ad will be seen by every single visitor to the website during the life of the placement, while a 50% “share of voice” would mean that the ad would be seen by approximately half of the website visitors during that time.

However, the “share of voice” concept can be applied to many different types of events, not just impressions.

There is no single definition of “Share of Voice”. Trading partners wishing to use the Share of Voice mechanism should agree in advance on the meaning they will apply to it, and in particular on whether and how Share of Voice should coexist with other key components of an order such as Placement Target, Capping and Throttling.

5.12.5.1 AdsML Handling

Share of voice is conveyed in ProductionDetail.Interactive/ShareOfVoice. It consists of two required elements: UnitOfMeasure and Value. UnitOfMeasure is a string, usually a code, which defines the type of event in question (e.g. "impression"), and Value is the share of voice percentage expressed as a decimal (e.g. "50").

5.13 Guarantees

AdsMLBookings includes a mechanism to indicate that the seller has agreed to explicitly guarantee some or all of the terms of the order: a Guarantees structure which is available at all three levels of the message (placement, placement group or order).

The Guarantees structure is completely generic. It contains a set of one or more codes or text strings which describe the guarantees that have been provided. For example, a *Guarantee* element containing the code value "Positioning" would indicate that the positioning information in the message has been guaranteed.

The scope of a guarantee is from the current level of the message down: guarantees conveyed at the top level of the message apply to the entire order, while guarantees contained in a placement group apply to the header of that group and all the placements within it, and guarantees conveyed inside a placement apply only to that placement.

In general, when an Order or Reservation message indicates that all or part of the order is "guaranteed", this means that the buyer strongly requests the specified handling and potentially is willing to pay more for it. When a Response or Status message indicates that all or part of the order is "guaranteed", this means that the seller has agreed to provide the requested handling. Any charges associated with the guarantee can be described in a PriceComponent.

5.13.1 Usage Rules and Guidelines

- Trading partners wishing to use *Guarantees* **SHOULD** agree in advance as to the code or text values they will use and the precise meaning of each such "guarantee" in practice.
- A guarantee that is transmitted at a given level of a message **MUST** apply to all of the information from the current level of the message downwards. For example, a guarantee conveyed at the top level of the message **MUST** apply to the entire order, including all of the placements in it, while a guarantee conveyed in the header of a placement group **MUST** apply to all of the placements in that placement group.
- The contents of a guarantee **MUST NOT** contradict or override any guarantees that were specified at a higher level of this message. Therefore, guarantees work additively: a guarantee that is transmitted at a lower level of a message applies to the information from that level downwards, *in addition to* any applicable guarantees that are found at a higher level of the message.

5.14 Planning for invoice reconciliation

Invoice reconciliation is the process by which the buyer of advertising, upon receiving an invoice for an advertisement, matches it with the original booking information and identifies any discrepancies between the two. Only if the match is good does the buyer approve payment of the invoice. One of the reasons many buyers wish to use e-commerce booking and invoice messages is to automate this process. Sellers also benefit if invoices can be reconciled more quickly and smoothly.

AdsMLBookings is quite a flexible standard and can be configured in a number of ways. For example, when transmitting an order to publish an advertisement on three different dates, it is possible to structure that order as either a single placement with three dates, or three placements each of which includes one of the dates.

Similarly, the AdsMLFinancials standard allows for a great deal of flexibility in order to accommodate different invoicing patterns. It is possible to structure an invoice so that each line on the invoice corresponds to a single publication date, or to a single placement that may have included multiple dates, or potentially to multiple placements covering a significant amount of activity (e.g. an entire month).

Automated invoice reconciliation usually works best if each line in the invoice corresponds to a single placement in a booking. So trading partners wishing to facilitate invoice reconciliation should consider how their bookings will eventually be reconciled and structure them appropriately. For example, if an order would span two weeks but invoicing will be done weekly (with one invoice line item per order during that week), then it might be best to split the order into two placements, one for each week, so that they match the way the invoices will be prepared. If invoicing is to be done daily, then it might be best to split the order into multiple placements, one for each day, again to match the line items on the invoices. And conversely, if invoicing is normally done monthly with a single line item per order per month, then it would be best for each placement to correspond to a month of activity for the publication in question.

In short: it is **RECOMMENDED** that trading partners wishing to facilitate invoice reconciliation structure their order messages so that each placement corresponds to the degree of activity that will be represented by a single line item on the resulting invoice.

NOTE: In some media and/or for some intermediary parties in the workflow, the relationship between orders and invoices occurs at the placement *group* level rather than the placement level. In this case the discussion above should be interpreted as applying to placement groups rather than placements.

5.15 Deciding where to put information that can be conveyed at more than one level of the bookings message

There are three levels at which business information can be conveyed in an AdsMLBookings message: the top level of the message (e.g. immediate children of `AdOrder`, `AdReservation`, `AdQuotation`, etc.); in the placement group header; and in the body of a placement.

Most elements in the AdsMLBookings schema are located at one and only one of these levels. However, there are several types of information that can be conveyed in the header, and/or in each placement group, and/or in each placement. These include:

- Advertiser
- Campaign
- Deal Code
- Contract
- Terms and Conditions
- Guarantees
- Cost Exempt
- Pricing information, including Currency code
- Payer information, including Payment Terms
- Document rendering
- Notes

The flexibility to convey the same information at potentially multiple levels of an AdsMLBookings message is provided in order to accommodate different business practices. However, this flexibility opens the door for potential mismatches and misunderstandings. It is important that trading partners follow a consistent set of rules and guidelines when deciding at what level of the message to transmit each type of information.

5.15.1 Usage Rules and Guidelines

Usage rules governing the location of Pricing, Payer and Currency information can be found in the "Pricing and Payments" section earlier in this document. Usage rules governing the location of Guarantees can be found in the "Guarantees" section earlier in this document. Usage rules and guidelines governing the *other* types of information that can be transmitted at more than one level of an AdsMLBookings message are provided below.

1. Information transmitted at a given level of a message **MUST** be valid for all of the placements that are its children or grandchildren at lower levels of the message. No "overriding" at lower levels is allowed.
   - For example, if an advertiser is specified in the header of an order, then all of the placements in the entire order **MUST** be for that advertiser and only that advertiser. If any of the lower levels contain a different advertiser, then `adsml:Advertiser` should not be used at the top level of the message.
2. Information in an AdsML message that could validly be transmitted at more than one level of the message **SHOULD** only be transmitted at one level.
   - For example, if an order contains multiple placements that are all for the same advertiser, then the advertiser should be specified <u>either</u> only at the top level of the message, <u>or</u> only in all of the Placement Groups, <u>or</u> only in all of the Placements.

3. It is **RECOMMENDED** that information in an AdsML message that could validly be transmitted at more than one level of the message should be transmitted at the lowest suitable level.
    - Therefore, if an order contained multiple placements that are all for the same advertiser, then the advertiser would be specified in all of the Placements.
    - Note: organizations wishing to implement systems in which information that could validly have been transmitted at a lower level of the message is transmitted at a higher level should agree in advance with their trading partners exactly where they will place the information.

5.16 Usage and Definitions of Controlled Vocabularies

AdsMLBookings enables trading partners to use controlled vocabularies (CVs), i.e. defined lists of values, for many element values. In most cases, CVs recommended by the AdsML Consortium are available in the AdsML Controlled Vocabularies schema, imported into the AdsMLBookings schema. In any case, trading partners may use any agreed value, either directly without schema based validation, or as schema defined CVs located in a user extension schema. Please see the *E-Commerce Usage Rules & Guidelines* for a general discussion about use of CVs.

6 Use Cases and Recommended Solutions

This section attempts to address, in a lightweight way, how to use the AdsML Bookings standard in many common situations. It assumes a basic familiarity with the booking format, and with the more detailed choreography and usage information provided above.

6.1 Orders and reservations

6.1.1 Place a new order (or reservation)

Scenario: A buyer keys an order into their system, which is then transmitted as an AdsMLBookings message to a seller. This is the first AdsML message relating to this transaction (i.e. it has not been preceded by an AdsML quotation or reservation message or by a non-AdsML contact such as a phone call).

AdsML handling: The buyer's message must contain the full details of the requested booking in the `AdMessageRequestModule`, and must include the primary booking identifier for this order conforming to the AdsML QID format in the `BookingIdentifier` element. (In addition, the buyer may optionally provide their own internal booking reference in the `adsml:BuyersReference` element in `AuxiliaryBookingReferences`.) The seller's response should use the `adsml:NatureOfResponse` element to indicate acceptance of the order, and should contain a confirmation copy of the full order details, possibly including pricing, as stored in the seller's system. The seller may optionally provide their internal reference for this transaction in the `adsml:SellersReference` element in `AuxiliaryBookingReferences`.

Notes:

- While the `BookingIdentifier` is used by the sequence of AdsML messages to unambiguously reference this order, from a business perspective, the IDs contained in `AuxiliaryBookingReferences` are often more important to the trading partners' business operations, because they contain the values by which each party internally refers to this order.
- The definition of what constitutes "full details" is up to the trading partner agreement between the buyer and seller; in particular, many order requests do not include pricing information.
- This same procedure applies to Reservation messages.

6.1.2 Place a conditional or "guaranteed" order

Scenario: The buyer wishes to place an order that is conditional based on the seller's ability to meet various terms.

AdsML handling: All orders are conditional, so in general there is no special handling required to accommodate this scenario. The buyer indicates the desired terms in the appropriate location(s) in the order message, and the details in the seller's response indicate the terms that the seller is willing to accept. If the buyer is unhappy with the seller's response, the buyer can always cancel the order.

However, if the parties wish to go further and identify specific portions of the order (e.g. size, color, positioning, scheduling, etc.) that must be published exactly as described in the order for the buyer to be willing to pay for publication, the parties can include a *Guarantees* element at an appropriate level of the message and use its *Code* and *Text* elements to describe the specific guarantees that have been agreed. For example, a *Guarantees* element in the header of a placement containing *CodeValue* "RunAsBooked" indicates that all aspects of the placement are guaranteed, while a *CodeValue* of "Distribution" indicates that the guarantee applies only to the distribution information in that placement.

Notes:

- Trading partners wishing to use *Guarantees* should agree in advance as to the code or text values they will use and the precise meaning of each such "guarantee" in practice.
- Any charges associated with the guarantee can be described in a PriceComponent.
- See the discussion of "Guarantees" earlier in this document for more information.

6.1.3 Record an Order that was made by phone or fax

Scenario: A buyer and seller have agreed the details of an order by phone or fax, and now wish to record it in their systems.

AdsML handling, preferred approach: The simplest approach is for the buyer to enter the details of the order into their system, which will then send an Ad Order request as usual. The seller sees the request come up in their system, confirms that it matches the agreed terms, and approves it. The seller's system sends a normal acceptance message back to the buyer.

If the parties want the seller to key in the order, the seller's system should send an Order Status message to the buyer containing the agreed terms of the order.

Note: It is not possible to send an Order Response message in response to a phone call or fax, because a response message must reference the AdsML message to which it is a response, and in this case no AdsML message has been received.

AdsML handling, variation for seller-initiated order transactions: If the seller wishes to initiate the message exchange even though the buyer's unique ID is not known, then the seller should generate a Quotation (AD-Q) message and send the Quotation to the buyer for approval. Once the buyer has reviewed and approved the order details contained in the quotation, the buyer can convert the quotation into an order and send an Ad Order message that references the seller's Quotation ID. When the seller responds with an Ad Order Response message accepting the order and providing the seller's unique booking ID, the transaction has been fully confirmed.

In this scenario, when the buyer sends the Ad Order message, it is treated as the initiating message in a request-response transaction pair, rather than a response to the Quotation message which preceded it. The Ad Quotation message is treated as a "broadcast" message which does not generate a business message response. In effect the three business-level messages constitute two message exchanges: a "broadcast" of the Quotation (one business message), followed by a "request-response" of the Ad Order messages (two business messages).

6.1.4 Convert a Reservation into an Order

Scenario: A buyer has previously placed a reservation using an AdsML Reservation (AD-R) message, which was accepted by the seller. Now the buyer wishes to convert that reservation into an order.

AdsML handling: The buyer sends an Ad Order (AD-O) message which uses the `ReservationReference` element to convey the booking identifier of the previously confirmed reservation. Use of this element indicates that the buyer wishes to "roll over" the reservation into an order. No other details about the reservation are provided. The seller replies with a completely filled-in response message, as if in reply to a fully specified AD-O message. The AD-OR uses the `BookingIdentifier` element to convey the required ID, and the elements in the `AdMessageRequestModule` to convey the details of the order and its price.

Notes:

- The `BookingIdentifier` in an order that was converted from a reservation must be the same as the `BookingIdentifier` in the source reservation, in order to preserve the continuity between them.
- This mechanism only works if the buyer is confirming the reservation without changing it. Changes must be handled either by first submitting a change to the reservation (and getting an acceptance back from the seller), after which the changed reservation can be converted into an order, or more likely by canceling the reservation (or letting it expire) and sending a new order request containing the changed details as if from scratch.
- The `ReservationReference` element is only available in the initial Ad Order message, and not in any Response, Change or Status messages which may refer to that order. Its purpose is to trigger the conversion of the reservation into an order in the seller's system. Once that conversion has been performed, all messages which reference the order should use the normal structures to reference it.

6.1.5 Accept an order "as-is"

Scenario: A seller wishes to accept an order or reservation.

AdsML handling: The seller sends an Ad Order Response (or Ad Reservation Response) message containing a copy of the full order details as they exist in the seller's system, and using `adsml:NatureOfResponse` to indicate that the order has been accepted "as is".

Notes:

- The definition of what details should be contained in an Order Response is up to the trading partner agreement between the buyer and seller. It is **RECOMMENDED** that the response contain as much information as possible, and that it use terminology and code values as stored in the seller's system.
- An "as-is" acceptance indicates that the seller accepts what they believe were the significant terms requested by the buyer, not necessarily the precise text or code values that were used in the request message. For example, if the buyer

misspelled the publication name but their intentions were clear, the seller can correct this spelling error while accepting the order without changing the "as-is" nature of that acceptance.

- It is possible that some of the values stored in the seller's system will be different from the equivalent values stored in the buyer's system. For example, if the buyer and seller did not synchronize all their code lists, then in order to process the order, the seller must convert the buyer's requested code values (e.g. size code, region code, etc) into the equivalent values that are used in the seller's system. In this case, the parties should agree in advance on whether the response message will include any such values that were changed by the seller, and if so, whether a response containing changed values will be coded as acceptance "as-is" or "with clarifications". Options include: the acceptance message echoes back the buyer's requested values even though they are not the values stored in the seller's system; the acceptance message reflects the seller's values even though they are not the values requested by the buyer; the acceptance message omits any values that had to be changed in the seller's system, but uses the `adsml:NatureOfResponse` code to indicate that all of the buyer's requests were accepted. It is **RECOMMENDED** that the response message contain the values that were recorded in the seller's system, and that the parties determine in advance which changed values should trigger a `adsml:NatureOfResponse` indicating that clarifications or changes were made.

6.1.6 Accept an order "with clarifications"

Scenario: A seller wishes to accept an order or reservation substantially as requested, but some of the details in the request needed to be changed in order to achieve the buyer's goals. (For example, the buyer used a text string such as "Northwest region" to specify the distribution, which the seller has converted into a code with value "NW.12" in their system. Or, the buyer included multiple possible classification codes, from which the seller has selected the most appropriate one for their publication.)

AdsML handling: The seller responds with an Ad Order Response (or Ad Reservation Response) message which mirrors back the details of the accepted order as they exist in the seller's system. The seller uses the `adsml:NatureOfResponse` element to indicate that the acceptance is "with clarifications", and optionally the `adsml:ChangeSpecification` structure to identify the specific Placements or Insertions in which the clarifications were made.

Note: Trading partners should agree in advance whether they will use this capability, and if so, which types of clarifications are acceptable in an accepted order "with clarifications".

6.1.7 Accept an order "with changes"

Scenario: A seller wishes to accept some, but not all, of the placement conditions for an order or reservation that it has received. (For example, the placement is for four days but only three of them are available.)

AdsML handling: The seller responds with an Ad Order Response (or Ad Reservation Response) message in which the details of the order have been changed to reflect the conditions the seller is willing to accept. The seller uses the adsml:NatureOfResponse element to indicate that the acceptance is "with changes", and optionally the adsml:ChangeSpecification structure to identify the specific Placements or Insertions in which the changes have been made.

Notes:

- From the AdsML perspective, once this Ad Order Response message has been sent, the message exchange transaction is finished, and it is assumed that the ad will run according to the seller's changed instructions in the response message. If the buyer wants to continue the dialogue, they must send either a cancellation or change message, which will initiate a new request-response message pair.
- Therefore, the "accept with changes" capability should only be used in cases where the seller is able to accept a substantial part of the order and the seller-initiated changes constitute either a simple filtering of the order (as in the example above), or a reasonably small date or time change, or a reasonably small change of the ad's position within the specified publication. This capability should not be used to add new placements, insertion dates, publications or targeting instructions to a booking.
- Trading partners should agree in advance whether they will use this capability.

6.1.8 Accept an order "with conditions"

Scenario: A seller wishes to accept an order pending satisfaction of external conditions, for example a credit check.

AdsML handling: The seller should send an Order Response message that includes a status code indicating the nature of the situation. Later, when the status of the order changes, the seller should either send a Status update message indicating that the order is now being successfully processed or an Order Cancellation message if the condition could not be resolved.

Notes:

- Important status codes like these should be defined and agreed in advance between the trading partners.
- Acceptance conditions can be specified in both the "accept as-is" and "accept with changes" and "accept with clarifications" scenarios described above.

6.1.9 Reject an order

Scenario: A seller wishes to reject an order or reservation.

AdsML handling: The seller sends a response message using the adsml:RequestDenied and ReasonForDenial elements in the AdMessageResponseModule to indicate the nature of the denial.

6.1.10 Renew an order

Scenario: The buyer wishes to place an order that is essentially a copy of a previous order with only a few changes, for example the appearance dates.

AdsML handling: AdsML does not explicitly support the concept of a "renewal", so the buyer must place a new order and indicate by text notes or user-defined codes that this order is similar to a previous order.

Note: Often a renewal involves re-running previous artwork, often called a "pickup". This concept is supported by use of the `PickUp` structure which is available in each placement.

6.1.11 Order an Insert

Scenario: Buyer wishes to order an insert.

AdsML handling: The buyer should use the structures in `Placement.Insert`.

6.1.12 Order an Outdoor ad

Scenario: Buyer wishes to order traditional outdoor billboard ads.

AdsML handling: This is not yet explicitly supported with a media specific placement structure. However, the details can be conveyed using the `Placement.Generic` structure, using a `MediaType` value of "Outdoor" and an appropriate `AdType` such as "OutdoorBillboard".

6.1.13 Order an Outdoor Digital ad

Scenario: Buyer wishes to order outdoor digital ads that will be transmitted to their delivery devices and displayed digitally in rotation with other advertisements.

AdsML handling: This is not yet explicitly supported with a media specific placement structure. However, the details can be conveyed using the `Placement.Generic` structure, using a `MediaType` value of "Outdoor" and an appropriate `AdType` such as "OutdoorBillboardDigital".

Note: Use of Placement.Generic is recommended for the general case. All of the necessary information can be transmitted using the generic structures. However, some trading partners may prefer to use `Placement.Interactive`, because it provides named elements such as `CappingSpecification`, `ShareOfVoice` and `AdServer` that may be useful in the outdoor digital context. Trading Partners wishing to exchange information about Outdoor Digital bookings should agree in advance on which of these approaches they will follow.

6.1.14 Order an Online Classified ad

Scenario: Buyer wishes to order an online classified ad for a newspaper or magazine's website.

AdsML handling: The buyer should use the structures in `Placement.Interactive`.

The section of the website in which the ad should appear is conveyed in `PrimaryPositioning/SectionCode`. The period during which the ad will be listed at the website is specified in `Scheduling`. Additional details about the contents of the ad can be conveyed in `adsml-ma:AdContent`, using the `AdContentSearchText`, `AdContentText` and `StructuredDescriptons` structures.

6.1.15 Order an Interactive ad

Scenario: Buyer wishes to order ads in an interactive medium, including capping and throttling information.

AdsML handling: The buyer should use the structures in `Placement.Interactive`.

The target for the booking (e.g. number of impressions or clicks) is described in `PlacementTarget`, while the section of the website in which the ad should appear is conveyed in `PrimaryPositioning/SectionCode`. Demographic and other targeting instructions are placed in `DistributionTarget`. Throttling instructions can be conveyed by providing multiple `Scheduling` structures and using `Scheduling/EventCount` to constrain the number of events (e.g. impressions or clicks) that should be generated in each schedule period.

Capping instructions and other technical details go in `ProductionDetail.Interactive`, using the `CappingSpecification`, `ShareOfVoice`, `AdServingSystem` and `TechnicalAdFormat` structures.

Notes: The AdsML Framework includes several sample interactive booking messages which demonstrate many of these capabilities.

6.1.16 Order a Broadcast or Cable ad

Scenario: Buyer wishes to order ads in a broadcast or cable medium, such as radio or television, where the primary factors are the channel on which the ad should appear and the programs, times and/or dayparts in which it should run.

AdsML handling: The buyer should use the structures in `Placement.Generic`.

Depending on how the broadcaster has structured its rate card, the channel and (optionally) the program during which the ad should run can be conveyed using `PublicationCode` and `SubPublicationCode`. Instructions regarding exactly when during the program the ad should be broadcast, e.g. "first commercial break", can be conveyed in `ProductionDetail.Generic` using `PrimaryPositioning/SectionCode`.

Each instance of the `Scheduling` structure conveys a date range within which the ad should run. This can be as narrow or wide a range as desired. Other elements in the `Scheduling` structure can be used to specify the number of times the ad should run during the specified period, and a recurrence pattern (e.g. "`Tuesdays`").

If the ad is being booked for a specific time of day (i.e. a daypart), this can also be specified in the `RecurrencePattern` element, using a code or text string that identifies the daypart in question (e.g. "`MorningDriveTime`").

The `Duration` of the ad can be conveyed in `ProductionDetail.Generic`, along with any other necessary technical information.

Notes:

Multiple codes and/or text strings can be combined within a single `RecurrencePattern` provided that they are not mutually exclusive. For example, the codes "`MorningDriveTime`" and "`Tuesdays`" can coexist inside a `RecurrencePattern` element because they have different time scales; together they indicate that the ad should run during morning drive time hours on Tuesdays.

However, it is not possible to specify multiple dayparts in a single `RecurrencePattern` because dayparts are mutually exclusive. Buyers wishing to book multiple dayparts will need to transmit two or more `Scheduling` elements, one for each of the desired dayparts.

6.1.17 Order a Sponsorship

Scenario: Buyer wishes to order a sponsorship.

AdsML handling: Use the `Placement.Generic` structure, specifying an Ad Type code value of "Sponsorship".

Notes: A sample sponsorship booking message can be found in the Bookings Samples directory of the AdsML Framework.

6.1.18 Request several equally acceptable positions

Scenario: The buyer wishes to request that an advertisement be published in one of two (or more) equally acceptable locations: for example, in either the Food or LifeStyle section, or on either the "first page" or "last page" of a section in print publications.

AdsML handling: The buyer populates two (or more) instances of `AlternativePositioning`, each of which fully describes one of the acceptable positions. If the buyer has a mild preference between the alternative positions, this should be expressed by means of the *`adsml:priority`* attribute. In this scenario the `PrimaryPositioning` structure is not used.

Notes:

The use of `AlternativePositioning` structures without providing `PrimaryPositioning` indicates that the buyer is willing to accept any of the alternative positions. The *`adsml:priority`* value (if one is provided) offers guidance as to the buyer's preference between the alternatives, and should be respected by the seller if possible.

If the buyer wishes to ensure that an advertisement is published in a particular position, the buyer should use the `PrimaryPositioning` structure to describe that position.

6.1.19 Place a "package" order for multiple publications

Scenario: A media agency wishes to transmit an order that applies identically to multiple publications. This will go to a middleman that will split the agency's order into a set of smaller orders, one for each of the target publications.

AdsML handling: The recommended approach is for the agency to transmit multiple placements, each of them identical except for having a different `Publication`. However, if the parties wish to consolidate the information into a smaller package, the buyer can transmit a single Placement that describes the details of the order and contains multiple instances of `Publication`. The recipient of the message can then clone it into multiple placements to be transmitted to the target publications.

Notes:

Multiple Publications can only be specified in a single placement if all of the other information in that placement is valid for all of the specified publications. For instance, if a target distribution is included in the placement, then the information in DistributionTarget must apply equally to all of the referenced publications.

When multiple Publications are specified in a single placement, then in subsequent messages in the thread (e.g. responses, changes and cancellations) it may be necessary to split the original placement into multiple placements. This may be required, for example, if it turns out that not all the publications can accept the requested order, or if the buyer or seller wish to change details for some but not all of the publications.

A "publication" in an AdsMLBookings message can represent either an individual publication or a package of publications that is sold by the seller via a single publication code.

6.1.20 Order an A/B split (print) or other forms of alternating content

Scenario: Buyer wishes to order an A/B split or equivalent, in which the advertiser will provide two or more versions of the artwork that should be distributed across the entire run.

AdsML handling: This order should be booked as a single Placement.NewspaperMagazine, Placement.Inserts or Placement.Generic. The MultipleAdContentHandling element in ProductionDetail should specify the special handling that is related to having more than one set of content – in this case, it would indicate that an A/B split is requested. The handling should be specified using machine readable codes, and/or human readable instructions.

Note: This is not the same as requesting different versions of an ad to run in different regions or zones, which would normally be handled by multiple placements, each of which would specify the target locations for that particular set of ad content.

6.1.21 Order blind boxes or additional services

Scenario: Buyer wishes to order "blind box" response-handling services to be provided by the publisher.

AdsML handling: Blind boxes and other ancillary services provided by the publisher (or its agents) are considered "additional services." Details about the services can be conveyed in instances of AdditionalService using codes and/or text, while related pricing information can be conveyed using one or more PriceComponents.

6.1.22 Order a "Cuttable" ad (print)

Scenario: Buyer wishes to place a print ad that contains a cuttable coupon. Requirement is that the ad does not appear back-to-back with any other cuttable ad. There may be an extra charge for this.

AdsML handling: Set the CuttablePosition element to '*true*' in the Positioning section of ProductionDetail.NewspaperMagazine.

6.1.23 Specify throttling instructions

Scenario: Buyer wishes to indicate that an ad should be published more frequently during the first two months of a six month campaign than in the later months, reserving the right to change these instructions later in the campaign.

AdsML handling: In the initial order, the buyer specifies the event count for the campaign as a whole (e.g. 500,000 impressions) in `PlacementTarget/EventCount`, but then divides this into individual monthly totals by creating a `Scheduling` structure for each month of the campaign and populates each instance of `Scheduling/EventCount` with the target number of events (e.g. impressions) for that month. In this scenario, the first two months are assigned a higher `EventCount` value than the later months of the campaign.

When the buyer wishes to change the throttling instructions during the course of the campaign, he sends an Ad Order Change message containing the updated values.

Notes:

- Throttling instructions can be conveyed in orders for any media, but in NewspaperMagazine or Inserts orders, the `InsertionPeriod` element is used instead of `Scheduling`.
- Throttling can be applied to any type of time period within an order (e.g. hours, days, weeks or months), provided that those time periods are defined by individual scheduling structures whose `EventCounts` add up to the total `PlacementTarget/EventCount` for the order as a whole.
- The scheduling periods to which throttling is applied do not all have to be the same type. For example, during the initial part of a campaign, throttling could be specified on a week by week basis, dropping back to monthly throttling later on.
- A more detailed description of the throttling rules can be found in **Usage of Business Messages** earlier in this document.

6.1.24 Specify capping instructions (interactive)

Scenario: Buyer wishes to indicate that an interactive ad should be shown no more than five times per day to each user who visits the website.

AdsML handling: Capping instructions are conveyed in `ProductionDetail.Interactive/CappingSpecification`. A textual description of the cap, e.g. "`no more than 5 impressions per visitor per day`," should be placed in `adsml:Description`. A machine-processable version of the same instructions can be conveyed in the `Cap` element, using the child elements `EventType` (e.g. "`impressions`"), `MaximumEventCount` ("`5`") and `Scope`. In this case two Scope elements are required, one containing the value "`visitor`" and the other "`day`". The union of the two indicates that the cap is "per visitor, per day".

Notes:

- It is recommended that, whenever possible, controlled vocabularies from the AdsMLUnitOfMeasureCV should be used for both `EventType` and `Scope`. In this example, "impressions" and "day" are both defined in the AdsMLUnitOfMeasureCV.

- There is a detailed description of capping, including several examples, in **Usage of Business Messages** earlier in this document.

6.1.25 Specify the billing statistics or proof of performance information that will be required

Scenario: Buyer wishes to indicate that invoices relating to this order will only be paid when supported by statistics from a specified third party indicating how often the ad actually ran.

AdsML handling: Information about each type of performance-related information that will be provided, including both the nature of the information and the party that will provide it, can be conveyed in an instance of `Placement.*/ProofOfPublication`. (All performance-related information is considered a form of "proof of publication" in the AdsML Framework.)

The type of proof, in this case "`PerformanceInformation`", is conveyed in the mandatory `ProofType` element. The sender populates `ProvenanceParty` with the name of the third party that will gather and authenticate the performance information.

The party that will deliver the performance information is the `ProofingParty`. In many cases the `ProvenanceParty` will also deliver the performance information, so `ProofingParty` can be omitted.

The buyer can optionally specify an address to which the proof of publication should be sent by populating the `SendTo` structure. In many cases the parties will already have agreed where to send this information as part of their Trading Partner Agreement, in which case this element can be omitted.

Notes:

- `ProofOfPublication` is repeatable, in order to support situations where more than one type of proof of publication will be provided for the same placement.
- Recommended code values for `ProofType` can be found in `AdsMLProofOfPublicationTypeCV`.
- In many cases the same party will both gather the performance related information and also deliver it to the seller.
- If `ProvenanceParty` and `ProofingParty` are both omitted, their absence indicates that the Publisher will play both roles.

6.1.26 Specify the starting column of an unbalanced double-truck ad (print)

Scenario: Buyer wishes to order a print ad that requires a specific position on just one axis of the page, in this case an unbalanced double-truck whose height fills the entire page but whose left side will start in the 3rd column of the left-hand page.

AdsML handling: Use the `AbsolutePosition` capability inside `PositionOnPage` but fill in only the `XCoordinate` (left-to-right axis) and omit the `YCoordinate`. Specify "`columns`" as the X-coordinate's `UnitOfMeasure`, and

"3" as its Value, thus indicating that the ad should begin in the third column from the left side of the page.

Note: It would also be possible to populate the YCoordinate with a value of "0" to show that there is no offset from the top of the page, but it is **RECOMMENDED** that when only one axis needs to be specified the other axis should be omitted.

6.2 Ordering "umbrella" and other linked ads

6.2.1 Ordering the umbrella "parent"

Scenario: A buyer wishes to place an "umbrella" ad, in which the buyer reserves a large space (e.g. a full page) within which other, related organizations will later submit orders for "child" ads. The buyer guarantees payment for the entire page, but payments from the child ads will offset the buyer's guarantee.

AdsML handling: The buyer places an order in which the IsStandAlone flag in the Placement is set to '*false*' in order to indicate that this placement is or will be linked to others. A code and/or descriptive text explaining that this is an umbrella parent should also be put in the adsml:AdType element of the Placement structure.

Note: The buyer should provide the BookingIdentifier for this order, as well as the PlacementIdentifier for the placement that defines the umbrella parent, to all parties who are likely to order child ads that will go inside the umbrella parent. This will enable them to specify, when they place their orders, which umbrella their ads belong to.

6.2.2 Ordering an umbrella "child"

Scenario: A buyer wishes to place a "child" ad that should be positioned inside a previously ordered umbrella parent.

AdsML handling: The buyer places an order in which the IsStandAlone flag in the Placement is set to '*false*', and uses the LinkedPlacements structure to indicate both the nature of the overall relationship (i.e. LinkedPlacements/adsml:Type = "UmbrellaAd") and the placement and booking identifiers of the parent umbrella ad (an instance of PlacementLink with RelationshipName = "PartOf" and pointers to the two placements in question). A code and/or descriptive text explaining that this is an umbrella child should also be put in the adsml:AdType element of the Placement structure.

Note: If the buyer is not able to populate the linked placement structure with the appropriate IDs of the parent ad, as much information as possible should be put in ProductionDetail/adsml:SpecialRequirements in order for the seller to resolve the linkage with manual handling.

6.2.3 Ordering a series of ads

Scenario: Buyer wishes to place a series of ads that have a physical or temporal relationship to each other, for example, appearing on successive pages of a publication or slots in a broadcast.

AdsML handling: Each ad is recorded as a separate placement, and the LinkedPlacements structure is then used to relate them to each other and indicate the nature of the relationship. It is **RECOMMENDED** to use a value from the AdsMLPlacementLinkCV controlled vocabulary in the on LinkedPlacements/adsml:Type/adsml:CodeValue to identify the nature of the overall linkage (e.g. "Scatter", "Sequence" or "UmbrellaAd"), and one or more values from AdsMLPlacementRelationshipCV in LinkedPlacements/PlacementLink/adsml:RelationshipName/adsml:CodeValue to identify each specific physical or temporal relationship between the linked placements (e.g. "HorizontallyAdjacent" or "SequenceAd").

Notes:

- If the ads are to be published in a specific order, a relationship name of "SequenceAd" should be provided, the attribute *adsml:sequenceNo* should identify the sequence in which they are to be published, and the attribute *anchor* should be "False" for all participants.
- It is possible to use LinkedPlacements to identify that an ad in one booking is linked (related) to an ad in another booking, which may or may not be in the same AdsMLBookings message.

6.2.4 Specifying presentation sequence and other logic for banners

Scenario: A buyer is booking a set of banner placements and would like to specify constraints for how individual banner ads should or should not appear in relation to one another. For example, the banners must not be shown at the same time and must be displayed in a particular sequence.

AdsML handling: Instructions on how ads from different placements relate should be given using the LinkedPlacement element. The use case "*Ordering a series of ads*", describes the handling of the requirement to display advertisements in general in a particular sequence. The concept of 'sequence' applies generally per publishing occasion; for interactive banner ads that should be interpreted as a display sequence *per recipient (reader)*.

The requirement for sequential presentation is implicitly covering a "do not publish simultaneously" use case that does not need to be explicitly stated. If additional specifications are required to describe the business case, then the optional and repeatable LinkageDetails element should be used to record them.

6.3 Changes

6.3.1 Add a placement or date

Scenario: Buyer wishes to add one or more placements or dates to an existing order.

AdsML handling: The buyer submits an Order Change message which copies the information from the existing order and adds the new information. If the change is simply to add more dates to an existing placement, the buyer simply adds or edits one or more InsertionPeriods or Scheduling entries in an existing Placement. If the change is more substantial, for example involving a different delivery channel or different artwork, the buyer adds one or more Placements to the order. As a convenience to the recipient of the message, the buyer can identify the added insertions or placements. This is done by listing the relevant identifiers for the placement or insertions in the adsml:ChangeLocationReference element.

Notes:

In NewspaperMagazine and Inserts bookings, scheduling information is called an "Insertion" and is conveyed in an InsertionPeriod element; for all other media the information is conveyed in a Scheduling structure.

While it is **RECOMMENDED,** the sender of the message is not required to identify the sections of the order that have been added. The burden is on the recipient to be able to process an incoming change order and update its database appropriately.

6.3.2 Change a placement or date

Scenario: Buyer wishes to change one or more placements or dates on an existing order.

AdsML handling: The buyer submits an Order Change message which copies the information from the existing order and changes the appropriate details. As a convenience to the recipient of the message, the buyer can identify the locations of the changes in the order down to specific Placements, InsertionPeriods or Scheduling entries . This is done by listing the relevant identifiers for the placement insertions or schedule entries in the adsml:ChangeLocationReference element.

Notes:

In NewspaperMagazine and Inserts bookings, scheduling information is called an "Insertion" and is conveyed in an InsertionPeriod element; for all other media the information is conveyed in a Scheduling structure.

While it is **RECOMMENDED,** the sender of the message is not required to identify the sections of the order that have been changed. The burden is on the recipient to be able to process an incoming change order and update their database appropriately.

6.3.3 Publisher-initiated changes

Scenario: The seller needs to change some of the details of an accepted order.

AdsML handling: The seller sends an Order Change message to the buyer containing the changed information. The buyer responds with an Order Response message either accepting or rejecting the proposed changes.

Note: In general, buyers and sellers should agree in advance on whether a buyer needs to respond to publisher-initiated changes before they go into effect. In practice, however, many types of changes (for example, the cancellation of one or more placements) are completely under the seller's control, while others (for example adding new placements) should be completely under the buyer's control.

6.3.4 Changes to status, notes and similar supplemental information

Scenario: The buyer or seller wish to communicate changes to supplemental information about a reservation or order, for example, a change in its status or the addition of a new note, without changing the business-significant terms of the order.

AdsML handling: The party in question sends a Status message (Ad Order Status or Ad Reservation Status) containing the changed supplemental information. The message is sent in "broadcast" mode so only an Administrative Response to it is provided.

Note: In general, updates to supplemental information that occur after the initial request-response message exchange should only be communicated by AdsML messages if the sending party is providing this information on a "For Your Information" basis. If a party wishes to trigger a particular response in the other party without changing or canceling the terms of the booking, the party should use a non-AdsML communications mechanism, i.e. phone or fax, to achieve this purpose.

6.4 Cancellations

6.4.1 Cancel an entire order (or reservation)

Scenario: A buyer wishes to cancel an entire order, including all of the placements that were part of that order.

AdsML handling: The buyer's system should send an Ad Order Cancellation message referencing the order to be cancelled. The seller responds with an Ad Order Response message either accepting the cancellation (via the `adsml:Status` element) or rejecting it (via the `adsml:RequestDenied` element).

Note: Once an order has been cancelled, neither party can later attempt to change any aspect of that order.

6.4.2 Cancel part of an order

Scenario: A buyer wishes to cancel part of an order, for example, one or more insertion or run dates within a larger order.

AdsML handling: Although the buyer's software system may call this a cancellation in its user interface, the system must send an Ad Order Change message to the seller's system (rather than an Ad Order Cancellation), because from AdsML's perspective this order is not being cancelled. Rather, it is being changed, where the "change" consists of canceling one or more of the dates within it.

Note: See also the *E-Commerce Usage Rules & Guidelines* for more information about how to use `adsml:ChangeSpecification` to described a partial cancellation.

6.4.3 Un-cancel an entire order

Scenario: After canceling an order and receiving a reply from the seller confirming the cancellation, the buyer wishes to un-cancel the order.

AdsML handling: This is not supported. The buyer must re-book the order.

6.4.4 Un-cancel part of an order

Scenario: After canceling part of an order (via a Change message) and receiving a reply from the seller confirming the cancellation, the buyer wishes to un-cancel the portion that was just canceled.

AdsML handling: AdsML does not provide an "un-cancel" capability. The buyer must send another Ad Order Change message, re-booking the portions of the order that were previously canceled.

Note: The seller has the option of accepting or rejecting these "un-canceled" placements as if they were new orders.

6.4.5 Publisher-initiated cancellation

Scenario: A seller discovers that it cannot fulfill all or part of a previously accepted order and now wishes to cancel the affected order or placements.

AdsML handling: The seller can initiate this transaction by sending either an Order Cancellation or Order Change message to the buyer. The buyer should respond with an Order Response message accepting the change or cancellation.

Note: Unlike a buyer-initiated cancellation, which can be accepted or rejected by the seller (for example, it will be rejected if the ad has already run), a publisher-initiated cancellation should always be accepted by the buyer because it is simply a statement of fact – the ad will not run as booked. Related business factors, such as the impact of this cancellation on business agreements between the buyer and seller, should be discussed directly between the parties, not in AdsML messages.

6.5 Quotations

6.5.1 Obtain a quotation

Scenario: A buyer wishes to obtain a quotation for the running of a particular advertisement.

AdsML handling: The buyer sends an Ad Quotation Request message (AD-RFQ) containing full details of the order they are thinking of placing. The seller responds with an Ad Quotation (AD-Q) which repeats back to the buyer the placement and production details that were included in the request, and adds appropriate pricing information and an `adsml:ExpirationTime` indicating the date on which the quotation will expire.

Note: The business commitment (if any) that is implied by the provision of a quotation should be agreed between the parties beforehand.

6.5.2 Convert a valid quotation into an order or reservation

Scenario: A buyer wishes to accept a quotation and convert it into an order. It is a valid, unexpired quotation that was previously recorded via an exchange of AdsML messages.

AdsML handling: AdsMLBookings does not support explicit conversion of a quotation into a confirmed order. The order is required to contain a complete listing of the data required to book the order, even though the same data may have been provided in a prior quotation.

An order may include a reference to a prior quotation using the `QuotationReference` element. There is, however, no requirement that the data in a booking that references a quotation should be the same as the data in the quotation. The quotation reference should be seen as "for information" only.

6.5.3 Convert an expired quotation into an order

Scenario: A buyer sends an Ad Order message that references an expired quotation (as indicated by the `adsml:ExpirationTime` inside the quotation response message).

AdsML handling: The seller can choose whether to accept the order, or reject it with an `adsml:ReasonForDenial` code indicating that the quotation has expired.

Note: The buyer should not have sent a message that referenced an expired quotation.

6.5.4 Convert a quotation that is not on file

Scenario: A buyer sends an Ad Order message that references a quotation ID that does not exist in the seller's system.

AdsML handling: The seller's system should reject the order and notify its users that this has occurred. The seller should phone the buyer and resolve the discrepancy.

6.5.5 Quote for the "renewal" of an existing ad

Scenario: A buyer wishes to receive a quotation for "renewing" an ad that is currently running or recently ran.

AdsML handling: The buyer submits a quotation, reservation or order request message that includes a text comment or user defined property identifying the order for which this will be a renewal.

6.5.6 Quote for a change or cancellation

Scenario: A buyer wishes to find out what costs would be incurred if they changed or cancelled an existing order.

AdsML handling: This scenario is not supported. The buyer should pick up the phone and ask.

Note: The AdsML Quotation messages should only be used for new orders, not for changes or cancellations.

6.6 Pricing and payments

6.6.1 Tiered package pricing within an order

Scenario: Seller wishes to report the price of a package deal, in which the buyer will receive price breaks based on committing to multiple placements in a single booking.

AdsML handling: The pricing structures can optionally be placed in the AdsML message at three levels: individual pricing for each placement; a subtotal or package price for a group of placements; and a total or package price for all of the placements in the order. In this scenario, the seller can optionally report the individual pricing of each placement, but the final price of the booking is reported by using a pricing structure at either the Placement Group or order header level. If the seller chooses to reveal the pricing details (rather than just reporting a total price), the package discount can be reflected using one or more `adsml:PriceComponent` elements.

Notes:

The highest-level total pricing information that appears in an order message overrides the sums of any sub-total pricing that may appear lower down.

Pricing information at any level can be broken down into `PriceComponents`, and for each `PriceComponent` it is possible to describe the calculation that was used and to reference a relevant rate card entry. If the package pricing is based on a particular contract, that contract can be specified in the relevant `PriceComponent`.

6.6.2 Multiple payers, including co-op splits

Scenario: The ad will be paid for by more than one party, possibly in unequal proportions – for example, a co-op ad.

AdsML handling: Use two or more `PayerInformation` constructs, one for each paying party. In each `PayerInformation`, populate `PayersPriceDetails` with that payer's full pricing information, and optionally use `ShareOfTotalPrice` to indicate their share of the total price.

Note: A single `PayerInformation` is used when there is only one payer.

6.6.3 Contractual pricing

Scenario: The buyer wishes to indicate that this order should be governed by a particular contract.

AdsML handling: The contract details and ID of the contract between the seller and the buyer should be specified in an `adsml:Contract` structure at the relevant level of the message (e.g. the instance of `adsml:Contract` in the order header, placement group or placement).

A contract between the seller and any of the *paying* parties can be described in the `PayerInformation/adsml:Contract` element for that payer. In some cases where the buyer is not the payer it is possible for two or more contracts to apply simultaneously: one between the seller and the buyer, and one with each payer.

If the payment mechanism to be used by a particular payer is defined in a contract, that contract can be referenced in `Payment/Mechanism/adsml:Contract`.

6.6.4 Rate card pricing

Scenario: The buyer wishes to indicate that this order as a whole should be governed by a particular rate card, e.g. the "NY Herald 2007 Rate Card".

AdsML handling: The rate card in question should be identified in adsml:Contract/ContractReference in the order header, placement group header or placement as appropriate. The logic here is that by providing the name of a rate card as the ContractReference, the parties indicate that the "contract" which governs the order is simply the publisher's specified rate card.

Notes:

If the buyer wishes to specify a particular rate from the rate card which governs the order as a whole, this can be done using adsml:Contract/RateReference.

If the order includes PriceComponents, the applicable rate card rate can be specified in each price component. In this case the name of the rate card goes in PriceComponent/RateCardReference, and the rate that was used to calculate the current price component goes in PriceComponent/RateCode.

6.6.5 Discount which spans multiple orders

Scenario: A Buyer wishes to indicate that the price of the booking is governed by an agreement that spans multiple orders, for example, a volume discount if a certain number of ads are booked within a specified period of time.

AdsML handling: Package discounts which span more than one booking are normally established by contract between the paying and selling parties, where the paying party is often (but not necessarily) the same as the buying party. The contract containing the terms governing a particular booking should be specified in an adsml:Contract structure at the relevant level of the message (e.g. order header, placement group or placement). Additional information such as the specific section of that contract which governs this order and the rate that will apply can be conveyed in SectionReference and RateReference, respectively.

6.6.6 Agency commissions or cash discounts

Scenario: A buyer or seller wishes to record an Agency Commission and/or Cash Discount.

AdsML handling: Both Agency Commission and Cash Discount are types of PriceComponents.

Note that the use of adsml:CalculationSpecification is optional. You might give the cash discount as a simple amount without specifying, for example, that it is based on 2% of the order value (or whatever other calculation basis applies).

6.6.7 Tax liability, tax exempt status

Scenario: Buyer's organization has tax-exempt status, and/or needs to convey information about how taxation should be handled.

AdsML handling: Use the adsml:PartyTaxScheme structure in the PayerInformation/adsml:PayerParty construct to specify the taxation status (e.g. "tax exempt organization") and associated government tax identifier for each paying party.

Notes:

- Tax information can be provided for any `Party` to the booking, not just the `PayerParty`.
- Rules and guidelines for populating `adsml:PartyTaxScheme` can be found in the *AdsML Type Library* specification*.*

6.6.8 Identifying a Make-Good Placement

Scenario: A buyer or seller wishes to indicate that there will be no charge for an order, placement or placement group because it is being provided as a make-good for a previous order.

AdsML handling: Use the `CostExempt` element at any level of a message (order, placement group or placement) to indicate that the order, placement group or placement will be provided at no cost to the buyer. The mandatory `ExemptionType` code defines the nature of the exemption, for example "Make Good" or "Charitable Donation". If the placement is a make-good, `PlacementReference` can identify the placement for which this is a make-good, and `ClaimReference` can convey the claim number that this make-good resolves.

Notes:

When a booking is not cost exempt, this element is omitted.

A `CostExempt` booking may also contain pricing information, for example `PriceComponents`, in order to indicate the price that would have applied if the order had not been `CostExempt`.

6.6.9 CPM (Cost Per Thousand) pricing

Scenario: A buyer or seller wishes to record pricing for distribution of inserts or publication of an interactive advertisment on a CPM basis, e.g. "$56 per 1000 pieces" or "$12.00 per 1000 impressions".

AdsML handling: CPM pricing can be described with a textual note in `PriceComponent/DescriptionLine`, and/or by expressing the formula in `PriceComponent/CalculationSpecification`. A CPM calculation specification uses the *`divisor`* attribute of `adsml:PricePerUnit` with a value of `1000`, to indicate that the price should be divided by 1000 when applied to an individual unit.

See the *AdsML Type Library* specification for a more detailed example.

6.6.10 Payment by credit card

Scenario: A party wishes to indicate that this order has been prepaid by credit card.

AdsML handling: The sender populates a `PayerInformation` structure in which the party that prepaid the order is identified as the `adsml:PayerParty`. The amount that was charged to the card is placed in `PayersPriceDetails/TotalPrice` along with any additional pricing details that the sender wishes to convey. The fact that this order has already been paid is indicated by placing an appropriate value (such as "Paid") in `Payment/adsml:Status`. The fact that a credit card was used is indicated by populating `Payment/Mechanism/adsml:CreditCard` with the relevant details of the card that was used.

Note: A similar approach is used when the credit card has been authorized but not yet charged, or when the card will not be processed until after the ad has been published. See the *AdsML Type Library* specification for more details about `adsml:CreditCard`.

6.6.11 Payment by more than one mechanism

Scenario: An order has been (or will be) paid by more than one payment mechanism, for example, the buyer has paid a deposit by credit card with the balance to be invoiced upon publication.

AdsML handling: The sender populates two (or more) instances of `PayerInformation`, one for each of the payments that has been made and at least one representing the balance that is still due. Each instance of `PayerInformation` is fully populated so as to describe one of the past payments or the amount that will be paid in the future.

Notes:

The total amount to be paid by the `PayerParty` is the sum of `PayersPriceDetails/TotalPrice/Amount` in the set of `PayerInformation` structures which reference the payer.

If the unpaid balance of the order will be paid by a single party using a single payment mechanism, then just one `PayerInformation` element should be used to describe that unpaid balance, even if it will result in the issuance of multiple invoices or payments.

6.6.12 Invoicing instructions for a long-running ad

Scenario: An ad will run for six months. The parties wish to specify that the seller will issue an invoice every month, with the amount of each invoice corresponding to the actual number of times the ad was published that month, and the price per publication being calculated on a straight line basis – that is, the cost will be equally allocated across all invoices.

AdsML handling: Invoicing instructions like these are transmitted in `BillingInstructions`, which can be found inside `PayerInformation`.

The `Frequency` code indicates how often invoices should be issued, in this case "Monthly", using values from `AdsMLRecurrencePatternCodeCV`. The Billing `Allocation` code indicates how the total amount of the order should be divided among the invoices, in this case on a "`StraightLine`" basis. And the Billing `Basis` code indicates whether the amount of each invoice should be based on the actual number of times the ad was published that month, or a monthly average, or some other basis; in this case its value is "`Actual`".

Notes:

- A set of recommended `Frequency` code values can be found in `AdsMLRecurrencePatternCodeCV;` recommended `Allocation` code values can be found in `AdsMLBillingAllocationCodeCV`; and recommended Basis code values can be found in AdsMLBillingBasisCodeCV.

- The parties may also transmit textual billing instructions using the appropriate Text elements that can be found in each section of BillingInstructions.

6.7 Credit management

6.7.1 Credit check, credit status

Scenario: Buyer wants to establish creditworthiness with the seller, in order to place an order that will be invoiced later.

AdsML handling: Information relating to the establishment of a business relationship between the buyer and seller should not be included in an AdsMLBookings message. It is assumed either that a business relationship between the two parties already exists, or that the order is being placed on a pre-paid basis.

Note: However, a requirement to validate the credit of an existing customer is a valid reason for rejection or conditional acceptance of an ad order; this information would normally be conveyed in the adsml:Status code of the Ad Order Response or Ad Order Status message.

6.7.2 Credit card information

Scenario: A buyer (for example a consumer placing a classified ad) wishes to pay by credit card.

AdsML handling: Credit card details can be conveyed in the adsml:CreditCard element of the PayerInformation/Payment/Mechanism structure, while the amount charged to the card would go in PayersPriceDetails/TotalPrice.

Note: XML is inherently human-readable, and so sensitive data such as credit card details can potentially be read in transit. It is the responsibility of organizations sending and receiving AdsML messages to ensure that they are suitably protected and/or encrypted while traveling between the sender and the recipient.

6.8 Scheduling

6.8.1 Recurring and "til canceled" scheduling

Scenario: Buyer wishes to schedule a recurring pattern of placements, possibly without providing an end date. For example, "every Friday for ten weeks" or "every weekday until canceled".

AdsML handling: Recurring scheduling is conveyed in the RecurrencePattern element of the InsertionPeriod and Scheduling structures. The pattern, e.g. "every Friday" or "every weekday", is specified using an agreed code or explanatory text string in RecurrencePattern/Code or RecurrencePattern/Text.

A recurrence pattern must also be accompanied by the date range within which the recurrence should occur. This can be expressed as either calendar dates, or a PreDefinedPeriod, or a starting date plus a status of "until cancelled".

Notes:

If the containing InsertionPeriod or Scheduling structure includes an EventCount, or the placement includes a PlacementTarget/EventCount, then publication of the recurring ad will cease as soon as either of those EventCounts has been achieved.

If the recurrence period is "until cancelled" and no EventCount has been provided, the ad will keep running until the buyer sends either an Order Cancellation message or an Order Change message which defines a stopping point for the recurring advertisement. This can take the form of either a LastPossibleTime or EventCount in the containing schedule structure or a PlacementTarget/EventCount for the placement as a whole.

6.8.2 Specifying a "volume" of a publication (newspaper or magazine bookings)

Scenario: Buyer wishes to place an ad in a particular volume of a publication.

AdsML handling: The volume number, or any other string identifying this specific issue of the publication, can be conveyed in the PredefinedPeriod element of the InsertionPeriod structure.

6.8.3 Relative scheduling

Scenario: A buyer wants to specify absolute (non-recurring) dates using a relative format such as "third Friday in November" or "first weekday in October".

AdsML handling: Relative dates are only supported in the context of recurring scheduling as described above. If the dates are known in advance, they must be converted into calendar dates.

6.8.4 Conditional scheduling

Scenario: A buyer wants to indicate that the scheduling of a particular insertion is dependent on an external event.

AdsML handling: Buyer should use the adsml:SpecialRequirements element in the InsertionPeriod and Scheduling structures to specify the fact and nature of the condition, either via an agreed-upon code or an explanatory text string. In both cases, this order will probably end up being routed for manual processing.

6.8.5 Daypart scheduling

Scenario: A buyer wants to schedule a broadcast ad to run during a particular part of the day, such as "morning drive time" or "prime time".

AdsML handling: Buyer should use the RecurrencePattern element in the Scheduling structure to provide one or more codes or textual descriptions of the required dayparts. If multiple dayparts are required, each needs to be placed in its own Scheduling structure.

Notes:

- Because dayparts vary widely by both region and medium the AdsML Framework does not provide a controlled vocabulary for them. Trading Partners wishing to exchange information about dayparts should agree in advance on the specific codes or text strings they will use.
- Multiple codes and/or text strings can be combined within a single `RecurrencePattern` provided that they are not mutually exclusive. For example, the codes "`MorningDriveTime`" and "`Tuesdays`" can coexist inside a `RecurrencePattern` because they have different time scales; together they indicate that the ad should run during morning drive time hours on Tuesdays. However, it is not possible to specify multiple dayparts in a single `RecurrencePattern` because dayparts are mutually exclusive.

NOTE: The AdsML Technical Working Group is considering enhancing the support for Dayparts in a future release of the Framework by creating a dedicated "Dayparts" structure, which would probably be either a child of `Scheduling` or a sibling to it. Implementers are encouraged to provide feedback as to whether you consider this a good idea, and if so, any suggestions or requirements you may have.

6.8.6 Repeated patterns at particular times

Scenario: A buyer requests in an interactive booking that a banner should only be displayed during a specified time period during a day, e.g. '`Wednesdays between 10-12am`'.

AdsML handling: As is often the case, the recommended AdsML handling varies according to how the buyer's and seller's business rules have been defined. In this case, two different approaches are recommended depending on whether the seller has defined and offers a specific slot with an associated price in the rate card for the 10-12 period, or whether the timing request is an additional requirement from the buyer, in addition to any standard terms for the banner product in question.

Dates and time for the display of the advertisement are specified in the `Scheduling` element (a.k.a `InsertionPeriod` in some media). The dates and times can be recorded by specifying precise date-time values for each appearance, or by specifying the start and end date-time of the booked period and using code(s) to specify the recurrence pattern of 'on Wednesday, 10-12am'.

For situations where the publisher does not offer a suitable predefined time slot in their rate card, the recommended solution is to use a separate `Scheduling` element for each publication event within the intended period. The `FirstPossibleTime` and `LastPossibleTime` child elements of `Scheduling` are used to explicitly set the dates and times for all of the Wednesdays when the ad will be published, as shown below.

```
<Scheduling>
  <ScheduleEntryIdentifier>myco.com:2011-05-01:123-1</ScheduleEntryIdentifier>
  <FirstPossibleTime>2011-05-04T10:00:00+01:00</FirstPossibleTime>
  <LastPossibleTime>2011-05-04T12:00:00+01:00</LastPossibleTime>
</ Scheduling >
<Scheduling>
  <ScheduleEntryIdentifier>myco.com:2011-05-01:123-2</ScheduleEntryIdentifier>
  <FirstPossibleTime>2011-05-11T10:00:00+01:00</FirstPossibleTime>
  <LastPossibleTime>2011-05-11T12:00:00+01:00</LastPossibleTime>
</ Scheduling >
<Scheduling>
  <ScheduleEntryIdentifier>myco.com:2011-05-01:123-3</ScheduleEntryIdentifier>
  <FirstPossibleTime>2011-05-18T10:00:00+01:00</FirstPossibleTime>
  <LastPossibleTime>2011-05-18T12:00:00+01:00</LastPossibleTime>
</ Scheduling >
<Scheduling>
  <ScheduleEntryIdentifier>myco.com:2011-05-01:123-4</ScheduleEntryIdentifier>
  <FirstPossibleTime>2011-05-25T10:00:00+01:00</FirstPossibleTime>
  <LastPossibleTime>2011-05-25T12:00:00+01:00</LastPossibleTime>
</ Scheduling >
```

It is also possible to define an `EventCount` in each `Scheduling` element, which specifies the number of impressions during the given period. This model is used for ‘throttling’ the appearance of an ad, which is described above, see “*Defining an online placement: Placement Target, Distribution, Capping, Throttling and Share of Voice*”.

For the other case – when the publisher has defined and offers a special product option, represented as a code for the 10-12 time slot – the recommended solution is to use a single `Scheduling` element with a child `RecurrencePattern` element to define the repetition within the overall timeframe given by `FirstPossibleTime` and `LastPossibleTime`. For instance, if the publisher’s code value ‘`LateMorning`’ represents the time slot 10-12, and the AdsML CV code ‘`Wednesdays`’ the day in week, then the following example expresses the scheduling requirement of ‘`Wednesdays 10-12am during May 2011`’using two codes:

```
<Scheduling>
  <ScheduleEntryIdentifier>myco.com:2011-05-01:123</ScheduleEntryIdentifier>
  <FirstPossibleTime>2011-05-01T00:00:00+01:00</FirstPossibleTime>
  <LastPossibleTime>2011-05-31T00:00:00+01:00</LastPossibleTime>
  <RecurrencePattern>
    <adsml:Code>
      <adsml:CodeList>AdsMLRecurrencePatternCodeCV</adsml:CodeList>
      <adsml:CodeValue xsi:type="adsml-
cv:AdsMLRecurrencePatternCodeCV">Wednesdays</adsml:CodeValue>
      <adsml:Description>Occurs on Wednesdays</adsml:Description>
    </adsml:Code>
    <adsml:Code>
      <adsml:CodeList>LocalReccurencePatterCodeCV</adsml:CodeList>
      <adsml:CodeValue>LateMorning</adsml:CodeValue>
    </adsml:Code>
  </RecurrencePattern>
</ Scheduling >
```

Notes:

- `FirstPossibleTime` and `LastPossibleTime` cannot be used to capture time points only. The given time points are always related to a particular date.
- For more information about the use of codes to specify recurrence patterns see also the Use Case “*Order a Broadcast or Cable Ad*”.

6.9 Distribution (Regions, Zones, Editions, Demographics, etc.)

6.9.1 Multiple zones or regions

Scenario: A buyer wishes to distribute an ad only to one or more "zones" or "regions".

AdsML handling: Use the `DistributionTarget` structure to define the distribution. Place the code values for the desired zones or regions in a set of `Code` elements. If no other types of targeting information (e.g. editions or demographics) are specified, put all the zone or region codes in a single `Target` element. Otherwise, follow the instructions in the *Distribution* section of this document to determine how to use the `Targeting` structures to achieve your desired goals.

Notes:

- It is also possible to indicate that the advertisement should <u>not</u> be distributed to certain zones or regions by including codes for those zones or regions in the message but setting their *`negated`* attribute to "`true`".
- See the *Distribution* section of this document, above, for more detailed descriptions about how to convey different types of targeting instructions.

6.9.2 "Zoned" ads

Scenario: Buyer wishes to place an order for multiple versions of an ad, each of which will run in the same publication at the same dates and times but in a different set of zones or regions.

AdsML handling: This situation is handled by specifying multiple `Placements` inside the Order, where the pricing will apply to the order as a whole. In each `Placement` the buyer identifies a set of artwork and the locations in which it should appear (`DistributionTarget`). Information common to all the placements, such as the publication name and insertion dates, is redundantly copied into each of them.

6.9.3 Specific Demographic targets

Scenario: A buyer wishes to distribute an ad only to customers meeting certain demographic criteria, for example, age or household income ranges.

AdsML handling: Use the `DistributionTarget` structure to specify the demographic information. Place code values for the desired demographic targets in a set of `Code` elements. If no other types of targeting information (e.g. regions or editions) are specified, wrap all the demographic codes in a single `Target` element. Otherwise, follow the instructions in the *Distribution* section of this document to determine how to use the `Targeting` structures to achieve your desired goals.

Notes:

- It is also possible to indicate that the advertisement should <u>not</u> be distributed to certain demographic groups by including codes for those groups in the message but setting their *`negated`* attribute to "`true`".

- See the *Distribution* section of this document, above, for more detailed descriptions about how to convey different types of instructions.

6.10 Inserts

6.10.1 Book an insert

Scenario: A buyer wishes to order a set of Inserts.

AdsML handling: Most of the information is provided in the same way as for a newspaper or magazine classified or display ad. The main differences are: use `Placement.Insert` (rather than `Placement.NewspaperMagazine`) to provide the placement information. The type of insert being booked should be identified in `adsml:AdType`. Use `Targeting` elements inside `DistributionTarget` to specify the desired distribution pattern (i.e. regions/zones) as well as the target distribution counts, and optionally use `ProductionDetail.Insert` to describe count, weight, thickness, etc. of each insert. Use `Pickup.Insert` if you wish to re-use inserts left over from a previous order, and use `MaterialsExpectations.Insert` to provide information about the physical delivery of the inserts to the printing location(s), including the date on which they are expected to arrive and any relevant text notes.

6.10.2 Specify distribution targets

Scenario: A buyer of Inserts wishes to specify the distribution targets (i.e. number of copies) that they wish to distribute via a specified distribution pattern.

AdsML handling:

Use one or more `Targeting` elements in `DistributionTarget` to specify the desired target set (regions, zones, editions, demographics, distribution methods, etc.). To indicate a distribution target for a particular set of regions, zones or editions, place a `DistributionCount` element inside the `Target` or `Targeting` structure which defines those regions, zones or editions. To indicate a target for the entire distribution, use the `TotalDistributionCount` element.

Notes:

- The primary target for the placement as a whole should be conveyed in `PlacementTarget/EventCount`. In most cases this will be the same as `TotalDistributionCount`.
- The buyer and seller should agree in advance on the purpose and use of distribution counts.
- See the *Distribution* section of this document, above, for more detailed descriptions about how to convey different types of targeting instructions.

6.11 Ad Content

6.11.1 "Pick-up" (re-use) previously delivered ad materials

Scenario: The buyer wishes to place a booking which re-uses previously delivered ad materials.

AdsML handling: Use the PickUp element to tell the publisher where to find the ad content that you want to re-use for this placement. Specify another placement that used the ad contents that you wish to pick-up in PlacementReference, and optionally provide a unique identifier for those ad contents in MaterialsReference. If you want the publisher to change the materials in any way before re-using them, set the withChange attribute to true and describe the necessary changes in the adsml:Instructions element.

Note: This mechanism can be used for any type of ad materials, including both textual and graphical (display) ads.

6.11.2 Convey information about the future delivery of ad materials

Scenario: The buyer wishes to tell the seller some potentially useful information about the materials which will later be delivered for a placement, for example, the name of the company that is expected to prepare and deliver them, or the ID with which the artwork will be labeled.

AdsML handling: Use adsml-ma:MaterialsExpectations to provide information about the materials that will be associated with this placement. The Materials Expectation structure can be thought of as containing two types of information: information from the buyer to the seller, and information from the seller to the buyer. The first set of elements (materials identifiers, provider party, available date/time, retrieval address and retrieval instructions) conveys information that is potentially provided by the buyer in order to facilitate the seller's production and copy chasing activities. The second set of elements (recipient party, due date/time, delivery address, delivery instructions and labeling instructions, size and tech spec details) conveys information that is potentially provided by the seller in their bookings Response message in order to tell the buyer where and when they should deliver the artwork and to describe the publisher's technical requirements.

Notes:

- The adsml-ma:AdContent element is used to describe or convey materials that are being delivered now, for which no future delivery will occur. Therefore, adsml-ma:AdContent should not be used in this situation, and similarly, adsml-ma:MaterialsExpectations should not be used when in situations when adsml-ma:AdContent would be more appropriate.
- The explanation above about the two sets of elements inside adsml-ma:MaterialsExpectations is explanatory rather than normative; there is no restriction on which party can populate each specific element.

6.11.3 Deliver the text of a classified lineage ad

Scenario: The buyer wishes to transmit both a booking for a lineage ad and the ad copy of that ad in the same message.

AdsML handling: The text of a classified ad can be conveyed in the `adsml-ma:AdContent` element in the `Placement` structure. If it is a lineage ad, then the text of the ad, including any formatting characters that have been agreed with the publisher, is put in the `adsml:ContentData` element inside `adsml-ma:Rendering`. (This is the required location for any ad content that is conveyed in-line inside a Bookings message.) If your system does not normally generate a unique identifier for the text of a lineage ad, fill in the required `adsml-ma:RenderingIdentifier` with a copy of the `PlacementIdentifier`.

A copy of the unformatted text of the ad can also be placed in `adsml-ma:AdContent/adsml-ma:AdContentText.`

If the parties have agreed to use the AdsML Structured Descriptions standard, then a version of the ad's content conforming to the Structured Descriptions standard is placed inside the `adsml-ma:StructuredDescriptions` element.

Notes:

- The `adsml-ma:AdContent` structure is provided in a Bookings message primarily for the purpose of conveying classified lineage ads between systems that are configured for buying and selling classified ads. However, this element is also capable of conveying any form of ad content, and it can do so either in-line in the message or by referencing an external delivery mechanism. In order to accommodate all of these possibilities, `adsml-ma:AdContent` includes a full subset of the structures found in an AM-M materials delivery message. These capabilities and structures can largely be ignored when using `adsml-ma:AdContent` to convey the text of a lineage ad.
- In particular, the `Delivery` structure is not used when conveying ad content inside a Bookings message, because the content is "delivered" by being included in the `adsml:ContentData` element.
- The `adsml-ma:AdContentText` element is primarily intended to support cases where the ad contents consist of digital artwork from which the text has been extracted and needs to be conveyed separately. In a lineage ad the text is already available inside the `adsml-ma:Rendering`, so there is usually no reason also to populate the `adsml-ma:AdContentText` element.

6.11.4 Multiple placements for a single set of ad materials that is transmitted as part of the booking message

Scenario: The buyer wishes to use the same set of in-line ad materials for multiple placements in a booking.

AdsML handling: Deliver the ad materials in the `adsml-ma:AdContent` element of the first placement. For each subsequent placement, use the `PickUp` element to indicate that the materials should be re-used from another placement, and in its `PlacementReference`, provide the ID of the placement in this booking in which you put the materials.

6.11.5 Order an Interactive 'banner' ad with artwork in multiple parts

Scenario: A buyer orders an interactive 'banner' ad in an online publication. The 'banner' is published in an 'L'-shaped area. For production the material is provided to the publisher as two pieces of digital artwork – one for the vertical part'|' of the 'L' and one for the horizontal part '_' of the 'L'.

AdsML handling: The recommended handling depends on whether the publisher defines and sells the L-shaped banner product as two separate placements on the page, e.g. 'LVertical' and 'LHorizontal' areas, or as a single placement representing the complete L-shaped area.

For the first case of two individual placements, the order should be booked using two Placement.Interactive elements referencing 'LVertical' and 'LHorizontal' placement codes respectively. Each Placement will reference one piece of artwork, for the vertical and horizontal parts, using the adsml-ma:AdContent element in each of the placements to do this. A LinkedPlacement element is used to express a 'PublishTogether' relationship between the two placements, ensuring that the digital artworks making up the ad are always displayed to users together and at the same time. The 'PublishTogether' requirement is recorded as a code in the PlacementLink/adsml:RelationshipName element located at the top level of the bookings message.

For the second case, with a single placement representing the whole L-shaped banner product, a single Placement.Interactive structure should be used. But as the two parts of the digital artwork do not represent material ready for production (at least they would have to be assembled into an 'L' in the production phase), they are represented using the repeatable adsml-ma:Component structure within the adsml-ma:AdContent element. The adsml-ma: ComponentIdentifier and adsml-ma:AuxiliaryComponentReferences should be used to record references for the artwork components to facilitate materials-to bookings-matching.

See section "*Linked placements*" above for further information about linked placements.

Notes

- The adsml-ma:AdContent structure holds the complete materials information describing the artwork for the placement. It can include material ready for production using the adsml-ma:Rendering element, and/or material components that should be processed before publication using the adsml-ma:Component element.
- The machine readable instructions in the LinkedPlacement structure may be complemented with human readable instructions in the ProductionDetail.Interactive/adsml:Specifications element.

6.12 Status requests and reports

6.12.1 Duplicate copies of the complete booking

Scenario: Buyer or seller wishes to send a full copy of the booking information to a third party.

AdsML handling: Either party can spontaneously send one or more order status messages (AD-OS) to any other parties with whom they have a business relationship such that it is appropriate for these other parties to see a copy of the booking information.

Notes:

- An AD-OS status message cannot be sent until after the seller's initial business response message (AD-OR) has been sent to the buyer.
- Only status messages may be used for this purpose. All other order-related messages should be sent only between the two parties that are the primary participants in the transaction.

6.12.2 Request status as of a specific date

Scenario: The buyer wishes to request the status of an order as of a particular date in the past.

AdsML handling: This is not supported. A status request is always for the "current" status.

Note: There may be a delay before the seller is able to respond. In all cases, the status reported in a Reservation or Order Status message should reflect that message's `adsml:BusinessMessageDate`, which is not necessarily the time at which the status request was made.

6.12.3 Snapshot of current status of a reservation or order

Scenario: The seller wishes to provide a snapshot of the current status of an order, for example, whether it has been accepted, or is on hold pending resolution of some issue, etc.

AdsML handling: The seller sends an Ad Order Status (or Ad Reservation Status) message which contains the full details of the order, and uses the `adsml:Status` element to convey as many status codes and/or descriptions as necessary. The `adsml:BusinessMessageDate` indicates the "as of" date/time at which this status information was accurate.

Note: The buyer can request the status of an order or reservation by sending an Ad Order (or Reservation) Status Enquiry message.

6.12.4 Reporting separate statuses in a multi-placement order

Scenario: The seller wishes to report status on an order that contains multiple placements and/or insertion dates, which may have different statuses from each other (some published, others awaiting arrival of artwork, etc.).

AdsML handling: The seller should use the status code at the placement level (element `adsml:Status` at the top level of `Placement`) to indicate the overall status of each placement, and the status inside the `InsertionPeriod` (in `Placement.NewspaperMagazine` and `Placement.Insert`) or `Scheduling` (in `Placement.Interactive` and `Placement.Generic`) elements to indicate the status of each individual insertion.

Note:

- In general, a status at a lower level of granularity (e.g. the insertion period) should be more precise than the status at the higher level (e.g. the placement).
- In order to avoid confusion, in bookings with mixed status, users are encouraged to use a higher-level `adsml:Status` code with a value like "Partly published" to indicate that the placement or order as a whole has a mixed status and that more detailed information can be found at a lower level.

6.12.5 Reporting status of a date range order

Scenario: The seller wishes to report status on one or more appearances of an ad that was booked by specifying several appearances over a date/time range rather than individual date/times.

AdsML handling: The seller should restructure the information so as to report each published date/time (or date range, in interactive and broadcast media) as a separate insertion period. For example, if the original order was for 5 insertions within a date range in a single `InsertionPeriod` element, and two of those insertions have now been published, then the status message should include 3 `InsertionPeriods`: two `InsertionPeriods` each with a single date and a status of "published", and a third `InsertionPeriod` reflecting the remaining three insertions that have not yet been published.

Notes:

- The solution described above, involving 3 insertion periods with differing levels of granularity, is illustrative rather than normative. Users are free to restructure the insertion information in a status message in any way which allows them to communicate the necessary level of granularity. As always, trading partners should agree in advance as to how insertion information will be communicated in orders, responses and status messages.
- This concept applies to any situation in which the seller wishes to report different statuses for the insertions within a single insertion period.

6.12.6 Report what was actually published (aka a "pre invoice")

Scenario: The seller wishes to provide information about the actual results of an order, i.e. which ads were published, where they appeared, estimated price, etc. This information precedes the delivery of a formal invoice and proof of publication or performance, which will come separately.

AdsML handling: The seller sends an Ad Order Status message containing the updated information, with a status code indicating the ad has been published. As always, the status message should contain the full details of the order. In case the order contained multiple placements not all of which have been published yet, all placements in the message on or dated before the `adsml:BusinessMessageDate` are considered reports of the “actual” publication, while all placement information dated after the `adsml:BusinessMessageDate` reflects pending bookings that have not yet been published. The status values of these placements should have status values indicating that their publication is “pending”.

Notes:

- In cases where the seller is an external agency acting on behalf of one or more publishers, this approach can also be used by a publishers to send “actual” results information to the seller, who in turn will pass it on to the buyer.
- Unless the trading parties explicitly agree otherwise, pricing information contained in an AdsML Booking message should be considered an estimate only. It is intended to support various processes such as invoice reconciliation, but it should not be considered a replacement for the invoice itself.

6.12.7 Status of artwork delivery and processing

Scenario: The buyer and seller wish to exchange messages that are specifically about the ad content materials – did it arrive, has it been flight-checked, etc.

AdsML handling: In general, materials management information should not be included in AdsML booking messages. There is a separate set of AdsML e-commerce messages covering artwork delivery and receipt, which will usually be exchanged by a different set of software applications.

Note: Some of this information can be reflected in the status of a placement or insertion: for example, the seller could report “pending artwork” as a valid order status, which could later be changed to “ready to publish” or “published”. The usefulness of such artwork-related statuses will depend entirely on whether the relevant information is available to the seller’s ad booking system, and how often the trading partners plan to exchange order status messages.

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. (The selection of Quotations, Reservations and/or Order messages falls into this category.)

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 Booking types

There are three available types of bookings:

- Quotations
- Reservations
- Orders

Trading partners must agree on which combination of Quotations, Reservations and/or Orders they will exchange.

7.2 Message exchange mode

There are three defined message exchange modes:

1. Full Request-Response
2. Datagram model from buyer to seller
3. Datagram model from seller to buyer

Trading partners must select one of these exchange modes.

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

7.3 Status messages

For reservations and orders there are two non-exclusive patterns available for the exchange of status messages:

- Status messages may be sent spontaneously by the seller in broadcast fashion
- Status messages may be sent by the seller in response to a status enquiry message from the buyer

For each supported booking type, trading partners must decide whether to exchange status messages, and if so, which of these two approaches will be supported.

7.4 Changes

7.4.1 Buyer-initiated changes

Trading partners must agree on whether or not they support the use of AdsML messages to change a previously accepted booking.

7.4.2 Seller-initiated changes

Trading partners must agree on whether or not they support an AdsML message flow in which changes to a booking can be initiated by the seller rather than the buyer.

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

7.4.3 Use of change location pointers

Trading partners must agree on whether or not they support the use of `ChangeLocationReference` ID pointers in `adsml:ChangeSpecification` to indicate which parts of the information in the message have changed.

Note that it is always possible to include a code or text string to communicate this information.

7.5 Cancellations

7.5.1 Buyer-initiated cancellations

Trading partners must agree on whether and under what circumstances they support the use of AdsML messages to cancel a previously-accepted booking.

7.5.2 Seller-initiated cancellations

Trading partners must agree on whether or not they support an AdsML message flow in which a booking can be canceled by the seller rather than the buyer.

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

7.6 Acceptance types

Besides "accept as-is", there are two available supplemental types of acceptance:

- Accept "with clarifications"
- Accept "with changes"

Trading partners must agree on which supplemental acceptance types (if any) they will support.

7.7 Ad materials

7.7.1 Conveyance of ad materials – by inline or external transmission

Trading partners must agree as to whether and how they support the exchange of ad materials in a placement:

- 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 in the `adsml-ma:Delivery` element in `adsml-ma:AdContent`..

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 package, etc.

7.7.2 Conveyance of ad materials – by in-line transmission

There are three possible levels of support for in-line transmission of ad materials in a placement:

1. Unsupported – no ad materials may be transmitted in-line in a placement
2. Only textual ad materials may be transmitted in-line in a placement
3. Both textual and binary materials may be transmitted in-line in a placement

Trading partners must select one of these levels of in-line content transmission.

Note that this selection does not restrict the ability to convey ad materials at the same time as a booking via an external reference, i.e. a simultaneous delivery.

7.7.3 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
- Encryption of inline content

7.8 Prices and Payments

7.8.1 Pricing levels

There are three levels of a booking at which pricing and payer information can appear:

- Placement (PlacementPrice)
- Placement Group (PlacementGroupPrice)
- Booking (TotalBookingPrice)

Trading partners must agree on which of these levels (if any) can contain pricing and payer information.

7.8.2 Pricing currencies

Trading partners must agree on whether a currency that is different from the booking’s Document Currency Code may be specified for a particular placement or placement group.

7.8.3 Multiple payers or payment methods

Trading partners must agree on whether a booking can specify multiple payers or payment methods (by having more than one `PayerInformation` element at the same level of the booking).

7.8.4 Payment currencies

Trading partners must agree on whether a currency that is different from the booking’s Document Currency Code may be specified for a particular payment (in `PayerInformation/PayersPriceDetails`).

7.9 Multiple publications

Trading partners must agree on whether a placement can specify multiple publications, and if so, on the guidelines they will follow for deciding whether to split it into multiple placements in subsequent messages.

7.10 Linked placements

Trading partners must agree on whether or not they support the use of `LinkedPlacements` to indicate that two or more placements are related to each other.

7.11 Placements per booking

There are three hierarchical levels of placements per booking, in ascending order of complexity:

1. One placement per booking, no placement groups
2. One or more placements per booking, but no placement groups
3. One or more placements per booking, any subset of which may be grouped into one or more placement groups

Trading partners must select one of these levels of placements-per-booking.

7.12 Dates per placement

Trading partners must agree in advance as to how many insertion or appearance dates may be associated with a single placement, and which mechanisms may be used to express those dates. The questions to consider are:

- Is use of `FirstPossibleTime` and `LastPossibleTime` supported?
    - If so, are they required to have the same value or may they express a range of dates?
- Is use of `RecurrencePeriod` supported?
- Is use of `PreDefinedPeriod` supported?
- How many instances of `InsertionPeriod` or `Scheduling` may be associated with a single placement?

7.13 Placement types

There are four available types of placements, each of which is represented by an extension of the Placement element:

- Placement.Generic
- Placement.Insert
- Placement.Interactive
- Placement.NewspaperMagazine

Trading partners must agree on which types of placements they will support in a booking message.

7.14 Guarantees

Trading partners must agree on which parts of an order (if any) can be "guaranteed", and if so, what codes will be used to convey such guarantees and what the business meaning is of each such code.

7.15 Share of Voice (interactive)

Trading partners wishing to use the Share of Voice mechanism should agree in advance on the meaning they will apply to it, and in particular on whether and how Share of Voice should coexist with other key components of an order such as Placement Target, Capping and Throttling.

7.16 Multilingual metadata

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

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

- Which languages will they use in their messages?

- Which language, if any, takes priority as the ‘default language’ of the message?
- Any rules for processing and presenting multilingual content to users.

8 Appendix A: Acknowledgment for contributions to this document

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:

- John Iobst (NAA Technical Solutions)
- Joe Kirk (K Media Solutions)
- Beat Doerr (Publigroup)
- Howard Robson (5 fifteen)
- Israel Viente (Vio Worldwide Limited)
- Christian Ratenburg (CCI Europe)
- Alan Leverenz (Vio AdSend)
- Matthew Leete (Associated Newspapers Ltd)

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

- The SpaceXML Working Group (IDEAlliance)
- Peter Meek (Miles33)