AdsML® Framework for E-Commerce Business Standards for Advertising

AdsMLMediaPack 1.0.0
Part 1
Usage Rules & Guidelines

Document Authors: AdsML Technical Working Group

Document ID: AdsMLMediaPack-1.0.0-SpecP1Usage-PS-2

Document File Name: AdsMLMediaPack-1.0-SpecP1Usage-PS.pdf

Document Status: Proposed Specification

Document Date: 15 April 2010

Draft Number: 2

Table of Contents

1 AdsMLMediaPack Standard Documentation

1.1 Document status and copyright

This is a Proposed Specification of the *AdsMLMediaPack 1.0 Part 1 Usage Rules & Guidelines*.

This is an AdsML Proposed Specification for review by AdsML Members and other interested parties. It is a draft document and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use AdsML Proposed Specifications as reference material or to cite them as other than "work in progress".

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

Copyright © 2010 AdsML Consortium. All rights reserved.

Copyright Acknowledgements: The AdsML Non-Exclusive License Agreement is based 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 AdsMLMediaPack-1.0.0-SpecP1Usage-PS-2, is freely available. It will be located at the AdsML website at http://www.adsml.org/.

1.5 Purpose of this document

This document provides rules and guidelines for how to use the messages defined in the AdsMLMediaPack standard. AdsMLMediaPack 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 AdsMLMediaPack 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 AdsMLMediaPack messages to address specific business requirements. A companion document,

AdsMLMediaPack – Part 2 - Specification & Schema, serves as the reference guide to the AdsMLMediaPack 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. AdsMLMediaPack 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
1.0 PS-2	15 April 2009	First Proposed Specification version • Updated the Data Overview diagram • Added diagrams of the message exchanges • Added rate and pricing usage information • Added Statistics usage information • Added Usage overview • Added FAQs • Added Configuration Checklist	TS/UW
1.0 WD -1	30 June 2009	First Working Draft version	UW

1.10 Acknowledgments

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

Primary authorship and editing was performed by,

• Ulf Wingstedt (CNet Svenska AB) ulf.wingstedt@cnet.se

• Tony Stewart (RivCom) – tony.stewart@rivcom.com

The AdsML Mediapack specification has been developed based on a proposal from Kärkimedia Oy, Finland.

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.

1.12 To do in future releases

The following topics have been proposed for future releases of this document:

- Provide an FAQ describing how use the TermsAndConditions element to convey both textual and PDF versions of the seller's Terms and Conditions.
- Provide an FAQ about publishing circulation and demographic information associated with a publication or edition (e.g. Statistics at the Rate Group level)
- Add further descriptions of usage scenarios including both human-oriented and machine processable rate qualifiers, especially for percentage values.
- Provide an FAQ about pricing for a special event like Olympics coverage. This could be handled either by a rate group or a special rate card.

2 Introduction

The AdsMLMediaPack standard has been developed by the AdsML Consortium to provide a global standard for the exchange of advertising media packs, including rate cards and technical specifications for advertising offers. It relies on earlier experience and practices that have been embraced and extended in order to support current advertising business requirements. In addition, AdsMLMediaPack 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.

A media pack can be divided into two main parts: the *Rate Card*, which lists prices for particular types of ads (based on such differentiators as the ad's size, scheduling, positioning and technical format as well as buying volume and frequency); and *Technical Specifications* which describe the requirements for artwork production and delivery that are associated with a given publication.

The Technical Specification data can further be divided into two categories, depending on the intended user:

- Detailed, production-oriented technical specifications are used by production companies in order to prepare ad materials that meet the requirements of the intended publications

- Higher level technical specifications are used by parties in the buying/selling process in order to qualify the published rates; for example, to indicate that a rate only applies to ads of a particular size or size range.

The current release of the AdsMLMediaPack specification supports the exchange of *Rate Card* messages. These can convey a lightweight set of technical specifications in order to support the media buying workflow as noted above.

This release of AdsMLMediaPack does not support the exchange of detailed, production-oriented *Technical Specification* messages. It is anticipated that support for such messages will be added in a later release of the standard.

Automated vs. manual workflows

Current practices in exchanging media pack information are entirely manual. PDF brochures and Excel spreadsheets can typically be downloaded from the sellers' web sites or are distributed by e-mail to media buyers that, with much manual labor, adapt the data to the structures of their IT buying systems. The current human-oriented media pack is also often oriented towards marketing, an aspect that is difficult to capture in machine-machine communication.

It is thus our understanding that human and machine oriented media packs will exist in parallel, and that they need to coexist. In a longer perspective, human readable media packs can be even more oriented towards marketing, while the detailed price lists and terms can be left to the computer to digest.

Existing rate cards often include complex rules and exceptions relating to terms for surcharges and discounts. Although the AdsMLMediaPack specification includes general mechanisms for expressing such terms, it should be noted that such information would not generally be machine-processable without prior agreement between trading partners. It is thus expected that media pack messages in general will use free text notes fields to record business rules and constraints to a larger extent than messages in other AdsML supported workflows do.

Supported choreography

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

This release of AdsMLMediaPack supports the "Pricing & Demographics" business messages defined in the ACIA, which are commonly referred to as a "rate card".

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

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

But it should nevertheless be noted that as the AdsML Envelope has been particularly developed to support message transfer within the advertising business and it is RECOMMENDED for use with the AdsMLMediaPack message format.

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

3 Business Messages Overview

3.1 Supported messages

AdsMLMediaPack 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, AdsMLMediaPack defines the set of business messages that belong to the media pack group (MP).

The complete list of business messages supported by this release of AdsMLMediaPack is:

Message Code	Message Name
MP-PR	Media Pack Pricing & Demographics Request
MP-P	Media Pack Pricing & Demographics
MP-X	Media Pack Withdrawal

3.2 Data overview

The figure below provides a diagrammatic overview of the potential information in a Media Pack Pricing & Demographics message (aka "Rate Card").

This view omits the generic message header as well as many of the smaller details, in order to see the main context-specific information "at a glance". Much of the information 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.

**Overview of the potential information
in an AdsML Rate Card message**

(AdsMLMediapack 1.0 April 2010)

Rate Card
Rate card identifier
Name, Issue date
Validity period, Status
Valid for:
    Customer category,
    Booking *Party*, Advertiser
    Party, Contract
Selling *Party*, Other *Parties*
Currency code
Terms and Conditions
Notes, Disclaimer text
Replacement reference,
    Change specification
Document rendering
Publications
Additional Services
*General Charges
 and Discounts*
Rate Groups

**General Charges
or Discounts**
Identifiers
Rate code
Amount or Percent
Description line
Qualifiers:
    Type, Code, Description

Publication
Identifier
Name, Code, Media type
Descriptive information
Languages
Parent *Publication*
Publisher *Party*, *Contacts*
Publishing schedule
    Dates, Periods, Recurrence
Statistics

Statistic
Statistic type
Scope: Geographic,
    Demographic,
    Temporal
Description
Count
Data period
Data source
Sub-statistics
    Type, Scope
    Count

Rate Groups
Identifiers
Name, Description
Publication references
Publishing schedule
    Dates, Periods, Recurrence
Languages
Terms and Conditions
Technical specifications
Notes
Charges and Discounts
Statistics
Rates

**Charges or
Discounts**
Identifiers
Rate code
Amount or Percent
Description line
Rate Qualifiers

Party
Identifiers
Name
Addresses
Contacts
Related *Parties*
Taxation details

Contact
Role
Name
Phone numbers
Addresses
Email addresses

**Additional
Services**
Service code
Name,
Description
Price per Unit
Notes

Rates
Identifiers
Code, Description
Price per unit (repeated)
Price is for "Each" or "All"
Ad type, Notes
Rate qualifiers

Rate Qualifiers
General
Publication references
Sizes, Durations, Distribution
Sections, Positioning codes
Frequency, Scheduling
Buying volume
General qualifiers

Interactive
Technical ad format

Inserts
Weights
Number of pages
Thicknesses

Print
Colors
Positions on Page
Placement s in Book

3.3 Content model

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

3.3.1   Rate Card

The top level of the RateCard messages includes basic mandatory elements
specifying the issue date and overall validity period as well as the selling party,
i.e. the party that is providing the advertising products for the rates listed in the
rate card.

All Rates are defined within a `RateGroup` wrapper. The `RateGroup` groups rates that belong to the same `Publications`, follow the same `PublishingSchedule` and have the same technical specifications.

The `RateGroup` and `Rate` structures are quite rich in order to support both simple cases (such as price/mm for a classified print ad or price per 20 second spot in a nationwide TV program) and complex ones (such as a multi-media combo offer which provides a single price for an ad in a newspaper, an ad in the newspaper's weekly magazine supplement and featured appearance on the newspaper's website).

All details that differentiate one `Rate` from another are described in the `RateQualifier` structures, e.g. `Size`, `Color` and `Positioning`. The `RateQualifier` exists in both media-agnostic and media specific versions.

Technical specifications are provided at the `RateGroup` level and are not directly connected to `Publications`. This is due to the fact that a print publication typically has different tech specs for different sections. For instance, editorial and classified sections in a newspaper can have different numbers of columns and different production or deadline requirements.

In a typical rate card there are many cross references between rates, rate groups and publications. In order to minimize the need for repeated data in messages, all publications are defined at the top level of the rate card using the `Publication` element and then referenced from other sections by `PublicationReference` element pointers.

Rates for `AdditionalServices` can be described at the top level of the `Ratecard`. General charges and discounts can also be expressed there, for instance Agency Commissions or premium fees.

`TermsAndConditions` includes sub-structures for terms specifically relating to Bookings, Cancellations, Claims and Payments, as well as elements to contain more generic terms and conditions. These structures are all simple strings intended to hold text for human consumption. Unlike AdsMLFinancials, it was not considered critical to provide machine-readable data structures in this area of the rate card.

4 Message Choreography

This is a normative section describing the expected message flow between communications partners in the exchange of media pack information.

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

AdsMLMediaPack includes two main categories of messages:

- Business messages, i.e. messages such as rate cards 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 AdsMLMediaPack 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 AdsMLMediaPack 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 AdsMLMediaPack 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 Media Pack 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 AdsMLMediaPack **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, a rate card request, a rate card, or a rate card withdrawal. AdsMLMediaPack supports a subset of business messages as defined in *AdsML Advertising Component Interactions Analysis*, namely messages from the Mediapack group (MP).

This release of the AdsMLMediaPack specification supports the following message transactions between advertising buyers and sellers, or agents acting on their behalf:

- A Media Pack Request is sent by a buyer to a seller.

- A Media Pack is sent by a seller to a buyer, either spontaneously (in broadcast fashion) or as a result of a Media Pack Request message that has been received from that buyer.

- A Media Pack Withdrawal is sent by a seller to a buyer in order to withdraw a previously issued rate card.

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

The sections below give a summary of the messages that have been implemented. For more information, see also the reference section for each message element (named as the message name in `CamelCase`).

4.4.1   Rate Card Request and Response

The buyer requests a rate card, and the seller responds to that request by providing one.

4.4.1.1 *Datagram messaging from buyer to seller*

1) The buyer requests a rate card by sending a Media Pack Pricing & Demographics Request message (MP-PR), after which the seller provides the requested information by non-AdsML means (i.e. mail or fax). This ends the message transaction.
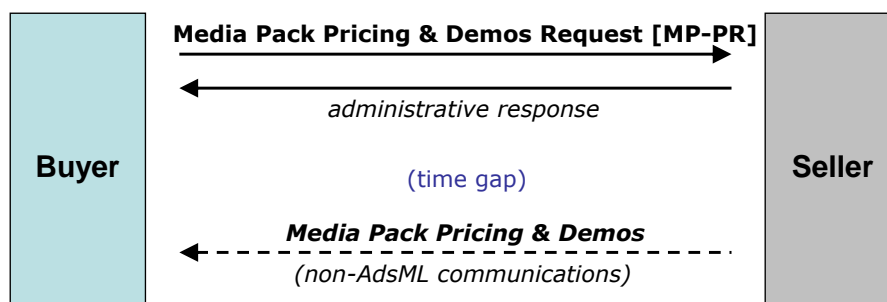
Media pack request datagram message from buyer to seller

4.4.1.2   Datagram messaging from seller to buyer

1) The buyer requests a rate card by a non-AdsML means such as a fax or phone call. The seller sends a Media Pack Pricing & Demographics (MP-P) message in response to that request. This ends the message transaction.
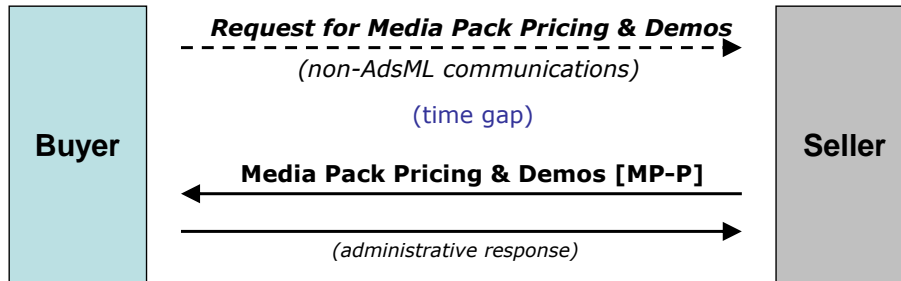
Media pack request datagram message from buyer to seller

4.4.1.3   Request-Response messaging model

1) A Media Pack Pricing & Demographics Request (MP-PR) can be issued by a buyer and **SHOULD** result in a Media Pack Pricing & Demographics message (MP-P) that provides the requested information.

2) If a Media Pack Pricing & Demographics message is a response to a Media Pack Pricing & Demographics Request, it **MUST** reference the Media Pack Pricing & Demographics Request's message identifier.
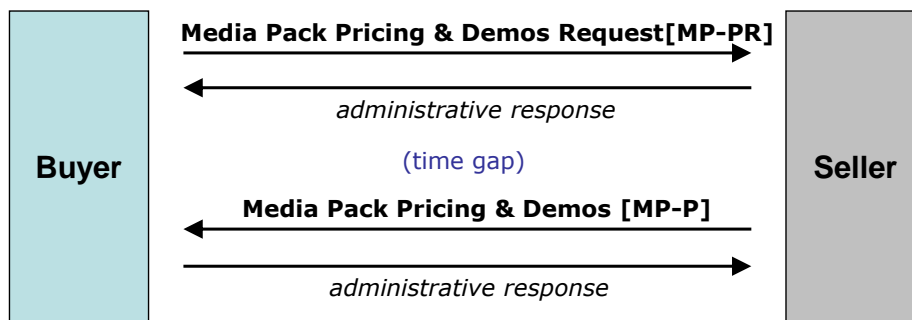
Media pack request-response messages

4.4.2   Rate Card Broadcast

The Seller broadcasts a new or updated rate card to all potentially interested parties.

1) A seller **MAY** send a Media Pack Pricing & Demographics message (MP-P) message spontaneously at any time.
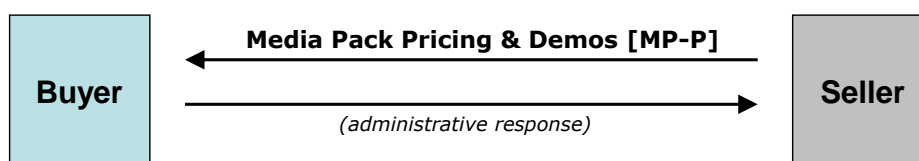
Media pack message

4.4.3   Rate Card Withdrawal

The Seller withdraws a previously issued rate card.

1) A seller **MAY** send a Media Pack Withdrawal message (MP-X) message spontaneously at any time.

Media pack withdrawal message
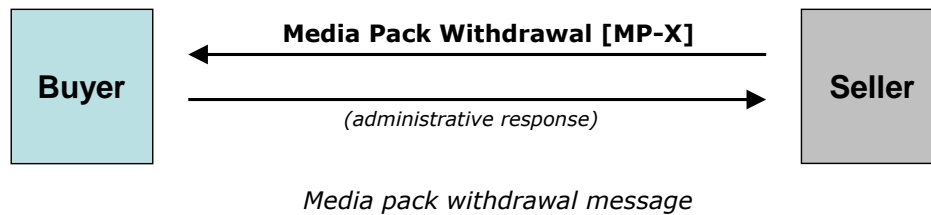
4.4.4   Message References – Ratecard Identifiers

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

Each Rate Card is uniquely identified with a `RatecardIdentifier` that conforms to the AdsML QID type. The Rate Card maintains its ID during its lifetime.

When a rate card is replaced, the replacement message explicitly references the `RatecardIdentifier` of the rate card that it replaces, much as an AdsML Materials message can explicitly identify a prior materials message for which it serves as a replacement.

Similarly, a Rate Card Withdrawal message references the `RatecardIdentifier` of the Rate Card that the seller wishes to withdraw.

Therefore, in order to support the Rate Card message choreography, buyers and sellers must retain a record of the `RatecardIdentifiers` for each rate card that they have sent or received for as long as there remains a possibility that the Rate Card may be replaced or withdrawn.

AdsML QID types exist for several additional objects, internal to the Rate Card, such as `RateGroup`, `Rate` and `AdditionalService`. These identifiers can be used as reference anchors for change specifications when a replacement message is sent, to indicate where in the message changes occur.

5 Usage of Business Messages

5.1 Overview

A rate card provides rates, or prices, for advertisements and services in one or more publications sold by a single selling party. Although the selling party is often the publisher or broadcaster, it may also be any intermediary that sells advertisements and related services.

Each selling party will have its own idea about how to arrange the products and services in its rate card. AdsMLMediaPack is designed to support this kind of flexibility. Selling parties should feel free to arrange the building blocks contained in the Rate Card specification so as to transmit rate cards that fully support their selling practices.

Further, in the world of e-commerce it is not necessary, or even desirable, for a selling party to try to squeeze all of its rates and products into a single rate card. AdsML Rate Card supports the creation of multiple rate cards for different purposes. Selling parties should feel free to create as many rate cards as necessary to represent both their standard rates and products and also any specially negotiated rates that may apply for specific customers or contracts.

The primary building blocks of a rate card are:

- A set of **header fields** that identify the selling party and **publications** to which this rate card applies, and that also define other scoping characteristics such as the effective dates of the rate card and any particular parties for whom these rates are applicable.

- One or more **rate groups**. Each rate group contains the rates, or prices, for a *related set of advertisements or services*. The primary components of a rate group are:

  - A subset of the same fields that are found in the header of the rate card as a whole. These define the **scope** of the rate group. For example, a rate group may apply to just one or two of the publications that were listed at the top of the rate card, or to specific time periods or days of the week.

  - A set of **rates,** or prices, for advertisements and services that are offered within the scope defined by the header of the rate group. Each rate can also have **rate qualifiers**, which provide further information about exactly when that particular rate takes effect.

- **Surcharges** and **discounts** that adjust the publisher's standard rates. These can be scoped to apply either within a single rate group or across the entire rate card.

There are also many other components of a rate card: additional services, terms and conditions, languages, etc. that supplement these core structures. Some of these components are available both in the header of the rate card as a whole and also at the Rate Group level, where their presence indicates that they apply to the advertisements governed by that particular rate group.

In order to understand how the primary building blocks work together to define the rate card, it can be helpful to compare them to a typical printed rate card:

- The **header information** identifies and scopes the overall rate card, much like the cover page of a printed rate card. There may also be supplemental information, such as a discussion of distribution of these publications and the demographics of their audience.

- Each **rate group** is similar to a page of a printed rate card: there is a header to explain what the rate group is about (e.g. a particular publication, edition, program, time period, etc.), followed by one or more pricing tables. There may also be supplemental information, such as a discussion of distribution of these advertisements and the demographics of their audience.

- Each **rate** corresponds to a row in a printed price table, listing a particular product or service and its related price. Its *rate qualifiers* correspond to the text in a printed rate card that indicates e.g. "Sundays only".

- **Surcharges** and **discounts** correspond to the extra prices that are often found in the footer or back page of a printed rate card: "setup charge", "color charge", "early purchase discount", etc. These can apply either to a specific page of the rate card (when they are inside a rate group) or to the rate card as a whole.

5.1.1   Header information

It is possible, even common, for a selling party to issue many rate cards. Reasons include: different sets of products may each be given their own rate card; there might be different validity periods, so that the "current" rates are in one card while "next year's" rates are in another; and individual customers might have negotiated special rates, in which case they can be provided with a "customized" rate card just for them.

The business information in the top level of the rate card message identifies and scopes the rate card. Relevant fields include:

- The date on which the rate card was issued (`adsml:IssueDate`)

- The date range within which it will be effective (`adsml:ValidityPeriod`)

- The `adsml:SellingParty` that offers the prices and services described in this rate card

- A list of the `Publications` to which this rate card applies

- An optional mechanism to indicate that this rate card contains custom prices that are valid only for a specific advertiser or media buyer or when governed by a particular contract (`ValidFor`)

All of these fields are self-explanatory except for `ValidFor`. See its definition in *AdsMLMediaPack – Part 2 - Specification & Schema* for a description of how to use it.

5.1.2   Publications

The `Publication` element in the header of a rate card is used to identify and describe all the publications related to that rate card[1]. These can then be related to individual rate groups, rates and charges/discounts within the rate card as necessary.

In general, a `Publication` in a rate card includes any version of a publication for which different ad rates can apply, e.g. `'FT Europe'` vs. `'FT America'`.

---

[1] As always in the AdsML Framework, "Publication" is a media-agnostic term encompassing print publications, television and radio programs, billboards, and any other medium that supports advertising.

Publication parts such as editions, supplements or sub web sites that have their own individual rates and prices **MUST** be represented using individual different `Publication` elements in the rate card.

When a "publication" is actually a sub-part of another publication, it is possible to indicate this fact by populating `ParentPublicationReference` in the child `Publication`.

The `PublishingSchedule` element includes the general publishing schedule for the publication as a whole. Detailed information about a publication's distribution and audience, including demographic and circulation information, can be provided in its `Statistics` element. (Usage of `Statistics` is described later in this section.)

> NOTE: Sub sections of a publication may have other schedules and statistics as expressed inside the `RateGroup` element.

The publication's language can be identified using the `Language` element. If the publication contains text in more than one language, repeated instances of the `Language` element can be used to identify them, and one can be specified as 'primary'.

> NOTE: Technical Specifications (`TechSpecs`) are located in `RateGroup`, and are not provided for the publication as a whole as this data is more related to individual parts/sub sections of a publication.

5.1.2.1   Relating publications to all or part of a rate card

Once identified at the top level of a rate card, a publication is available to be referenced from elsewhere in the rate card using the mandatory unique `PublicationIdentifier` element. There are several levels of a rate card that can be associated with one or more publications:

- Every **rate card** as a whole is associated with one or more publications. This is accomplished by populating a full `Publication` element for each publication in the top level of the `RateCard` as described above.

- An entire **rate group** is considered by default to apply to all publications listed in the rate card, but may optionally be associated with only one or more of these publications. This is accomplished by placing `PublicationReferences` in the root of the rate group, each of which points to one of the `Publications` at the top level of the rate card. The effect of this is to indicate that the rates in this rate group <u>only</u> apply to advertisements in the identified publications.

- An individual **rate qualifier** in a Rate or Charge/Discount may optionally be associated with one or more publications. This is accomplished by placing `PublicationReferences` inside a `RateQualifier` in the `Rate` or `ChargeDiscount`. The association can be used to specify that a particular rate qualifying property only applies to the listed publication(s).

See the discussions of Rate Groups, Rates, and Charges and Discounts, below, for more information about these kinds of associations.

5.1.2.2   Rules and Guidelines

Each publication that is related to a rate card in any way **MUST** be identified in a `Publication` element at the top level of the rate card.

Publications sub parts for which a seller wishes to charge different prices **MUST** be represented using individual different `Publication` elements in the rate card.

Since a publication may appear in many rate cards, it is **RECOMMENDED** that each publication maintain the same identifier in `PublicationIdentifier` across all the rate cards in which it appears.

5.1.3   Rate Groups

A rate group serves as a holder for a basket of rates, or prices, relating to the publication of advertisements that share specific qualities. The fields in the top level of the `RateGroup` define those shared qualities. For example, by populating the fields in its top level a Rate Group can be associated with:

- One or more specified publications, for example, "Time Magazine" or "Who Wants to be a Millionaire" or the "FT Asian Edition" (`PublicationReference`).

- A particular publishing schedule, for example "Weekdays" or "April 1–April 5" or "Morning Drive Time" (`PublishingSchedule`)

- A set of Statistics that describe the distribution of, and audience for, the advertisements governed by this rate group (`Statistics`)

- One or more human languages (`Language`)

- A set of Terms and Conditions (`TermsAndConditions`)

- A set of Technical Specifications to which all of the advertisements must conform (`TechSpecs`)

The qualifiers listed above are optional and may be omitted to create a generally applicable Rate Group. In particular, when no publications are explicitly listed, the Rate Group **MUST** be applicable for all publications listed in the Rate Card.

A selling party may define its rate groups as narrowly or broadly as it desires. For example, it is possible to create a rate group for each individual publication handled by the seller, or one for all of the publications whose audiences share particular characteristics, or one for a particular time of day or geographic region, or any combination of the above.

5.1.3.1   Rules and Guidelines

All advertisement offers listed in a `RateGroup` **MUST** follow the same Technical Specifications, Publishing Schedule and other metadata provided at the top level of the `RateGroup` element.

If a `PublicationIdentifier` appears in a `RateGroup` it **MUST** point to a `Publication` that is defined in the same message.

5.1.4   Rates, charges and discounts

Each `Rate` element defines an individual price relating to the publication of an advertisement that falls within the scope defined by the containing `RateGroup`. Each `ChargeDiscount` element describes a surcharge or discount that can be applied on top of the base rate under specified circumstances.

For example, there might be different base prices based on the size, duration or technical characteristics of the advertisement. These might then be modified by surcharges such as a handling charge for late arriving artwork and discounts for various reasons.

A detailed discussion of rates and prices can be found in the "Pricing" section below.

5.2 Transactions

5.2.1   Rate card broadcast

It is expected that rate cards normally will be transmitted in a broadcast fashion without any prior request.

A selling party will typically issue many rate cards, especially because rates are valid for different validity periods, but also for other, practical, reasons arising from established business practice.

A rate card with a particular `RatecardIdentifier` can be transmitted many times, to different receivers, but any two rate card document instances that have the same `RatecardIdentifier`, **MUST** have identical content (i.e. be copies).

5.2.2   Rate card request

The AdsML Media Pack specification also defines a rate card request message. This message includes a number of *selectors*, i.e. elements that can be used to specify which rate card the sender is interested in. For instance, by specifying a booking party, an advertiser, or a contract, the sender of a rate card request message could ask a selling party to deliver specific rate cards, possibly with special rates according to negotiated contracts.

NOTE: A single rate card request may generate several rate card response messages depending on how the rate card provider has structured its rate information.
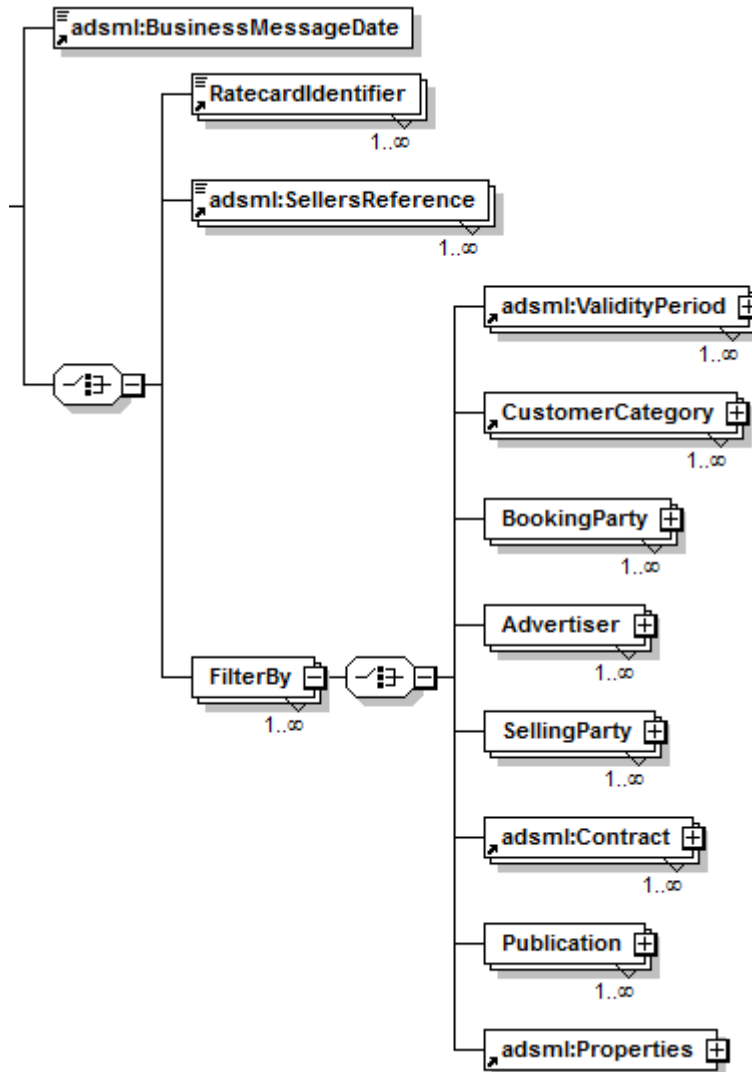
Selectors such as `adsml:ValidityPeriod` or `CustomerCategory` may be repeated inside a wrapper `FilterBy` element, which is also repeatable. The combination of the two levels allows simple as well as more complex selection criteria.

5.2.2.1   Example: Rate card for two publications

If more than one instance of a particular selector is provided inside a `FilterBy`, the recipient **MUST** interpret this as a request for a rate card for each of the values. The following filter should thus be interpreted as a request for rate cards for both of the two publications 'General Health' and 'Classic Cars':

```
<FilterBy>
  <Publication>
    <Name>General Health</Name>
  </Publication>
  <Publication>
    <Name>Classic Cars</Name>
  </Publication>
</FilterBy>
```

5.2.2.2   Example: Rate cards for two advertiser booking through an agency

It is further possible to combine several of the selectors, for instance having both a booking party and an advertiser specified. In this case, the recipient **MUST** interpret this as a request for a rate card where both selectors are valid, i.e. a rate card for a particular advertiser <u>with orders booked by the listed booking party</u>.

Combining these two rules, a request for rate cards for a booking party and any two advertisers should be expressed as:

```
<FilterBy>
  <BookingParty>
    <Name>MediaAgency ABC</Name>
  </BookingParty>
</FilterBy>
<FilterBy>
  <Advertiser>
    <Name>Food Store</Name>
  </Advertiser>
  <Advertiser>
    <Name>The Grocery Store</Name>
  </Advertiser>
</FilterBy>
```

The request above should generate rate cards for two combinations of values: 1) `MediaAgency ABC` and `Food Store`, and 2) `MediaAgency ABC` and `The Grocery Store`.

Note that the number of rate cards that the seller provides in response to this request will depend on how the seller has structured its rate information.

5.2.2.3   Example: Selectors with many values

Several of the selectors can take multiple child elements; for instance, an `adsml:Identifier` and a `Name` are found in `Advertiser`. If more than one child element is populated in the request, then all the populated child element values **MUST** be used for the selection (AND logic). For instance:

```
<FilterBy>
  <Advertiser>
    <adsml:Identifier>
      <adsml:IDLabel>VATNo</adsml:IDlabel>
      <adsml:IDValue>SE5565061198</adsml:IDValue>
    <Name>Food Store</Name>
  </Advertiser>
</FilterBy>
```

The `Advertiser` selector as specified above is asking for a rate card for the company `Food Store` with a VAT number of `SE5564061198`; both these values must be fulfilled.

5.2.2.4   Usage Rules and Guidelines

There is an AND logic between `FilterBy` instances, and OR logic between multiple selector instances. There is AND logic between values internal to selector instances.

5.2.3   Rate card change

A rate card can (and usually does) supersede a previously-issued rate card. The superseded rate card can be explicitly identified in the

`ReplacesRatecardReference` element at the top of the replacement rate card.

When an updated rate card is issued that references the QID of an earlier rate card, the old rate card with that QID **MUST** be replaced completely by the new one.

An updated rate card **MUST** have a new `RatecardIdentifier`.

5.2.4   Rate card withdrawal

The Rate Card Withdrawal message can be used to withdraw a rate card, without replacing it with another one. The most important piece of information in this message is the `RatecardIdentifier` that must be the ID of the rate card to withdraw.

The business meaning of a rate card withdrawal is that the products defined in the rate card are no longer available for sale according to the terms sent earlier.

5.3 Pricing

5.3.1   Rates and Prices

A rate card typically contains many rates. Each `Rate` element defines an individual price and the conditions, or *qualifiers*, which are associated with the price. (See below for a discussion of rate qualifiers.)

Each Rate is part of a rate group. All of the rates in a rate group share certain characteristics that are defined in the top level of the `RateGroup` structure. See the discussion of Rate Groups earlier in this chapter for more information.

Individual prices are provided by the `PricePerUnit` element. It may be repeated to provide the price in different currencies and/or units. The unit can be used to explicitly specify prices such as price per millimeter, price per module or 'CPM' (cost per thousand).

Additional information about the rate can be provided as a human-readable text in the `adsml:Note` element. It may be repeated to express the content in different languages.

Like the rate card as a whole, the `Rate` structure in AdsMLMediaPack is designed to support both automated and manual processing. Implementers can define a rate as consisting of little more than a code and a price, optionally augmented by a description line. Or you can go further and provide machine-processable information indicating exactly when and how the rate should be applied. It is **RECOMMENDED** that both types of information should be included in a rate card: each rate **SHOULD** be fully described in human-readable text in the `adsml:Description` and (when appropriate) `adsml:Note` elements, and the machine-processable elements such as `PricingBasis` and `RateQualifier` **SHOULD** also be populated so as to enable automated parsing of the rate card by those systems that are capable of doing so.

5.3.1.1   Usage Rules and Guidelines

If the `Unit` in `PricePerUnit` is left unspecified, it **MUST** be interpreted as "a single instance of publication" of the advertisement. Depending on the media in question this could take the form of an "insertion," "impression", "appearance", etc. It is **RECOMMENDED** to explicitly specify the unit.

5.3.2   Package pricing for more than one publication

The rate card supports rates for package products involving more than one publication. This is done by creating a `RateGroup` that includes multiple `PublicationReferences` that identify the publications in the package, and then using the `PriceIsFor` element inside each `Rate` to indicate how the pricing defined by that `Rate` applies to the package as a whole.

5.3.2.1   Use of PriceIsFor

Inside a `Rate`, the mandatory `PriceIsFor` specifies whether "top-down" or "bottom-up" pricing is used, i.e. if the price is given per *each publication* in the parent `RateGroup`, or for *all publications* in the package. `PriceIsfor` takes either '`AllPublications`' (top-down pricing) or '`EachPublication`' (bottom-up pricing) as values.

For example, in a `RateGroup` that references four publications, a rate with `PricePerUnit` of '1 000 dollars' and `PriceIsFor` = '`AllPublications`' will cost a total of $1,000 to run in all four specified publications, whereas the same rate with `PriceIsFor` = '`EachPublication`' would cost $4,000 to run in those four publications.

5.3.3   Charges and discounts

In addition to the publisher's standard rates, a Rate Card can contain additional charges and discounts. Charges and discounts usually consist of adjustments to the publisher's standard prices that occur under specific circumstances, although it is possible to use them for other purposes. They are conveyed in the `ChargeDiscount` element that can appear both at the top level covering the complete rate card, or with a limited scope inside a Rate Group.

Each `ChargeDiscount` defines an individual charge or discount and the conditions, or *qualifiers*, which are associated with the charge or discount. Charge/Discount qualifiers are expressed using the same `RateQualifier` structure that is used for Rates. (See below for a discussion of rate qualifiers.)

A charge or discount amount is recorded as a mandatory choice between:

1) a percentage specified in an `adsml:Percent` value, or

2) an absolute price specified as a `PricePerUnit` element. The `PricePerUnit` element is repeatable to allow for prices in several different currencies and/or units.

Discounts are provided as negative numbers, both for percentages and absolute amount values.

5.3.4   When to use Rates vs. Charges/Discounts

There can be some question as to when a price is a `Rate` and when it is a `ChargeDiscount`. The following rules and guidelines apply:

• If a price is a discount, i.e. a negative amount, it **MUST** be expressed as a `ChargeDiscount`.

• If a price is a percentage of another value, it **MUST** be expressed as a `ChargeDiscount`. (This is enforced by the schema.)

- If a price is the "base price" for publishing an advertisement, it **MUST** be expressed as a `Rate`.

- If a price is for a surcharge or discount relating to the publication of advertisements but is not associated with any specific publication it **MUST** be expressed as a `ChargeDiscount` at the root level of the message. \

- If the publisher would normally characterize a price as a "surcharge", it **SHOULD** be expressed as a `ChargeDiscount`.

- Prices that are not directly related to the publication of advertisements are considered to be "additional services" rather than either a rate or charge/discount, and **SHOULD** be described using the `AdditionalService` element instead.

In all other cases, the publisher may use its discretion when deciding whether to express a price as a `Rate` or a `ChargeDiscount`.

5.3.5   Rate Qualifiers

Each rate and charge/discount can be qualified with a set of conditions that must be fulfilled in order to get the listed price. These conditions are called *rate qualifiers* and can be provided using the repeatable `RateQualifiersGroup` that provides a repeatable choice of qualifiers.

For instance, a 'color charge' would include a rate qualifier for `Color` with a value that defines that the rate applies *only when the ad is published in full color*:

```
<RateQualifier.NewspaperMagazine>
  <Colors>
   <adsml-bo:ColorType>
     <adsml:CodeValue>Color</adsml:CodeValue>
   </adsml-bo:ColorType>
  </Colors>
</RateQualifier.NewspaperMagazine>
```

An additional size qualifier can be added to express that a rate is for 'full page and full color'. This is implemented by creating a second `RateQualifier` element alongside the first one:

```
<RateQualifier.NewspaperMagazine>
  <Colors>
   <adsml-bo:ColorType>
     <adsml:CodeValue>FullColor</adsml:CodeValue>
   </adsml-bo:ColorType>
  </Colors>
</RateQualifier.NewspaperMagazine>
<RateQualifier>
 <Sizes>
   <adsml-bo:Size>
     <adsml-bo:AdSizeCode>
       <adsml:CodeValue>FullPage</adsml:CodeValue>
     </adsml-bo:AdSizeCode>
   </adsml-bo:Size>
 </Sizes>
</RateQualifier>
```

The combination of the two qualifiers indicates that the rate applies only to ads that are both full page and in color. The presence of the media specific `RateQualifer.NewsPaperMagazine` further indicates that the rate applies only to colors for print ads. Additional qualifications can be expressed by adding more `RateQualifiers`.

In addition to rate qualifiers for color and sizes as in the example above, also qualifiers like `Positioning` (where on a page or on a site), `Distribution` (to whom), `Frequency` (how often), `adsml:Duration` (how long), `Weights` and `Thicknesses` (for inserts) and `BuyingVolume` (how much) are available. A general purpose extension point is also provided, the `GeneralQualifier` that can take any rate qualifying value not explicitly catered for.

> NOTE: `RateQualifiers` are primarily intended for machine processing. A human-readable description of each rate's qualifiers should be included in that rate's `adsml:Description`, with additional information about the qualifiers (if necessary) in its `adsml:Note`. Basically, the name of the rate, which is expressed as its `adsml:Description`, should include information about when the rate applies such as "Base price" or "Color surcharge on full page ads". This is usually the case in printed rate cards.

5.3.5.1   Media specific Rate Qualifiers

Most rate qualifiers are media-independent and expressed in the `RateQualifier` element, but some rate qualifiers, such as `RateQualifier.NewspaperMagazine`, are closely related to a particular media type and are grouped in media-specific versions of the `RateQualifier` element. Media-specific rate qualifiers serve two purposes:

1. They indicate that the qualifier in question is related to that particular media, thus making it easier to interpret its meaning;

2. Their presence indicates that the parent **Rate** or **ChargeDiscount** applies only to advertisements published in the specified medium.

Many of the media-specific rate qualifiers are based on structures defined in AdsMLBookings. For further information about the use of these structures see the AdsMLBookings documentation.

5.3.5.2   Combining several Rate Qualifiers

`RateQualifiers` are repeatable within their parent structure (e.g. `Rate` or `ChargeDiscount`) with a single qualifier per instance, and they can be combined in any order to express complex conditions.

In this regard there is no functional difference between the media-independent and media-specific rate qualifiers. They all obey exactly the same processing rules, and can be mixed and matched as necessary in order to express complex conditions.

5.3.5.3   Rules and Guidelines

1. In case of many `RateQualifier.*` elements, the rate given applies to the total combination of ALL rate qualifiers (AND logic).

2. The specific qualifiers for size, position etc. located inside the `RateQualifier.*` elements are also repeatable. In the case of multiple qualifiers inside a single `RateQualifier.*` element, the rate given applies to ANY combination (OR logic) of these qualifiers.

3. A list of publications may be referenced in relation to a qualifier using the `PublicationReference` element to express that the qualifier applies to specific publications ONLY.

5.3.5.4   *Example: Both AND and OR logic*

Two `Positioning` qualifier elements define '`Page 4`' and '`Page 6`', and two
`Scheduling` qualifier elements specify '`Mondays`' and '`Tuesdays`' respectively.
The price given in the `Rate` applies then to '`Page 4/Mondays`', '`Page
4/Tuesdays`', '`Page 6/Mondays`', and '`Page 6/Tuesdays`'.

```
<RateQualifier.NewspaperMagazine>
  <Positioning>
    <adsml-bo:PlacementInBook>
      <adsml-bo:Page>4</adsml-bo:Page>
    </adsml-bo:PlacementInBook>
  </Positioning>
  <Positioning>
    <adsml-bo:PlacementInBook>
      <adsml-bo:Page>6</adsml-bo:Page>
    </adsml-bo:PlacementInBook>
  </Positioning>
</RateQualifier.NewspaperMagazine>
<RateQualifier>
  <Scheduling>
    <adsml-bo:RecurrencePattern>
      <adsml:Code>
        <adsml:CodeValue>Mondays</adsml:CodeValue>
      </adsml:Code>
    </adsml-bo:RecurrencePattern>
  </Scheduling>
  <Scheduling>
    <adsml-bo:RecurrencePattern>
      <adsml:Code>
        <adsml:CodeValue>Tuesdays</adsml:CodeValue>
      </adsml:Code>
    </adsml-bo:RecurrencePattern>
  </Scheduling>
</RateQualifier>
```

5.3.5.5   *Example: Prices for specific publications*

A combo offer of a '`full-page ad in a weekly magazine supplement plus a
newspaper last page offered for a single rate`' would have two rate
qualifiers, one per publication.

```
<RateQualifier.NewspaperMagazine>
  <Positioning>
    <adsml-bo:PlacementInBook>
      <adsml-bo:PlacementCode>
        <adsml:CodeValue>LastPage</adsml:CodeValue>
      </adsml-bo:PlacementCode>
    </adsml-bo:PlacementInBook>
  </Positioning>
  <PublicationReference>pub.com:2009-01-01:ABC123</PublicationReference>
</RateQualifier.NewspaperMagazine>
<RateQualifier>
  <Sizes>
    <Size>
      <AdSizeCode>
        <adsml:CodeValue>FullPage</adsml:CodeValue>
      </AdSizeCode>
    </Size>
  </Sizes>
  <PublicationReference>pub.com:2009-01-01:XYZ456</PublicationReference>
</RateQualifier>
```

5.3.6   Determining when a Charge or Discount applies

When describing a charge or discount, it is often necessary to indicate specific conditions under which it should be applied; for example, in the case of a `handling charge`, exactly when does the handling charge take effect?

The Rate Card provides two mechanisms to specify the applicability of a surcharge or discount:

- implicit scoping by location, and
- rate qualifiers.

These two mechanisms are ANDed together to define the circumstances in which the `ChargeDiscount` should be applied.

Firstly, each charge or discount is constrained by its location in the message. A `ChargeDiscount` that is inside a `RateGroup` can only apply to the products and circumstances specified in that rate group, while a `ChargeDiscount` at the root level of the `RateCard` can apply to all of the rates or products in the rate card. For example, a `Holiday Charge` that applies to all of the products in the rate card would be defined in a `ChargeDiscount` at the root level of the rate card.

Secondly, a charge or discount may be associated with a set of conditions and requirements on position, color, sizes, etc. that further constrain when to apply the charge or discount. For instance, a `color charge` would include a rate qualifier for `Color` which indicates that it applies to any of the rates in the same context (rate group or rate card) but *only when the ad is published in color*. Similarly, a `Holiday Charge` would normally include one or more `Scheduling` rate qualifiers to indicate the holidays in question.

Of course, a `color charge of 15%` will probably apply to the rates for some products (e.g. `display ads`) but not others (e.g. `classifieds`). How can this be expressed?

- If the Rate Card is structured so that the pricing for Display Ads and Classifieds are defined in two different `RateGroups`, then one can use implicit scoping to associate the color charge with the Display ads simply by including the color surcharge's `ChargeDiscount` element in the same `RateGroup` that governs the display ads.

- If such implicit scoping is not feasible, the `GeneralQualifier` element in the `RateQualifier` child element can be used to provide a human-readable `Note` to explain exceptions from the default applicability of the charge or discount – in this case, `display ads only`.

5.3.6.1   Usage rules and guidelines for all Charges and Discounts

- A `ChargeDiscount` that is defined inside a `RateGroup` **MUST** apply only to the products and under the circumstances that are defined in the same `RateGroup`.

- A charge or discount that is not associated with a specific product, or that applies generally to all products, **SHOULD** be positioned at the root level of the `RateCard`.

- If a `ChargeDiscount` includes `RateQualifiers`, then it **MUST** apply only to products and rates that have matching `RateQualifiers`.

5.3.7   Understanding percentage-based charges and discounts

If a charge or discount is specified as a percentage, the trading partners need to understand which price(s) the charge or discount will be based on; for example, in the case of a `color charge of 15%`, 15% *of what*? The base price for publishing an ad may have several components. Should the surcharge or discount be calculated by multiplying 15% times *all* of those price components, or perhaps just *some* of them? Or could it be a surcharge on top of another surcharge?

In order to remove this ambiguity, when a charge or discount is expressed as a percentage, the publisher should use the `PricingBasis` element that is a sibling to `adsml:Percent` to convey an explicit basis for the pricing calculation. The pricing basis can be conveyed as either `Code` or `Text`; for example, the text `applies to base rate only` would indicate that the surcharge or discount will be calculated based on the base rate of the ad, ignoring any other prices or charges that may also be in effect. Conversely, `applies to all rates` would indicate that this surcharge or discount is based on the sum of all the applicable `Rates` in this rate card.

If no `PricingBasis` is explicitly provided, the default rule is that the surcharge or discount should be calculated by multiplying the specified percentage times the sum of all applicable rates. The previous section of this document described how to determine the circumstances under which a given `ChargeDiscount` is in effect. When determining the implicit pricing basis for a percentage `ChargeDiscount`, the same logic is applied: the percentage should be multiplied times all rates which are in the same scope of the message as the `ChargeDiscount` and have matching `RateQualifiers`.

5.3.7.1   Usage rules and guidelines for percentage-based Charges and Discounts

- An explicit pricing basis **SHOULD** be defined by the inclusion of a `PricingBasis` element inside the `ChargeDiscount` containing either a code value that has been agreed between the trading partners or else text that is sufficiently self-explanatory.

- The percentage basis for a `ChargeDiscount` **MUST** only include products and rates that are in the same scope as the `ChargeDiscount`, based on its location in the message.

- If no `PricingBasis` is provided, then the pricing basis of the `ChargeDiscount` **SHOULD** be deemed to be the sum of all of the Rates that are defined in the same scope of the message and have matching `RateQualifiers`.

5.3.8   Additional services

A Rate Card can also provide prices for services that are not directly related to the publication of an advertisement. Such services are described in `AdditionalServices` elements at the top level of the rate card.

5.3.9   Currencies

AdsMLMediaPack 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 where the currency is defined beforehand in the Trading Partner Agreement.

Each price within the message also contains its own optional currency code which, if populated, overrides the document's default currency for that particular price.

5.4 Publication statistics (demographic and circulation information)

"Statistics" is the term used in an AdsML Rate Card to describe information about a publication's distribution and audience, including demographic and circulation information. For example:

- How many copies of this publication are normally distributed to each region?

- How many people viewed or listened to this program during a particular period?

- What type of people were they?

- What is their household income?

AdsML provides a generic, extensible mechanism to represent any appropriate statistic about a publication and its audience. The structure is agnostic in terms of media and the types of statistics that can be represented.

Statistics that apply to a publication as a whole can be captured in a repeatable `Statistics` element inside the `Publication` element. Sub sections of a publication, such as a particular page, may have other statistics which are expressed in a repeatable `Statistics` element in the relevant `RateGroup`.

Despite its plural name, each instance of `Statistics` represents one set of information of a particular type. In most cases it will be necessary to provide multiple `Statistics` elements in order to convey all of the desired information about a publication.

> NOTE: The information in the `Statistics` structure is primarily intended to be machine processed. A human-readable description of the statistics associated with a `Publication` would normally be found in that publication's `Name` and `DescriptiveInformation`. Similarly, a human-readable description of the statistics associated with a `RateGroup` would normally be found in the rate group's `Name` and `adsml:Description`.

5.4.1   Components of a Statistic

Each instance of `Statistics` contains a mandatory `Type` code to define the type of statistic, e.g. 'Readership' or 'HouseholdDistribution'. This is the only mandatory element. The optional elements include

- A description line;

- A set of geographic, demographic and temporal *Scope Qualifiers* that make it possible to provide statistics for a smaller scope than the overall coverage for the publication. For instance, statistics for 'Southeast Region during May 2009' can be expressed as a combination of `GeographicScope` and `TemporalScope`;

- A `Count`;

- Two *Data Qualifiers* that can indicate the validity period of these statistics and the source from which they were obtained; and

- A `SubStatistics` structure that can convey an embedded list of information, such as *the count of listeners in each region in which this broadcast is heard*.

If a `Count` is provided, the type of item being counted is defined by the value of `StatisticsType`, e.g. a count of 'Readers' or 'Households'.

If no `Count` and no `SubStatistics` are provided, then the *Scope Qualifiers* constitute the payload of the statistic. For example, a `StatisticsType` of "Distribution" combined with a `GeographicScope` of "Worldwide" is sufficient to indicate that this publication or section of a publication is distributed globally. (Of course, it is then up to the publisher to explain to its trading partners what it means by "Worldwide".)

The `DataPeriod` element describes the overall time coverage of the statistics. For example, it can be used in combination with the `TemporalScope` to express that this instance of `Statistics` is valid for 'Weekends' during 'Q1 2009'.

The source of the statistics is recorded in the `DataSource` element. This records the provenance of the statistics and is often an authoritative industry body, such as the 'Audit Bureau of Circulations' that issues the ABCs, or another source such as the publisher itself.

5.4.2   Scope Qualifiers

The optional *Scope Qualifiers* (`GeographicScope`, `TemporalScope` and `DemographicScope`) constrain the nature of the `Statistics` by indicating one or more geographic regions, temporal periods or demographic characteristics of the audience.

If multiple qualifiers are provided, the statistics are deemed to apply to all possible combinations of the qualifiers. If a `Count` is provided, then a mathematical summation is also performed, which consists of adding up the number that is associated with each possible combination in order to arrive at the total count.

For example, suppose that `StatisticsType` = 'Circulation' and there are two region codes ('Region 1' and 'Region 2') in `GeographicScope` and two time periods ('Mondays' and 'Wednesdays') in `TemporalScope`. This tells us that the publication is distributed to Region 1 AND Region 2 on both Mondays AND Wednesdays.

Suppose that in addition to the two regions and two days of the week a `Count` element is provided whose value = '500,000'. This tells us that if we add up the number of copies distributed to each of the two regions on each of those two days, the total will be 500,000. We do not know how many copies were distributed to each individual region or on each individual day, but we do know the grand total.

If a scope qualifier is omitted or empty, then it is considered not to apply at all. When the user wants to convey the fact that no qualifier applies, that constraint (e.g. `GeographicScope`) should be omitted.

If no qualifiers are provided, the `Statistics` are deemed to represent the entire distribution/audience of the publication.

5.4.3   Sub Statistics

The `SubStatistics` element is a repeatable structure that allows for a further detailed level of statistics within the overall scope. The `SubStatistics` element includes a subset of the same elements as the `Statistics` element itself, with the same internal logic and semantics.

`SubStatistics` are most useful when it is necessary to convey an embedded list that consists of codes and associated counts, for example, a list of the regions to which this publication is distributed along with the circulation in each region. It is possible to accomplish this by transmitting a set of `SubStatistics` elements inside a single `Statistics`. The parent `Statistics` would have `StatisticType = 'Circulation'`. Each of the `SubStatistics` would also have `StatisticType = 'Circulation'`, plus a `GeographicScope` element containing a region name or code and a `Count` element containing the number of copies distributed in that region.

5.4.4   Patterns and Examples

Each instance of `Statistics` in a Rate Card generally follows one of four patterns:

A single code or text string defining the publication's distribution or audience. For example, *the Region in which this publication is distributed*. This can be accomplished by populating the mandatory `StatisticsType` plus a single scope qualifier, in this case a `GeographicScope` containing the region's name or code.

A single count accompanied by an explanation of what that count represents. For example, *total audience as of last December 31*. This is accomplished by transmitting the mandatory `StatisticsType`, the `Count`, and as many scope qualifiers as necessary to explain what the count represents. In this example no scope qualifiers are needed, since the count is for total audience, but the `DataPeriod` element would be used to indicate *as of December 31*.

A list consisting of codes or text strings. For example, *the set of Regions in which this publication is distributed*. This is accomplished by populating the mandatory `StatisticsType` plus a stack of `Code` or `Text` elements inside the appropriate scope qualifier, which in this case is `GeographicScope`.

A list in which each item is accompanied by a count. For example, *the circulation in each Region to which this publication is distributed.* This is accomplished by providing a set of SubStatistics, each of which contains a single GeographicScope and its associated Count.

Because the Statistics element is repeatable, many aspects of Statistics can be provided by populating multiple instances of the element, each representing a different aspect of the total picture.

5.4.5   Usage Rules and Guidelines

- If a `Count` is provided, the type of item being counted **MUST** be defined by the value of `StatisticsType`.

- When the user wants to convey the fact that no qualifier applies, that constraint (e.g. `GeographicScope`) **SHOULD** be omitted. If a scope qualifier is omitted or consists of an empty element, then that constraint **MUST** be considered not to apply at all.

- If no scope qualifiers are provided, the `Count` **SHOULD** be deemed to represent the entire distribution/audience of the publication.

5.5 Taxation information

AdsMLMediaPack supports the transmission of taxation information in two contexts:

> 1) Each `Party` in a Rate Card 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 Business message dates

The message types defined in AdsMLMediaPack contain a business-significant date at the top of the message. It comes in two variations depending on context:

- `adsml:IssueDate` (in the `Ratecard` message)

- `adsml:BusinessMessageDate` (in the `RatecardRequest` and `RatecardWithdrawal` messages)

`adsml:IssueDate` specifies the original date when the Ratecard was first issued and should as such would normally be different from the date of transmission.

`adsml:BusinessMessageDate` **MUST** specify the date on which the business information in the current message was considered valid by its sender. (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.)

5.7 Usage and Definitions of Controlled Vocabularies

AdsMLMediaPack 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 AdsMLMediaPack 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 AdsMLMediaPack standard in many common situations. It assumes a basic familiarity with the message format, and with the more detailed choreography and usage information provided above.

6.1 Creating rate cards

6.1.1   Publish a standard rate card

Scenario: A selling party distributes its standard rate card to all of the media buyers with whom it regularly does business. This is a broadcast message containing the publisher's latest rates; no one specifically requested it.

AdsML handling: The seller creates a Rate Card message by populating the `Ratecard` element in AdsMLMediaPack and transmits it in broadcast mode to all potentially interested parties. The seller populates `RatecardIdentifier` with a unique QID for this rate card, but omits both `ReplacesRateCardReference` and `adsml:ChangeSpecification` because they are not appropriate in this situation.

The seller populates `adsml:IssueDate` with the date on which the rate card was first publicly released (e.g. today) and `adsml:ValidityPeriod` with the date on which the prices in this rate card will take effect, and populates the mandatory `adsml:Status` code with a value indicating that the prices in the rate card are "active", approved prices. For purposes of clarity the seller should also provide a `Name` for this rate card, e.g. "`Standard rates and prices as of [date]`". Because this is a standard rate card the `ValidFor` element is omitted.

For every publication, including editions, channels, web sites or any other publication that is mentioned anywhere in the rate card, the seller populates a `Publication` element at the top level of the message. Inside each `Publication` the seller uses the `Statistics` element to convey information about that publication's distribution and audience, potentially including formal ratings or audited circulation.

In order to group its prices in a manageable way the seller creates one or more `RateGroups`, each of which contains the prices for publishing a related set of advertisements – for example, ads in a specific publication, or ads broadcast during a particular time period. The seller provides a human-readable `Name` and `adsml:Description` for the rate group, and also populates other elements in its top level with machine-processable information to define the scope of that rate group. For example, if the rate group applies to ads in just one publication, a pointer to that publication is placed in `PublicationReference`; if the rate group applies to ads published during a specific time period, the time period is defined in `PublishingSchedule`. If specific Terms and Conditions or Technical Specifications apply to advertisements governed by this rate group, they are provided in the `TermsAndConditions` and `TechSpecs` elements, respectively. And information about the distribution and demographic reach of ads governed by this rate group can be provided in the embedded `Statistics` element.

Inside each rate group the seller creates a set of `Rate` and `ChargeDiscount` elements, one for each available product or service that relates to publishing

advertisements within the scope of the rate group. For example, the rate group might contain different possible base prices based on the size, duration or technical characteristics of the advertisement, as well as surcharges and discounts for various reasons that apply on top of the base price.

Inside each `Rate`, the seller populates the mandatory `adsml:RateCode` and `PricePerUnit` with the name of this rate and its price, and the mandatory `PriceIsFor` element with a code that indicates whether the rate should be applied repeatedly to each publication in which the ad is published whether it applies just once to the entire set of publications governed by this rate group.

The seller provides an optional human-readable `adsml:Description` for the rate, and also supports machine processing by populating one or more `RateQualifier.*` elements with structured codes that govern exactly when the rate applies.

For each surcharge or discount in the rate group the seller creates a `ChargeDiscount` element, which is similar to a `Rate` except that its price can be expressed as either a `PricePerUnit` or an `adsml:Percent`. If the surcharge or discount is percentage-based, the seller should also use `PricingBasis` to explain how to calculate the amount of the surcharge or discount.

Prices that are related to publishing advertisements but that do not fit within the scope of any particular rate group (for example, a "Holiday Surcharge") can be conveyed in a `ChargeDiscount` element at the top level of the rate card. Prices for products and services that are not directly related to the publication of advertisements are conveyed as Additional Services using the `AdditionalService` element.

Finally, any Terms and Conditions relating to the rate card as a whole are placed in `TermsAndConditions` at the top level of the `RateCard`. These can be represented both as in-line text strings and also in one or more formatted documents (e.g. PDFs) via the `adsml:TermsAndConditionsDetails` element.

Notes:

If any of these publications in a rate card was previously referenced in another rate card, the seller should be careful to re-use the same publication identifier as before.

6.1.2   Publish rates for a specific trading partner

Scenario: A selling party has negotiated special rates with an important advertiser and now wishes to publish a rate card containing those rates.

AdsML handling: The seller creates a Rate Card message containing the special rates. The seller indicates that these rates are valid only for that particular customer by populating `ValidFor/adsml:Advertiser`, in the top level of the rate card, with the advertiser's name and ID.

Notes:

Using the `ValidFor` mechanism it is possible to indicate that a rate card is valid for one or more specific advertisers, buyers or categories of customers (such as a trade association), and/or in the context of one or more specific contracts.

A seller may publish multiple rate cards, each with different validities, which are simultaneously in effect. For example, a publisher may publish its "standard" rates as well as a series of special rate cards, each for a specific trading partner.

6.1.3   Publish a rate card with future prices

Scenario: A selling party wishes to inform its trading partners of new rates that will take effect on January 1 of the coming year.

AdsML handling: The seller creates a rate card message containing the new rates. The seller indicates that these rates will be applied from January 1 by populating `adsml:ValidityPeriod/StartDateTime` in the top level of the rate card.

Notes:

When a validity period contains a start date that is in the future, the rate card contains prices that will be applied from that date. When a validity period contains an end date that is in the past, the rate card is no longer in effect.

It is also possible to indicate the end date of a validity period by providing a start date and duration, such as "`January 1`" for "`one year`". This has the same effect as explicitly specifying an end date.

6.1.4   Provide rates for several publications in a single rate card

Scenario: A selling party wishes to provide a rate card containing rates for more than one publication.

AdsML handling: A single rate card may include rates for one or more publications. All publications must be listed at the top level of the rate card using the `Publication` element, where each publication is required to have `PublicationIdentifier`, a unique `adsml:QIDType` identifier.

For each publication with a unique set of rates, a particular Rate Group is created using the `RateGroup` element. The association between a rate group and publication is maintained by recording the publication's unique `PublicationIdentifier` in the `PublicationReference` element inside `RateGroup`.

In case several of the publications share the same rates, a rate group may be associated with all those publications by repeating `PublicationReference`.

Note

There might also be several rate groups associated with the same publication. The different rate groups are required in case subsections of a publication have different rates. Publication subsections often also differ in terms of publishing schedule, technical specifications, terms and conditions and other data that can be expressed at the rate group level.

6.1.5   Provide different rates for editions or sub-sections of a publication or broadcast

Scenario: A selling party wishes to publish a rate card that includes different prices for the different versions of a publication, for example, regional editions. Further, some of these publications include individual sections that have their own pricing.

AdsML handling: The seller creates a rate card in which each of these publications, sub-publications, sections and editions is defined in its own `Publication` element at the top level of the rate card. For each publication (or sub-publication etc.) that has its own pricing, the seller creates a `RateGroup`

containing those prices. In the top level of the rate group it provides one or more `PublicationReferences` to identify the specific publications (or sub-publications etc.) that are associated with this rate group.

6.1.6   Replace a rate card

Scenario: A selling party wishes to send a replacement rate card that will supersede a previously provided rate card.

AdsML handling: The seller sends a Rate Card message containing the new rates, and populates the `ReplacesRatecardReference` element in the top level of the message with the QID of the earlier rate card that has now been superseded.

Notes:

The `ReplacesRatecardReference` mechanism is designed to be used when a rate card becomes incorrect or invalid prior to its normal validity period having expired. It is not necessary to use this mechanism when sending a successor rate card that will go into effect after the expiration of the validity period of a previously issued rate card.

See "Withdraw a Rate Card" for a related scenario.

6.1.7   Provide rates for combo offers involving more than one publication

Scenario: Seller wishes to provide a combo offer consisting of ads in two publications that, when purchased together, qualify for a special package price.

AdsML handling: The seller defines a rate for this combination and uses `adsml:DescriptionLine` to describe the package in human readable text, for example: 'A full-page ad in a weekly magazine supplement plus a newspaper last page offered for a single rate.' This is accompanied by two `RateQualifier` elements, each of which identifies a required component of the package in a machine-processable manner.

Rate qualifiers can be scoped to apply to a particular Publication using the `PublicationReference` element provided with each rate qualifier. For the scenario above, the seller should provide two `RateQualifers`. In the first `RateQualifer`, a `Positioning` element specifies the last page in the newspaper and a `PublicationReference` identifies the specific newspaper. In the second `RateQualifer`, the `Size` element specifies 'full page' and is accompanied by a `PublicationReference` pointing to the magazine in question. These two `RateQualifiers` are ANDed together, thus indicating that the rate applies only when both qualifications have been met.

Notes

See the discussion of Rate Qualifiers earlier in this document for an explanation and examples of the processing rules.

6.1.8   Particular rates for special dates

Scenario: A publication offers special rates on weekends, except for the July-August period. There is also a "Thanksgiving day surcharge of 10%".

AdsML handling: A rate group may be associated with a specific `PublishingSchedule`, the presence of which indicates that the rates in this rate group are applicable according to dates and periods defined here. (A

publishing schedule may have been provided in the `Publication` structure at the top level of the rate card, but a local schedule in the rate group will override the publication schedule for the rates and publications that are included in this rate group.)

Inside `RateGroup/PublishingSchedule`, the `adsml-bo:RecurrencePattern` element is used to express that the rates apply during weekends; this constraint can be given as a machine-readable code or a human readable text string.

The `Period` element is then used to define that the rates are <u>not</u> applicable during July-August by setting the `StartDateTime` and `EndDateTime` to '2010-07-01' and '2010-08-31' respectively and the attribute negated to 'true'.

The 'Thanksgiving day surcharge' could have been defined as a special rate group explicitly listing all rates for all products available this day. But because it is a percentage surcharge, it is simpler to use the `ChargeDiscount` element in the rate group we already have. In this case, `RateQualifier/Scheduling` is used to define the day when this surcharge applies, by either an explicit date (`Period` element) or a `PreDefinedPeriod` code.

Note

If the 'Thanksgiving day surcharge' applied to all rates in the rate card, rather than only the rates in this rate group, the seller would define a top level `ChargeDiscount` structure for it.

6.1.1   Rates for particular times of day, e.g. "dayparts"

Scenario: A broadcast network provides rates for defined times of day such as "morning drive time" and "prime time".

AdsML handling: Times of day are considered to be a type of recurrence pattern, and are specified in the `adsml-bo:RecurrencePattern` element. This element can be found both in `PublishingSchedule` at the top level of each rate group, and also in `RateQualifier/Scheduling` inside each rate or charge/discount.

So the question is: in which of these locations should the seller convey daypart information? The answer depends on how the seller has structured its rate card. If the rate groups themselves are bound to dayparts – for example, if there is a rate group for 'Morning Drive Time', a rate group for 'Prime Time', etc. – then the `PublishingSchedule` element at the top level of each rate group should define the daypart(s) with which that rate group is associated. But if the dayparts are associated with individual rates inside a rate group – for example, if a rate group called 'Nationwide' contains individual rates such as 'Morning Drive Time – 30 seconds' and 'Prime Time – 15 seconds' – then the appropriate daypart for each rate should be specified in `RateQualifier/Scheduling` inside that rate.

In all cases, each daypart is conveyed as either a `Code` or a human-readable `Text` element in an instance of `adsml-bo:RecurrencePattern`. Although the `Code` and `Text` elements are repeatable, in practice only one daypart can be included in a recurrence pattern. Sellers wishing to indicate that a rate applies to two or more dayparts (e.g. Morning Drive Time <u>and</u> Evening Drive Time) will need to transmit multiple instances of `RateQualifier/Scheduling`, each of which contains a recurrence pattern that includes one of the dayparts in question.

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.

- Although only one daypart can be included in each recurrence pattern, it is possible to combine it with other types of recurrence information inside an instance of `adsml-bo:RecurrencePattern` provided that they are not mutually exclusive. For example, the codes 'MorningDriveTime' and 'Tuesdays' can coexist inside a recurrence pattern because they have different time scales; together they indicate that the rate applies to ads that run during morning drive time hours on Tuesdays.

- See the discussion of Rate Qualifiers earlier in this document for more information about how to combine dayparts with other types of qualifiers.

> 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.

6.2 Rate card requests

6.2.1   Request a seller's latest rate card

Scenario: A media buyer wishes to make sure that it has the latest rates offered by a particular seller.

AdsML handling: The buyer creates a Rate Card Request message and sends it to the seller. In the top level of the message the buyer populates at least one of the `FilterBy` selectors, `FilterBy/BookingParty`, in which it places its own name and identifier. This is a simple filter that signals that the buyer wants to receive any and all rate cards that will be relevant to it.

The seller responds by sending to the buyer all rate cards that it considers relevant for the filters that were specified, in this case that buyer. This may well be the publisher's "standard" rate card, or there may be one or more special rate cards that this buyer is eligible to receive.

Notes:

In addition to `BookingParty`, the buyer may populate other filters such as `adsml:ValidityPeriod` to indicate for which period in time rates are requested, and `Publication/PublishedBy` to express which publisher's rate card is requested.

See the section on rate card request transactions above for a discussion of how to convey complex requests using AND and OR logic.

6.2.2   Request a specific rate card

Scenario: A media buyer wishes to make sure that it has the latest rates offered by a particular seller under specific circumstances, for example, the rate that a particular advertiser would receive if it wished to buy time on a particular broadcast.

AdsML handling: The buyer creates a Rate Card Request message and sends it to the seller. In the `FilterBy` element in the top level of the message the buyer populates `adsml:Advertiser` with the name and identifier of the advertiser in question, and `Publication` with the name and product code of the broadcast in question.

The seller responds by sending to the buyer all rate cards that it considers relevant for this situation.

Notes: See the section on rate card request transactions above for a discussion of how to convey complex requests using AND and OR logic.

6.2.3   Request a replacement rate card

Scenario: A party wishes to request that a seller re-send a previously sent rate card.

AdsML handling: The party creates a Rate Card Request message and sends it to the seller, populating the `RateCardIdentifier` element in the top level with the QID of the rate card that it wishes to receive again.

Notes:

If the QID of the rate card is not known, the requesting party can alternatively populate `FilterBy` with values that will cause the seller to re-send the appropriate rate card.

6.2.4   Withdraw a rate card

Scenario: A selling party wishes to withdraw a rate card without simultaneously issuing a replacement for it.

AdsML handling: The seller sends a Rate Card Withdrawal message and populates `RatecardWithdrawal/RatecardIdentifier` at the top level of the message with QID of the rate card that is being withdrawn. The seller may optionally provide an explanation in `ReasonForWithdrawal`.

Notes:

See "Replace a Rate Card" for a related scenario.

6.3 Pricing

6.3.1   Defining a general discount

Scenario: A general agency commission discount of 15% is given on the total net amount.

AdsML handling: The general agency commission discount is applied on top of all rates; therefore, the `ChargeDiscount` element at the top level of `Ratecard` should be used to specify the discount.

The `ChargeDiscount` is required to have a rate code in the same way as other rates, e.g. '`AgencyCommission`'. A human readable description should be stored in the `adsml:DescriptionLine`.

The agency commission is '`-15%`' and needs to be recorded in the `adsml:Percent` element. The default rule is that percentage-based charges/discounts should apply to all products/rates listed within the context in which that charge or discount appears, i.e. all rates in the rate card in this case.

Since that is how the agency commission should be calculated, no additional information needs to be transmitted to define it.

Notes

Exceptions to the default calculation base for a percentage-based charge or discount can be provided using the optional `PricingBasis` element. This element conveys a list of machine-processable code values and/or human-readable texts that explain how the percentage should be calculated.

6.3.2   Defining a complex price (price per mm with multiple qualifiers and a minimum size)

Scenario: Seller wishes to describe a rate for black & white classified ads placed in a publication's car section, with a price per millimeter and a minimum size of 10 mm.

AdsML handling: Rates are defined inside a Rate Group that is associated with one or more Publications. In this scenario, the newspaper is defined as a Publication and associated with the rate group by means of the `PublicationReference` element.

A Rate is required to have a unique `adsml:RateCode` serving as the main identifier for the Rate. The optional `adsml:DescriptionLine` is used to capture a human friendly description such as "Classified Cars mm price".

The mandatory `PriceIsFor` element should be 'EachPublication' in this scenario with a single publication.

The actual price is listed in the `PricePerUnit/Price` element where the `PricePerUnit/Unit` is 'millimeter'.

The `RateQualifier` structures are then used to describe what the requirements are for getting this mm price. In `RateQualifier.NewspaperMagazine`, the `Colors` qualifier specifies 'black' (this is the standard AdsMLControlledVocabularies value for 'b&w'), and a value of 'ClassifiedCars' in `adsml:Positioning/SectionInBook`/`adsml-bo:SectionCode` indicates that the rate only applies to ads in the Classified Cars section. The minimum size of '10 mm' can be defined in `RateQualifer/Sizes` by populating the `MinSize/Height` qualifier.

Notes

The `adsml:RateCode` is also used in the bookings workflow where advertisements may be booked that reference a particular rate.

6.4 Multilingual use cases

6.4.1   Describe a multilingual publication

Scenario: A primarily Finnish publication listed in a rate card can have pages in Finnish and Swedish.

AdsML handling: The seller creates a rate card with the publication defined in the `Publication` element at the top level of the rate card. The seller can use the repeatable `Language` element to define all languages that the publication provides content in, Finnish ('fi') and Swedish ('sv') in this case. As the majority of the content is in Finnish, this language is marked as primary.

6.4.2   Send a rate card in multiple languages

Scenario: The Selling Party is required to prepare a media pack for usage in different languages without knowing beforehand which language a particular reader of the media pack will prefer. He would like to provide human-readable strings in several languages in parallel.

AdsML handling: Text fields in the AdsMLMediaPack that include content intended for human reading are normally repeatable to allow for the same text content in different languages. Examples are names, terms and condition strings and descriptions of code values.

The language used is recorded using the `xml:lang` attribute. There is also an optional `dir` attribute to explicitly record the reading direction, e.g. 'ltr' for 'left-to-right'.

The following example shows a `DescriptionLine` describing an Agency Commission discount in Finnish, Swedish and English:

```
<ChargeDiscount>
  <RateCode>
    <adsml:CodeValue xsi:type="adsml-
cv:AdsMLPriceComponentCV">AgencyCommission</adsml:CodeValue>
  </RateCode>
  <DescriptionLine xml:lang="fi">Mainostoimistoalennus 15%</DescriptionLine>
  <DescriptionLine xml:lang="sv">Mediabyråkommission 15%</DescriptionLine>
  <DescriptionLine xml:lang="en">Agency Commission @ 15%</DescriptionLine>
  <adsml:Percent>-15.00</adsml:Percent>
</ChargeDiscount>
```

Notes:

Usage of multiple parallel languages is the only permitted rationale for repeating many string elements in a rate card. See the specification of the parent elements for detailed information.

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 Media Pack content and usage is normally structured according to the selling party's view of its business, products and rates. Decisions such as one or many rate cards, one or many publications and rate groups per rate card are made by the selling party alone without need for prior acceptance by the buying (receiving) party.

The selling party should create rate cards that are as complete as possible including as much as possible of the available data. It will then be up to the buyer's (receiver's) decision to extract more or less information from the rate card message based on his business requirements.

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. Others reflect important choreography choices that need to be agreed between trading partners when they are establishing AdsML communications.

Each package consists of either:

- a set of hierarchical levels from which one must be selected (represented by a numbered list), or

- a set of non-exclusive options from which any combination can be selected (represented by a bullet list), or

- a list of mutually-exclusive choices from which one must be selected (represented by a textual description).

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

7.1 Message exchange mode

There are three defined message exchange modes:

1. Full Request-Response

2. Datagram model from seller to buyer

3. Datagram model from buyer to seller

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.

7.2 Multilingual metadata

Trading partners must agree on whether or not they support the provision 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 that they will impose.

7.3 Pricing

7.3.1   Currencies

Trading partners must agree on whether a currency that is different from the rate card's Document Currency Code may be specified for a particular rate (in `Rate/PricePerUnit`).

7.3.2   Top down of bottom up pricing

Trading partners must agree on whether package pricing is supported or not, i.e. if `Rate/PriceIsFor` can take the value "`AllPublications`".

7.4 Rate Card Request

Trading partners need to agree on whether they will support the Rate Card Request message, and if so, which selectors in the `FilterBy` structure (`Advertiser`, `Publication` etc.) may be used.

7.5 Updates – Frequency and processing

New rate cards can be transmitted from the seller to the buyer at any time, likely including changes. A rate card can also be withdrawn by the seller at any time.

7.5.1   Use of change location pointers

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

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

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:

- Jay Cousins (RivCom)
- Beat Doerr (Publigroup)
- John Iobst (NAA)
- Joe Kirk (K Media Solutions)
- Mark Kondrad (Colorquick)
- Jari Kosonen (Kärkimedia)
- Tony Stewart (RivCom)

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

- Petra Förars (Dagmar)
- Pekka Kalkkinen (Helsingin Sanomat)
- Juha Vuorinen (Helsingin Sanomat)