AdsML® Framework for E-Commerce Business Standards for Advertising

AdsMLBookings 2.5.0
Part 2
Specification & Schema

Document Authors: AdsML Technical Working Group

Document ID: AdsMLBookings-2.5.0-SpecP2Schema-AS-1

Document File Name: AdsMLBookings-2.5-SpecP2Schema-AS.pdf

Document Status: Approved Specification

Document Date: 15 April 2010

Draft Number: 1

Table of Contents

1 AdsMLBookings Standard Documentation

1.1 Document status and copyright

This is the Approved Specification of the *AdsMLBookings 2.5 Part 2 Specification & Schema*.

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.

1.2 Non-Exclusive License Agreement for AdsML Consortium Specifications

USER LICENSE

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

1. All AdsML Consortium Copyrightable Works are licensed for use only on the condition that the users agree to this license, and this work has been provided according to such an agreement. Subject to these and other licensing requirements contained herein, you may, on a non-exclusive basis, use the Specification.

2. The AdsML Consortium openly provides this specification for voluntary use by individuals, partnerships, companies, corporations, organizations and any other entity for use at the entity's own risk. This disclaimer, license and release is intended to apply to the AdsML Consortium, its officers, directors, agents, representatives, members, contributors, affiliates, contractors, or coventurers (collectively the AdsML Consortium) acting jointly or severally.

3. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this Usage License are included on all

such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the AdsML Consortium, except as needed for the purpose of developing AdsML specifications, in which case the procedures for copyrights defined in the AdsML Process document must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by AdsML or its successors or assigns.

4. Any use, duplication, distribution, or exploitation of the Specification in any manner is at your own risk.

5. NO WARRANTY, EXPRESSED OR IMPLIED, IS MADE REGARDING THE ACCURACY, ADEQUACY, COMPLETENESS, LEGALITY, RELIABILITY OR USEFULNESS OF ANY INFORMATION CONTAINED IN THIS DOCUMENT OR IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE PRODUCED OR SPONSORED BY THE ADSML CONSORTIUM. THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN AND INCLUDED IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE OF THE ADSML CONSORTIUM IS PROVIDED ON AN "AS IS" BASIS. THE ADSML CONSORTIUM DISCLAIMS ALL WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY ACTUAL OR ASSERTED WARRANTY OF NON-INFRINGEMENT OF PROPRIETARY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER THE ADSML CONSORTIUM NOR ITS CONTRIBUTORS SHALL BE HELD LIABLE FOR ANY IMPROPER OR INCORRECT USE OF INFORMATION. NEITHER THE ADSML CONSORTIUM NOR ITS CONTRIBUTORS ASSUME ANY RESPONSIBILITY FOR ANYONE'S USE OF INFORMATION PROVIDED BY THE ADSML CONSORTIUM. IN NO EVENT SHALL THE ADSML CONSORTIUM OR ITS CONTRIBUTORS BE LIABLE TO ANYONE FOR DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, COMPENSATORY DAMAGES, LOST PROFITS, LOST DATA OR ANY FORM OF SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY KIND WHETHER BASED ON BREACH OF CONTRACT OR WARRANTY, TORT, PRODUCT LIABILITY OR OTHERWISE.

6. The AdsML Consortium takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available. The AdsML Consortium does not represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication, assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the Secretariat of the AdsML Consortium.

7. By using this specification in any manner or for any purpose, you release the AdsML Consortium from all liabilities, claims, causes of action, allegations, losses, injuries, damages, or detriments of any nature arising from or relating to the use of the Specification or any portion thereof. You further agree not to file a lawsuit, make a claim, or take any other formal or informal legal action against the AdsML Consortium, resulting from your acquisition, use, duplication, distribution, or exploitation of the Specification or any portion thereof. Finally, you hereby agree that the AdsML Consortium is not liable for any direct, indirect, special or consequential damages arising from or relating to your acquisition, use,

duplication, distribution, or exploitation of the Specification or any portion thereof.

8. This User License is perpetual subject to your conformance to the terms of this User License. The AdsML Consortium may terminate this User License immediately upon your breach of this agreement and, upon such termination you will cease all use duplication, distribution, and/or exploitation in any manner of the Specification.

9. This User License reflects the entire agreement of the parties regarding the subject matter hereof and supercedes all prior agreements or representations regarding such matters, whether written or oral. To the extent any portion or provision of this User License is found to be illegal or unenforceable, then the remaining provisions of this User License will remain in full force and effect and the illegal or unenforceable provision will be construed to give it such effect as it may properly have that is consistent with the intentions of the parties. This User License may only be modified in writing signed by an authorized representative of the AdsML Consortium. This User License will be governed by the law of Darmstadt (Federal Republic of Germany), as such law is applied to contracts made and fully performed in Darmstadt (Federal Republic of Germany). Any disputes arising from or relating to this User License will be resolved in the courts of Darmstadt (Federal Republic of Germany). You consent to the jurisdiction of such courts over you and covenant not to assert before such courts any objection to proceeding in such forums.

10. Except as expressly provided herein, you may not use the name of the AdsML Consortium, or any of its marks, for any purpose without the prior consent of an authorized representative of the owner of such name or mark.

IF YOU DO NOT AGREE TO THESE TERMS PLEASE CEASE ALL USE OF THIS SPECIFICATION NOW. IF YOU HAVE ANY QUESTIONS ABOUT THESE TERMS, PLEASE CONTACT THE SECRETARIAT OF THE ADSML CONSORTIUM.

AS OF THE DATE OF THIS REVISION OF THE SPECIFICATION YOU MAY CONTACT THE AdsML Consortium at www.adsml.org.

1.3 AdsML Code of Conduct

The AdsML Code of Conduct governs AdsML Consortium activities. A reading or reference to the AdsML Code of Conduct begins every AdsML activity, whether a meeting of the AdsML Consortium, AdsML Working Groups, or AdsML conference calls to resolve a technical issue. The AdsML Code of Conduct says:

Trade associations are perfectly lawful organizations. However, since a trade association is, by definition, an organization of competitors, AdsML Consortium members must take precautions to ensure that we do not engage in activities which can be interpreted as violating anti-trust or other unfair competition laws.

For any activity which is deemed to unreasonably restrain trade, AdsML, its members and individual representatives may be subject to severe legal penalties, regardless of our otherwise beneficial objectives. It is important to realize, therefore, that an action that may seem to make "good business sense" can injure competition and therefore be prohibited under the antitrust or unfair competition laws.

To ensure that we conduct all meetings and gatherings in strict compliance with any such laws and agreements in any part of the world, the AdsML Code of Conduct is to be distributed and/or read aloud at all such gatherings.

- There shall be no discussion of rates, fares, surcharges, conditions, terms or prices of services, allocating or sharing of customers, or refusing to deal with a particular supplier or class of suppliers. Neither serious nor flippant remarks about such subjects will be permitted.
- AdsML shall not issue recommendations about any of the above subjects or distribute to its members any publication concerning such matters. No discussions that directly or indirectly fix purchase or selling prices may take place.
- There shall be no discussions of members' marketing, pricing or service plans.
- All AdsML related meetings shall be conducted in accordance with a previously prepared and distributed agenda.
- If you are uncomfortable about the direction that you believe a discussion is heading, you should say so promptly.

Members may have varying views about issues that AdsML deals with. They are encouraged to express themselves in AdsML activities. However, official AdsML communications to the public are the sole responsibility of the AdsML Consortium. To avoid creating confusion among the public, therefore, the Steering Committee must approve press releases and any other forms of official AdsML communications to the public before they are released.

1.4 Document Number and Location

This document, Document Number AdsMLBookings-2.5.0-SpecP2Schema-AS-1, is freely available. It will be located at the AdsML website at http://www.adsml.org/.

1.5 Purpose of this document

This document specifies the definition of the AdsMLBookings standard. AdsMLBookings is an XML-based language used for encoding and routing advertisement booking transaction messages.

1.6 Audience

The intended audience for this document is primarily user and vendor organizations who seek to implement the AdsMLBookings standard in their workflows, advertising systems, or software products. Those assessing the conformance of vendor products to the standard may also use the document.

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

1.7 Accompanying documents

This document serves as the reference guide to the AdsMLBookings schema. A companion document, *AdsMLBookings Part 1 – Usage Rules & Guidelines*, provides an overview as well as additional rules and guidance for using AdsMLBookings messages to address specific business requirements. They are meant to be read together.

Both 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.

In addition, elements and structures that are used in multiple AdsML schemas are documented in the *AdsML Type Library* specification. AdsMLBookings makes extensive use of such structures, therefore the *Type Library* specification is an essential reference.

A description of the entire document set can be found in the *ReadMeFirst* html file associated with this release of the AdsML 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)

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

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

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

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

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

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

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

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

1.8.3 Typographical conventions

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

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

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

1.9 Change History

Version	Date	Changes	Editor
2.5.0 AS-1	15 April 2010	First approved version of 2.5.	UW
2.0.0 AS-1	30 May 2008	Second approved version	UW
1.0 AS -1	1 June 2006	First approved version	UW

1.9.1 Changes in version 2.5

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

Version 2.5 of AdsML Bookings has considerably enhanced functionality in the support of bookings of interactive advertisement, plus a large number of smaller improvements.

For an overview of changes, please see *AdsMLBookings Part 1 – Usage Rules & Guidelines.*

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

Portions are based on material written by:

- Jay Cousins (RivCom) – jay.cousins@rivcom.com
- Tony Stewart (RivCom) – tony.stewart@rivcom.com

Acknowledgments and thanks to other contributors for additional input to this document are listed in *Appendix A: Acknowledgment for contributions to this document.*

1.11 The AdsML Consortium

The documents comprising the AdsML standard were written by the AdsML Technical Working Group, a committee charged with creating the consortium's technical deliverables, and then approved by the entire membership.

More information about the consortium can be found on the consortium's website: www.adsml.org.

2 AdsMLBookings XML Schema – Overview

This section describes the use of XML Schema from W3C (http://www.w3.org) in the definition of AdsMLBookings.

2.1 Schema Architecture

AdsMLBookings uses a modular schema architecture as defined by the AdsML Framework architecture consisting of the following schemas,

- The **Main Schema** – This schema defines the root element AdsMLBookings and all other components used in the standard, either by local definitions or by importing and/or including other schema files.
- The **Public Type Library** – This schema includes all components from AdsMLBookings that may be imported into other standards and reused.
- The **AdsML Type Library** – This schema defines reusable components from the AdsML Framework.
- The **AdsMLMaterials Public Type Library** – This schema defines components that make up the public part of the AdsMLMaterials standard, which is reused within AdsMLBookings.
- The **AdsMLStructuredDescriptions Public Type Library** - This schema defines components that make up the public part of the AdsMLStructuredDescriptions standard, which is reused indirectly in AdsMLBookings as part of the imported AdsMLMaterials Public Type Library.
- The **AdsML Controlled Vocabularies** – This schema defines all controlled vocabularies recommended by the AdsML Consortium.

All structures specific to AdsMLBookings are defined in the Main Schema or the Public Type Library that is included into the Main schema. These structures are all defined in the AdsMLBookings namespace.

Where possible, AdsMLBookings specific structures have been defined as derivations of general AdsML Framework components defined in the AdsML Type Library that is imported into both the Main Schema and the Public Type Library.

The `adsml-ma:AdContent` element for specification of advertising material within a booking is defined on a public component from the AdsMLMaterials standard exposed in its public type library. As AdsMLMaterials imports the AdsMLStructuredDescriptions, its public type library schema is also used.

The AdsML Controlled Vocabularies schema provides a set of controlled vocabularies (CVs) that may be used in AdsML messages. The CVs are made available to all document instances through import into the Main Schema.

2.1.1 Schema Files

The schema files from a particular standard are named as follows:

```
AdsMLBookings-2.5-Main-AS.xsd
```

The format starts with the name of the standard, "AdsMLBookings" followed by the current version number and the name of the schema within the standard. The last two characters provide the status of the standard as either PS (Proposed Specification) or AS (Approved Specification) for public releases (internal working documents have status code WD for Working Draft).

The complete set of schema files used in AdsMLBookings version 2.5, Approved Specification is thus:

AdsMLBookings-2.5-Main-AS.xsd

AdsMLBookings-2.5-PublicTypeLibrary-AS.xsd

AdsMLTypeLibrary-2.0-AS.xsd

AdsMLMaterials-2.5-PublicTypeLibrary-AS.xsd

AdsMLStructuredDescriptions-1.0-PublicTypeLibrary-AS.xsd

AdsMLControlledVocabularies-3.0-AS.xsd

2.2 AdsMLBookings Namespaces

AdsMLBookings defines a namespace according to W3C's Recommendations (http://www.w3.org):

'http://www.adsml.org/adsmlbookings/2.5'

This is defined as the default namespace of the AdsMLBookings Schema. The schema specifies this using *targetNamespace* and *xmlns* attributes as illustrated below,

```
<xs:schema targetNamespace="http://www.adsml.org/adsmlbookings/2.5"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.adsml.org/adsmlbookings/2.5" ... >
```

Components reused from other standards carry their own namespaces that also have to be declared. The following external namespace definitions are also used:

adsml='http://www.adsml.org/adsmltypelibrary/2.0'

adsml-ma='http://www.adsml.org/adsmlmaterials/2.5'

adsml-sd='http://www.adsml.org/adsmlstructureddescriptions/1.0'

adsml-cv='http://www.adsml.org/adsmlcontrolledvocabularies/3.0'

It is **RECOMMENDED** to use namespace prefixes as listed above.

It is **RECOMMENDED** to have the AdsMLBookings namespace as default namespace in AdsMLBookings document instances. If however a namespace prefix is wanted, it is **RECOMMENDED** to use "adsml-bo".

2.3 Validation and Schema Location

A trading partner **MUST NOT** send any invalid AdsMLBookings messages. However, use of XML Schema based validation of production messages in runtime is **OPTIONAL**. Systems are allowed to use any available approach to ensure that their output is valid.

For production messages, a schema location **SHOULD NOT** be given in document instances using the *xsi:schemaLocation* attribute. Systems are **REQUIRED** to be able to identify which schema a particular document instance belongs to by reading the mandatory *adsml:schemaVersion* attribute.

2.4 No empty values for elements and attributes

All elements and attributes that appear in an instance **MUST** take a value, i.e. are not allowed to be empty. The only exception to this is the case of elements which are defined with an empty content model.

2.5 Fixed and Default values

All fixed or default values specified for elements or attributes in the schema **MUST** be present in an XML document instance conforming to that schema; schema validation and the post-schema-validation infoset (PSVI) **SHOULD NOT** be relied upon in order to make fixed or default values available for processing.

This restriction is imposed so that a particular mode of validation (XML Schema validation and the PSVI) is not relied upon to ensure that all data content of a message is present in an instance messages. This allows for non-XML Schema validation of an instance.

This constraint is enforced in the schema by specifying attributes that carry fixed values with a 'use' of required, by not specifying default values, and by the policy that element content should not be empty in instances.

3 Content Model Reference

This is a reference section describing elements, attributes and other building blocks of the AdsMLBookings XML vocabulary's content model. The building blocks are listed in alphabetical order. The `AdsMLBookings` element is the root element, i.e. the top node of an AdsMLBookings message.

Each building block is briefly described with the intention of providing context and background as well as some technical detail about its usage. Particular focus is placed on issues and business rules that are not possible to express using XML Schema. Note that the XML Schema specification includes additional rules.

Components from imported external schemas are not described here; please see their specific specification documents. Such components are named with their recommended namespace prefix when discussed in the context of AdsMLBookings elements.

Elements and attributes with namespace prefix:	**Are described in the document:**
`adsml-ma`	*AdsMLMaterials Schema & Specification*
`adsml`	*AdsMLTypeLibrary Schema & Specification*

3.1 Root Element: AdsMLBookings

An AdsMLBookings message is an e-commerce business transaction that includes information to facilitate message transmission (a header with sender and recipient information) and the business content relevant to the transaction (e.g. ad order transaction data).

`AdsMLBookings` is the root element of the XML instance message where the namespace declaration is made. The namespace is defined on a string reflecting AdsML's ownership and the main version number. The namespace declaration **MUST** be based on the following string:

`'http://www.adsml.org/adsmlbookings/2.5'`

The choice of namespace prefix is not defined in the standard, but it is **RECOMMENDED** that the AdsMLBookings namespace be the default namespace in AdsMLBookings messages. If a namespace prefix is required, it is **RECOMMENDED** to use `'adsml-bo'`. A namespace declaration will in this case look like:

`xmlns:adsml-bo="http://www.adsml.org/adsmlbookings/2.5"`

Every AdsMLBookings message contains a mandatory `Header` element followed by one or more elements of a specific business message type such as `AdOrder` for bookings. The bookings in a message need not be related to each other in any other ways than that they are transmitted in the same physical XML message.

The root element `AdsMLBookings` is defined on the `adsml:AdsMLItemType`, please see this type for further details.

The optional `adsml:Properties` element can be used to define application-specific extensions.

Attributes

Please see `adsml:AdsMLItemType` for details on attributes.

3.2 Transaction Messages

3.2.1 Element: AdOrder

The `AdOrder` element is the top level element for the corresponding business message. An order is either created from a complete specification defined according to the reusable module `AdMessageRequestModule`, or from a previous reservation that is specified using the `ReservationReference` element.

The `AdOrder` is required to be identified using the `BookingIdentifier` element. It can optionally refer to a previous quotation using the `QuotationReference` element.

A stack of other references to this booking can be provided using the `AuxiliaryBookingReferences` element.

See the section on "*Message References…*" in *AdsMLBookings 2.5 Part 1, Usage Rules & Guidelines* for further information about the use of booking identifiers.

Two optional dates can be provided. The `BookingDate` records the date of the initial booking and the `adsml:BusinessMessageDate` holds the business significant date of the current message. In an `AdOrder`, these two dates will often be the same, but note that an order may be agreed well before the actual transmission is made and that the `BookingDate` in this case would rather record the day of the business level agreement rather than the day of the transmission of the business message.

Finally, the `MessageFooterGroup` includes elements available in all transaction messages including extensibility points for application-specific data and human oriented notes.

Attributes

messageCode (fixed: 'AD-O')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

3.2.2 Element: AdOrderCancellation

The `AdOrderCancellation` element is the top level element for the corresponding business message. An order cancellation request is defined according to the reusable module `AdMessageRequestModule`.

An order cancellation request message **SHOULD** contain the complete details of the order that is being cancelled.

The `AdOrderCancellation` is required to be identified using the `BookingIdentifier` element.

A stack of other references to this booking can be provided using the `AuxiliaryBookingReferences` element.

It is **RECOMMENDED** to specify the reason for the cancellation using the `adsml:ReasonForCancellation` element.

The `adsml:RevisionIdentifier` is an optional part of all bookings messages. It should be used to identify a particular booking's revision during the life time of a booking spanning many transaction messages. It can be recorded as a number or any other string value. When used, the originating `AdOrder` is considered as revision zero '0', and all later change messages or cancellation message **MUST** use an incremented value. An `AdOrderCancellation` of an order that has not changed since it was first booked would then have revision '1'.

Two optional dates can be provided. The `BookingDate` records the date of the initial booking and the `adsml:BusinessMessageDate` holds the business significant date of the current message.

Finally, the `MessageFooterGroup` includes elements available in all transaction messages, including extensibility points for application-specific data and human oriented notes.

Attributes

messageCode (fixed: 'AD-OX')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:lastReceivedMessageID (optional)

The unique message ID for the last message that the sender of this message has received regarding the booking. It can be used for conflict detection in case of simultaneously transmitted messages by the communication parties.

3.2.3 Element: AdOrderChange

The `AdOrderChange` element is the top level element for the corresponding business message. An order change request is defined according to the reusable module `AdMessageRequestModule`.

An order change request message **MUST** contain the complete details of the order that is being changed, reflecting its expected state once the changes have been accepted. In addition, it is **RECOMMENDED** to specify a specification of changes requested using the `adsml:ChangeSpecification` element.

The `AdOrderChange` is required to be identified using the `BookingIdentifier` element.

A stack of other references to this booking can be provided using the `AuxiliaryBookingReferences` element.

The `adsml:RevisionIdentifier` is an optional part of all bookings messages. It should be used to identify a particular booking's revision during the life time of a booking spanning many transaction messages. It can be recorded as a number or any other string value. When used, the originating `AdOrder` is considered as revision zero '0', and all later change messages or cancellation message **MUST** use an incremented value. The first `AdOrderChange` of a booking would then have revision '1'.

Two optional dates can be provided. The `BookingDate` records the date of the initial booking and the `adsml:BusinessMessageDate` holds the business significant date of the current message.

Finally, the `MessageFooterGroup` includes elements available in all transaction messages, including extensibility points for application-specific data and human oriented notes.

Attributes

messageCode (fixed: 'AD-OC')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:lastReceivedMessageID (optional)

The unique message ID for the last message that the sender of this message has received regarding the booking. It can be used for conflict detection in case of simultaneously transmitted messages by the communication parties.

3.2.4 Element: AdOrderResponse

The `AdOrderResponse` element is the top level element for the corresponding business message. A response is defined according to the reusable module `AdMessageResponseModule`.

The `AdOrderResponse` is required to be identified using the `BookingIdentifier` element.

A stack of other references to this booking can be provided using the `AuxiliaryBookingReferences` element.

The `adsml:RevisionIdentifier` is an optional part of all bookings messages. It should be used to identify a particular booking's revision during the life time of a booking spanning many transaction messages. It can be recorded as a number or any other string value. When used, the originating `AdOrder` is considered as revision zero '0', and all later change messages or cancellation message **MUST** use an incremented value. This includes `AdOrderResponse` "with change" which would then have revision '1'.

Two optional dates can be provided. The `BookingDate` records the date of the initial booking and the `adsml:BusinessMessageDate` holds the business significant date of the current message.

Finally, the `MessageFooterGroup` includes elements available in all transaction messages, including extensibility points for application-specific data and human oriented notes.

Attributes

messageCode (fixed: 'AD-OR')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:inResponseToMessageID (required)

The message ID for the message that the response is about.

adsml:inResponseToMessageCode (required)

The message code for the message that the response is about. For example, a response to an AdOrderChange would have the value 'AD-OC' for this attribute.

3.2.5 Element: AdOrderStatus

The `AdOrderStatus` element is the top level element for the corresponding business message. The content model is similar to the `AdOrderResponse`, except for the *`messageCode`* attribute (see below) and the presence of the optional *`adsml:lastReceivedMessageID`*.

Note that status messages can be sent without a prior status request.

Attributes

messageCode (fixed: 'AD-OS')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:inResponseToMessageID (optional)

The message ID for the message that the status response is about. The attribute **MUST** be used when the message is sent as a response to a StatusEnquiry request, and **MUST NOT** be used when the Status message is sent without a prior request.

adsml:inResponseToMessageCode (optional)

The message code for the message that the response is about. . The attribute **MUST** be used when the message is sent as a response to a StatusEnquiry request, and **MUST NOT** be used when the Status message is sent without a prior request.

adsml:lastReceivedMessageID (optional)

The unique message ID for the last message that the sender of this message has received regarding the booking. It can be used for conflict detection in case of simultaneously transmitted messages by the communication parties.

3.2.6 Element: AdOrderStatusEnquiry

The `AdOrderStatusEnquiry` element is the top level element for the corresponding business message.

The booking to which this `AdOrderStatusEnquiry` refers is required to be identified using the `BookingIdentifier` element.

The `adsml:BusinessMessageDate` holds the business significant date of the message

Attributes

messageCode (fixed: 'AD-OSE')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

3.2.7 Element: AdQuotation

The `AdQuotation` element is the top level element for the corresponding business message. A quotation is defined according to the reusable module `AdMessageResponseModule`.

The `AdQuotation` is required to be identified using the `QuotationIdentifier` element.

A stack of other references to this quotation can be provided using the `AuxiliaryBookingReferences` element.

It is required to specify an expiration time for the quotation using the `adsml:ExpirationTime`.

Two optional dates can be provided. The `QuotationDate` records the date of the initial quotation and the `adsml:BusinessMessageDate` holds the business significant date of the current message. In a quotation these are always the same value.

Finally, the `MessageFooterGroup` includes elements available in all transaction messages, including extensibility points for application-specific data and human oriented notes.

Note that it is possible to issue an `AdQuotation` message without a prior `AdQuotationRequest`.

Attributes

messageCode (fixed: 'AD-Q')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:inResponseToMessageID (optional)

The message ID for the AdQuotationRequest message that the AdQuotation response is about. The attribute **MUST** be used when the message is sent as a response to an AdQuotationRequest request, and **MUST NOT** be used when the AdQuotation is sent without a prior request.

adsml:inResponseToMessageCode (optional)

The message code for the message that the AdQuotation is a response about. The attribute **MUST** be used when the message is sent as a response to an AdQuotationRequest request, and **MUST NOT** be used when the AdQuotation is sent without a prior request.

3.2.8 Element: AdQuotationRequest

The `AdQuotationRequest` element is the top level element for the corresponding business message. A quotation request is defined according to the reusable module `AdMessageRequestModule`.

The `AdQuotationRequest` is required to be identified using the `QuotationIdentifier` element.

See the section on "*Message References…*" in *AdsMLBookings 2.5 Part 1, Usage Rules & Guidelines* for further information about the use of booking identifiers.

A stack of other references to this request for quotation can be provided using the `AuxiliaryBookingReferences` element.

An optional `adsml:BusinessMessageDate` holds the business significant date of the message.

Finally, the `MessageFooterGroup` includes elements available in all transaction messages, including extensibility points for application-specific data and human oriented notes.

Attributes

messageCode (fixed: 'AD-RFQ')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See adsml:commonMessageAttributes definition.

3.2.9 Element: AdQuotationStatus

The `AdQuotationStatus` element is the top level element for the corresponding business message. The content model is identical to the `AdQuotation` message, except for the *`messageCode`* attribute (see below).

Note that status messages can be sent without a prior status request.

Attributes

messageCode (fixed: 'AD-QS')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:inResponseToMessageID (optional)

The message ID for the message that the status response is about. The attribute **MUST** be used when the message is sent as a response to a StatusEnquiry request, and **MUST NOT** be used when the Status message is sent without a prior request.

adsml:inResponseToMessageCode (optional)

The message code for the message that the response is about. The attribute **MUST** be used when the message is sent as a response to a StatusEnquiry request, and **MUST NOT** be used when the Status message is sent without a prior request.

3.2.10 Element: AdQuotationStatusEnquiry

The `AdQuotationStatusEnquiry` element is the top level element for the corresponding business message.

The quotation to which this `AdQuotationStatusEnquiry` refers is required to be identified using the `QuotationIdentifier` element. An optional `adsml:BusinessMessageDate` may be provided.

Attributes

messageCode (fixed: 'AD-QSE')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *adsml:commonMessageAttributes* definition.

3.2.11 Element: AdReservation

The `AdReservation` element is the top level element for the corresponding business message. It is required to be identified using the `BookingIdentifier` element. It can optionally reference a previous quotation using the `QuotationReference` element.

A reservation request is defined according to the reusable module `AdMessageRequestModule`. Its content model is very similar to `AdOrder`, please see its description for further information.

See the section on "*Message References…*" in *AdsMLBookings 2.5 Part 1, Usage Rules & Guidelines* for further information about the use of booking identifiers.

Attributes

messageCode (fixed: 'AD-R')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See adsml:commonMessageAttributes definition.

3.2.12 Element: AdReservationCancellation

The AdReservationCancellation element is the top level element for the corresponding business message. A reservation cancellation request is defined according to the reusable module `AdMessageRequestModule`.

A reservation cancellation message **SHOULD** contain the complete details of the reservation that is being cancelled.

The AdReservationCancellation is required to be identified using the `BookingIdentifier` element. Its content model is very similar to `AdOrderCancellation`, please see its description for further information.

Attributes

messageCode (fixed: 'AD-RX')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *adsml:commonMessageAttributes* definition.

adsml:lastReceivedMessageID (optional)

The unique message ID for the last message that the sender of this message has received regarding the booking. It can be used for conflict detection in case of simultaneously transmitted messages by the communication parties.

3.2.13 Element: AdReservationChange

The `AdReservationChange` element is the top level element for the corresponding business message. A reservation change request is defined according to the reusable module `AdMessageRequestModule`.

A reservation change request message **MUST** contain the complete details of the reservation that is being changed, reflecting its expected state once the changes have been accepted. In addition, it is **RECOMMENDED** to provide a specification of changes requested using the `adsml:ChangeSpecification` element.

The `AdReservationChange` is required to be identified using the `BookingIdentifier`.

Attributes

messageCode (fixed: 'AD-RC')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:lastReceivedMessageID (optional)

The unique message ID for the last message that the sender of this message has received regarding the booking. It can be used for conflict detection in case of simultaneously transmitted messages by the communication parties.

3.2.14 Element: AdReservationResponse

The `AdReservationResponse` element is the top level element for the corresponding business message. A response is defined according to the reusable module `AdMessageResponseModule`.

The `AdReservationResponse` content model is very similar to `AdOrderResponse`, please see that description for further information, but note that in addition, in an `AdReservationResponse` it is mandatory to specify an expiration time for the reservation using the `adsml:ExpirationTime` element.

The `AdReservationResponse` is required to be identified using the `BookingIdentifier`.

Attributes

messageCode (fixed: 'AD-RR')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:inResponseToMessageID (required)

The message ID for the message that the response is about.

adsml:inResponseToMessageCode (required)

The message code for the message that the response is about. For example, a response to an AdReservationChange would have the value 'AD-RC' for this attribute.

3.2.15 Element: AdReservationStatus

The `AdReservationStatus` element is the top level element for the corresponding business message. The content model is identical to the `AdReservationResponse`, except for the *`messageCode`* attribute (see below).

Note that status messages can be sent without a prior status request.

Attributes

messageCode (fixed: 'AD-RS')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

adsml:inResponseToMessageID (optional)

The message ID for the message that the status response is about. The attribute **MUST** be used when the message is sent as a response to a StatusEnquiry request, and **MUST NOT** be used when the Status message is sent without a prior request.

adsml:inResponseToMessageCode (required)

The message code for the message that the response is about. The attribute **MUST** be used when the message is sent as a response to a StatusEnquiry request, and **MUST NOT** be used when the Status message is sent without a prior request.

adsml:lastReceivedMessageID (optional)

The unique message ID for the last message that the sender of this message has received regarding the booking. It can be used for conflict detection in case of simultaneously transmitted messages by the communication parties.

3.2.16 Element: AdReservationStatusEnquiry

The `AdReservationStatusEnquiry` element is the top level element for the corresponding business message.

The reservation to which this `AdReservationStatusEnquiry` refers is required to be identified using the `BookingIdentifier`.

The `adsml:BusinessMessageDate` holds the business significant date of the message

Attributes

messageCode (fixed: 'AD-RSE')

The AdsML Framework message type code for the message.

***attribute group*: adsml:commonMessageAttributes**

See *`adsml:commonMessageAttributes`* definition.

3.3 Component Reference

3.3.1 Element: AccountDebit

The `AccountDebit` element is used within the `Payment` structure to record the means by which the payment can be taken by directly debiting the payer's account (bank or otherwise). The child element `Reference` is used to record the account details.

See `Payment` for more information.

Attributes

No attributes.

3.3.2 Element: AdditionalPaymentInstructions

The `AdditionalPaymentInstructions` element is used within the `Payment` structure.

See `Payment` for more information.

Attributes

No attributes.

3.3.3 Element Group: AdMessageRequestModule

The `AdMessageRequestModule` element group is an assembly of elements that carry the information needed by request messages for quotations, reservations and orders. With the exception of status enquiries that only need to refer to the `BookingIdentifier`, all request messages have a common core set of content elements defined by this module.

The element group includes elements specifying the organizations, or parties, which take part in the booking. The required `adsml:BookingParty` identifies the booking organization, which might be different from the party sending the message, the advertiser or the payer. The also required `adsml:SellingParty` identifies the organization selling advertising to the advertiser or its agents.

It includes an optional set of `adsml:OtherParty` elements for specification of other parties that may be involved in the booking, e.g. repro houses or sales companies. Note that the `adsml:OtherParty` structure **MUST NOT** be used to record parties that are particularly catered for elsewhere in the bookings message by explicitly defined elements. For instance, materials providers, receivers of proofs, payers, advertisers and booking parties are parties where explicit elements exist and where `OtherParty` must not be used.

A reference to a purchase order can be included using the optional `adsml:PurchaseOrderReference` element.

Information about a campaign that the booking belongs to can be recorded as a formal code identifier using the `adsml:Campaign` element, and a more elaborated structure, `CampaignDescription`, includes fields for textual descriptions of the campaign's goal, running period and constraints. `CampaignDescription` may be repeated for information in alternative languages, but **MUST NOT** be repeated for any other reason.

An optional `DealCode` element can be used to specify whether the Booking is requested according to terms in a pre-defined package, or deal. Reference to a deal can be used to replace specific details regarding publication, production detail and/or insertion schedules.

An `adsml:Contract` structure allows for explicit identification of the "rate" or "level" that applies to a given Booking. It may also be specified at the placement

group or placement levels. Note that there may be several contracts which apply to a single order depending on the element context where the `adsml:Contract` element is used. For example the buying agency might have a sales contract with the seller, and each payer might have a different contract specifying their payment terms.

The `TermsAndConditions` element holds a set of human-readable textual terms regarding cancellations, claims, disclaimers and any other legalistic information that applies to the Booking as a whole. `TermsAndConditions` may be repeated for information in alternative languages, but **MUST NOT** be repeated for any other reason.

The `Guarantees` element can be used to record that certain aspects of the Booking as a whole have been "Guaranteed", and to convey the nature of that guarantee as a set of codes or text strings as appropriate.

Pricing information such as paying parties, amounts and discounts can be specified using the optional `TotalBookingPrice`, `CostExempt` and `PayerInformation` elements.

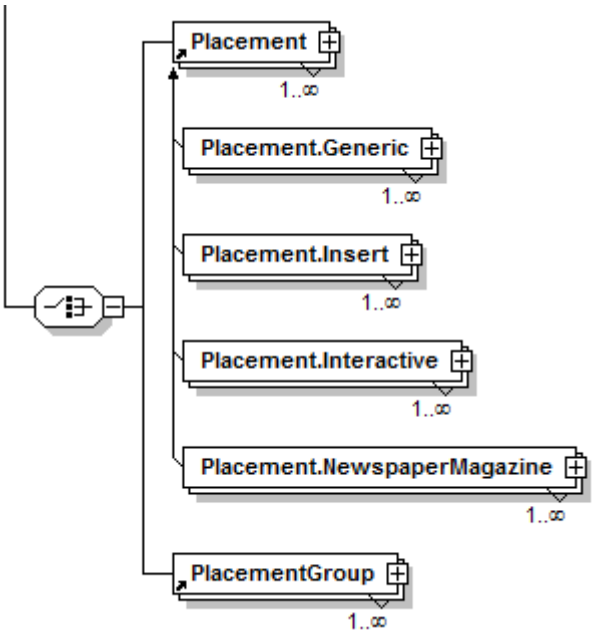
Prices and payers can be specified here, at the package level, and/or below for each `Placement` or `PlacementGroup`. `PayerInformation` is repeatable so that multiple payers can be specified for shared bookings. See `PayerInformation` for further information.

If the Booking is provided for free, the `CostExempt` element should be used to record the type and reason. The element **MUST** be omitted when the Booking is not cost exempt.

A single currency that applies to all prices in the whole Booking may be expressed using the `adsml:DocumentCurrencyCode` element. Note however that an alternative currency may be expressed per placement. The `adsml:ExchangeRate` structure expresses the relationship between the currency defined in the `adsml:DocumentCurrencyCode` and another currency from which its amount was derived or into which it can be converted. Either the source or target currency must correspond to the document currency.

A short descriptive text may be added using the `adsml:DescriptionLine` element. It may be repeated to capture the description in alternative languages, but **MUST NOT** be repeated for any other reason.

The element `LinkedPlacements` can be used to declare relationships between placements such as "`publish together`" or "`2 page spread`", and the referenced placements may come from possibly multiple Bookings.



The `AdMessageRequestModule` is media agnostic and can be used for any media. The media specific aspects are handled in one or more elements specifying placements whose names indicate that they include media specific data, for example, `Placement.NewspaperMagazine`. Different placements do not need to be related in any other way than being booked by the same booking party and sold by the same selling party.

Placements can be grouped to form packages within a package. See `PlacementGroup` for further details.

In addition, request messages can also sometimes include additional, message specific, content. For more information, see the specific message element definitions.

Attributes

No attributes.

3.3.4 Element Group: AdMessageResponseModule

The `AdMessageResponseModule` element group is a reusable assembly of elements that carry the information needed by response messages. All business level response messages use this module, while some also include additional, message specific, content elements.

The overall approach to responses is to include the same data structures as in the requests. This means that the `AdMessageResponseModule` element group is very similar to the `AdMessageRequestModule`, but with a set of extra elements:

If a request was denied, the `adsml:RequestDenied` element must be used to specify the business reason(s) as to why the request could not be fulfilled. In this case no other information is provided.

A response, other than a denial, may either be a complete acceptance of the terms in the prior request, or include changes compared to the terms in the request. The mandatory `adsml:NatureOfResponse` element must be used to indicate this. It should take values from an agreed code list with typical values such as "`AcceptedAsRequestedByBuyer`" or "`AcceptedWithChangesBySeller`".

In case the response is a response-with-changes, a specification of the changes made is **RECOMMENDED** to be provided in the optional `adsml:ChangeSpecification` element. This mechanism is also used in the same manner in change requests.

The `adsml:Status` element includes the current status of the booking within the systems of the responder.

For more information, see the `AdMessageRequestModule` and the message element definitions.

Attributes

No attributes.

3.3.5 Element Group: AdMessageStatusModule

The `AdMessageStatusModule` element group is a reusable assembly of elements that carry the information needed by status response messages.

The overall approach to responses is to include the same data structures as in other response messages. This means that the `AdMessageStatusModule` element group is similar to the `AdMessageResponseModule`, with the following exceptions: The `adsml:NatureOfResponse` and `adsml:ChangeSpecification` elements are not used.

For more information, see the `AdMessageResponseModule` and the message element definitions.

Attributes

No attributes.

3.3.6 Element: AdServingSystem

The `AdServingSystem` element is used within the interactive domain to record a code identifying the system that will be used to manage the run of the ad.

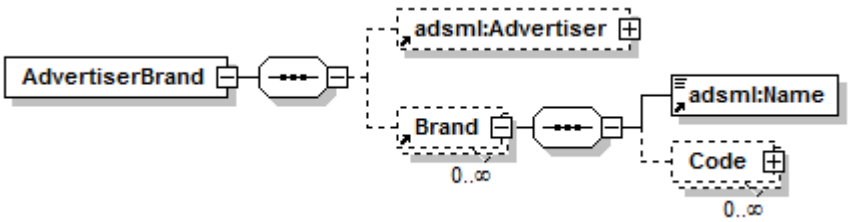
See `ProductionDetail.Interactive` for further information.

Attributes

No attributes.

3.3.7 Element: AdvertiserBrand

The `AdvertiserBrand` element specifies a package of an advertiser and a set of brands (or products) that the advertiser is advertising in a placement.



In order to support situations where either of advertiser or brand(s) is not known, both `adsml:Advertiser` and `Brand` child elements are optional.

Note that if the same advertiser and brand(s) are related to more than one placement in a booking, this information should be repeated in each such placement.

Attributes

No attributes.

3.3.8 Element: AlternativePositioning

See `Positioning`.

Attributes

adsml:priority (optional)

The `adsml:priority` attributed may be used to define the priority order when more than a single `AlternativePositioning` element exists. A lower value **MUST** be interpreted according to the rules defined for `adsml:PriorityType`.

3.3.9 Element: Area

The `Area` element records an area value as a decimal. It is defined as an `adsml:DecimalMeasurementType`.

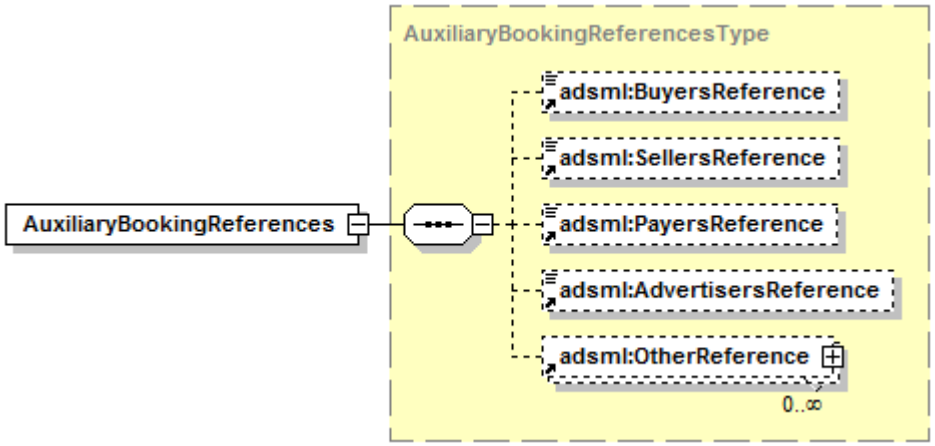
See `Size` for more information.

Attributes

No attributes.

3.3.10 Element: AuxiliaryBookingReferences

The `AuxiliaryBookingReferences` element is used in many contexts, such as the overall booking or placements, for recording references to the related object that are additional to that object's primary identifier. For instance, the overall booking has a `BookingIdentifier` as its primary identifier. The `AuxiliaryBookingReferences` element can then be used to record additional references to the booking assigned by the buyer, the advertiser or any other party engaged in the transaction.



The buying and selling parties can use the optional `adsml:BuyersReference` and `adsml:SellersReference` to record their reference values. The `adsml:PayersReference` and `adsml:AdvertisersReference` play the same roles for the paying and advertiser parties, respectively.

Other references may be recorded in the repeatable `adsml:OtherReference` element providing additional information about who in the referenced party should be used for communications, as well as other data. Please see `adsml:OtherReference` for more information.

Attributes

None.

3.3.11 Element: AuxiliaryCampaignReferences

The `AuxiliaryCampaignReferences` element is used to record references to a campaign. Its content model includes `adsml:BuyersReference` and `adsml:AdvertisersReference` to record their reference values.

Other references may be recorded in the repeatable generic `adsml:OtherReference`. Please see `adsml:OtherReference` for more information.

Attributes

None.

3.3.12 Element: AuxiliaryPlacementReferences

The `AuxiliaryPlacementReferences` element is used to record references to a placement. It has the same content model as `AuxiliaryBookingReferences`, please see its description for further information.

Attributes

None.

3.3.13 Element: AuxiliaryPlacementGroupReferences

The `AuxiliaryPlacementGroupReferences` element is used to record references to a placement group. It has the same content model as `AuxiliaryBookingReferences`, please see its description for further information.

Attributes

None.

3.3.14 Element: BillingInstructions

The `BillingInstructions` elements records how invoices should be calculated and/or distributed based on time, volume or other measures when the length of a placement spans more than one billing cycle.

The `Frequency` child element describes how often the booking should be billed, the `Allocation` child element describes the allocation of money over payments, and `Basis` records the basis for the allocation.

A generic `AdditionalBillingInstructions` is also available.

All child elements are optional, and defined as `adsml:RequirementSpecType`.

Attributes

None.

3.3.15 Element: Bleed

The `Bleed` element allows the recording of a code and description indicating how the ad should be treated when placed in the publication. Bleed is the term used when the area the ad covers extends beyond the 'normal' boundaries of the page.

As the element is derived from `adsml:CodeRootType`, it can use a controlled vocabulary for validation

Attributes

No attributes.

3.3.16 Element: BookingDate

The `BookingDate` element allows the recording of a business significant booking date (or order date) for the booking. This date may be used in addition to other dates such as *`messageAssembledDateTime`* and *`transmissionDateTime`* which records dates that may be different from the booking date from a business standpoint.

Attributes

No attributes.

3.3.17 Element: BookingIdentifier

The `BookingIdentifier` element declares the primary identifier for the booking. It is defined as an `adsml:QIDType`. See the Section "*Message References – Booking and Quotation Identifiers"* for detailed information about the use of identifiers.

Attributes

No attributes.

3.3.18 Element: Brand

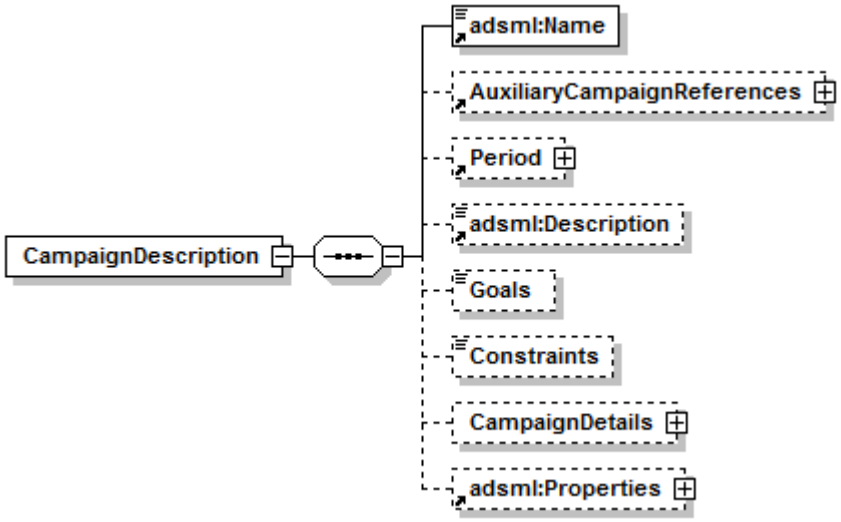
The `Brand` element is used to record data about the goods or services that are advertised. It includes a mandatory `adsml:Name` and may also have a set of `Code` elements defined as `adsml:CodeType`.

Attributes

No attributes.

3.3.19 Element: CampaignDescription

The `CampaignDescription` element includes a set of child elements that can be used to convey descriptions of different aspects of an advertising campaign. The data captured is mainly intended for human consumption.



The mandatory `adsml:Name` element should record the name of the campaign. The `AuxiliaryCampaignReferences` can hold additional references related to important parties in the business workflow.

The `Period` element records a time period during which the campaign is running. It may be provided as exact start and end dates, or as a duration such as '2 months'.

The `adsml:Description`, `Goals` and `Constraints` elements can be used to record free text describing the campaign's objectives, constraints and other information making the advertiser's overall intent with a booking available as a background for the ad selling organization.

The `CampaignDetails` element, defined as an `adsml:DocumentRenderingType`, can be used to reference an external document that provides additional information.

Attributes

adsml:i18nAttributes (optional)

The *`i18nAttributes`* group supports internationalization by providing attributes to record language, directionality and source.

3.3.20 Element: Cancellations

The `Cancellation` element holds free text. See `TermsAndConditions` for further information.

Attributes

No attributes.

3.3.21 Element: Cap

See `CappingSpecification` for information.

Attributes

No attributes.

3.3.22 Element: CappingSpecification

Capping is a set of constraints that limit the number of times an advertisement may be published within a defined scope. Capping instructions are recorded in the `CappingSpecification` element where they can be expressed either as purely textual instructions in the `adsml:Description` element, or as a stack of one or more machine-processable `Cap` elements.

The `adsml:Description` may be repeated to provide a description of the capping specification in alternative languages, but **MUST NOT** be repeated for any other reason.

Each `Cap` child element defines a capping constraint as a maximum number of events (`MaximumEventCount`) of a given event type (`EventType`) within a scope (`Scope`). The event type may be the same as the Placement target's event type, but this is not required. If two or more scopes are provided within a given *`Cap`*, they **MUST** be combined with AND logic.

This feature is frequently used in the interactive domain. For example, an ad may be capped to be shown "no more than 5 times to the same visitor" during the run of the ad, or more commonly, "no more than twice to the same visitor on the same day."

Attributes

No attributes.

3.3.23 Element: Cash

The `Cash` element is used to describe a payment that has been received involving physical currency. This will generally be recorded as a `Reference` to a Cash Book item or some other reconciliation method. The method can be described using the optional `Label` child element.

See `Payment` for more information.

Attributes

No attributes.

3.3.24 Element: Check

The `Check` construct is used to describe where payment has been made by check. The check number is recorded in the `CheckNumber` element.

See `Payment` for more information.

Attributes

No attributes.

3.3.25 Element: CheckNumber

The `CheckNumber` element records the unique identifying number on the check provided in the `Check` element.

See `Payment` for more information.

Attributes

No attributes.

3.3.26 Element: ClaimReference

The `ClaimReference` element holds a textual reference to a claim. See `CostExempt` for further information.

Attributes

No attributes.

3.3.27 Element: Claims

The `Claims` element holds free text. See `TermsAndConditions` for further information.

Attributes

No attributes.

3.3.28 Element: ClassifiedPlacementCode

See `PlacementInBook` for information.

Attributes

No attributes.

3.3.29 Element: Code

The `Code` element is used to capture codes. It is based on the `adsml:CodeType`.

Attributes

No attributes.

3.3.30 Element: ColorName

The `ColorName` element records the name of a color as a string.

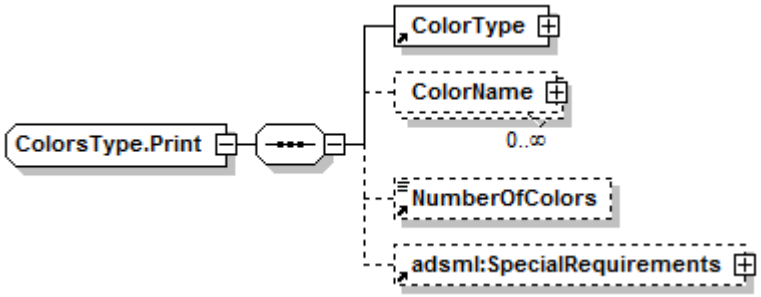
See `Colors` for more information.

Attributes

No attributes.

3.3.31 Element: Colors

The `Colors` element should be used to specify colors booked for an advertisement. The color specification is not intended to provide all details required for print production, but enough information to support price calculation, and production planning.



The structure contains four elements, only the first of which is mandatory:

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

Attributes

No attributes

3.3.32 Element: ColorType

The `ColorType` element is used to record the type of color in a color specification. As this element is based on a `adsml:CodeType`, it can use a controlled vocabulary for validation of both children elements `adsml:CodeList` and `adsml:CodeValue`. An optional text `adsml:Description` of the color type may also be provided.

See `Colors` for more information.

Attributes

No attributes.

3.3.33 Element: Constraints

See `CampaignDescription` for further information. Attributes

No attributes.

3.3.34 Element: CostExempt

The `CostExempt` element enables trading partner to indicate that a Booking, or part of a Booking, is cost exempt, along with the associated reason. This supports workflows in which "free" ads are handled differently than paid ones.

The mandatory `ExemptionType` child element must be used to provide a type code for the cost exemption. A set of reasons for the Booking being cost exempt can be recorded in the `adsml:ExemptionReason` element.

In case the booking is cost exempt due to a makegood for a claim based on an earlier Booking, or Placement within a Booking, a reference can be provided in the `PlacementReference` element. Also, a reference to the claim may be provided in the `ClaimReference` element.

Attributes

No attributes.

3.3.35 Element: CuttablePosition

The `CuttablePosition` elements records if a placement in a print media is cutable, i.e. can be cut without destroying an ad at the other page.

See the `Positioning` element description, `Placement.NewspaperMagazine` context.

Attributes

No attributes.

3.3.36 Element: DealCode

A `DealCode` element can be used to specify if a booking, placement group or placement is requested to be made according to terms in a pre-defined package, or deal. Reference to a deal can be used to replace specific details regarding publication, production detail and/or insertion schedules. It is defined as an `adsml:CodeType`.

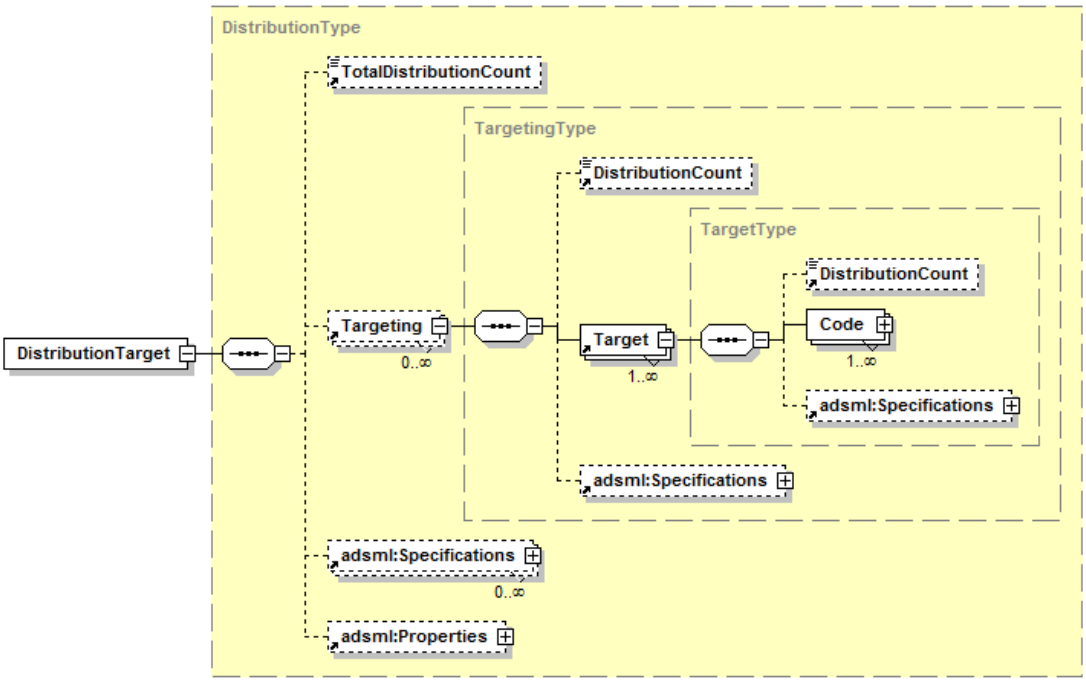
Attributes

No attributes.

3.3.37 Element: DistributionTarget

The `DistributionTarget` element can be used to record in which editions, regions, zones or other slice of a publication (or sub-publication) the ad should run, and how large a volume of circulation should be distributed to those slices.

The `Targeting` element allows the buyer to restrict the distribution of the advertisement to a subset of the audience, the demographic population, that the publication is capable of reaching. This includes, but is not limited to, temporal (e.g. editions, etc.), geographic (e.g. regions, zones, micro-zones, etc.), distribution (e.g. subscription, newsstand, etc.) and demographic (e.g. gender, age, income, etc.) targets. If `Targeting` is not present, the ad **MUST** be expected to run in all parts of the publication or sub-publication.



The targeting mechanism consists of an optional and repeatable set of `Target` elements, each of which contains one or more `Code` constructs. Each `Code` conveys three pieces of information: an `adsml:CodeValue` (e.g. "Northwest" or "Early City" or "Newsstand"), the name of the `adsml:CodeList` from which the value was taken (e.g. "Regions" or "Editions" or "Distribution"), and an optional *`negated`* attribute.

`Codes` for which the *`negated`* attribute is provided and set to true identify target points (i.e. regions, editions, distributions or demographic groups) that should be excluded from the resulting set. All other `Codes` identify target points that should be included in the set.

Each of the three levels, `DistributionTarget`, `Targeting` and `Target`, includes the ability to specify a distribution count that is the target distribution volume to the zones, editions, demographic groups etc specified in the structure beneath. `Targeting` and `Target` both have `DistributionCount` child elements while a `TotalDistributionCount` is provided in the overall `DistributionTarget` element.

Guidelines, examples and mandatory processing rules for use of the `Targeting` structure are provided in the *Targeting* chapter of *AdsMLBookings – Usage Rules & Guidelines*.

See `Targeting` for more information.

Attributes

No attributes.

3.3.38 Element: DistributionCount

The `DistributionCount` element records a number that determines the number or amount of a publication's distribution.

It is used to specify a distribution target within a `Distribution` structure. See `Distribution` for more information.

Attributes

No attributes.

3.3.39 Element: EventCount

The `EventCount` element is used in several contexts to record a number of events.

See for instance `PlacementTarget` or `SchedulingType` for further information.

Attributes

No attributes.

3.3.40 Element: EventType

The `EventType` element is used in several contexts to define a type of event.

See for instance `PlacementTarget` or `Cap` for further information.

Attributes

No attributes.

3.3.41 Element: ExemptionType

The `ExemptionType` element is defined as an `adsml:CodeType`. See `CostExempt` for more information.

Attributes

No attributes.

3.3.42 Element: FirstPossibleTime

See `SchedulingType` for more information.

Attributes

No attributes.

3.3.43 Element: General

The `General` element holds free text. See `TermsAndConditions` for further information.

Attributes

No attributes.

3.3.44 Element: Goals

See `CampaignDescription` for further information.

Attributes

No attributes.

3.3.45 Element: Gutter

The `Gutter` element records an optional value as a decimal and unit of measure. It is defined as an `adsml:OptionalDecimalMeasurementType`.

See `Size` as used in the `Placement.NewspaperMagazine` context for more information.

Attributes

included (required)

Indicates if the gutter is included in the ad size or not

3.3.46 Element: Height

The `Height` records a height value as a decimal together with its unit of measure. It is defined as an `adsml:DecimalMeasurementType`.

Attributes

No attributes.

3.3.47 Element: InsertionPeriod

The `InsertionPeriod` captures requirements on insertion dates, i.e. when an ad should run.

See `SchedulingType` for more information.

Attributes

No attributes.

3.3.48 Element: Invoice

The `Invoice` element is marked for sending an invoice to the party identified as `Payer` with possible additional copies sent to third-parties.

Where the invoice copy is to be sent to a third-party or, for example, a different department within the payer's company, the `SendCopyTo` element should be used to record the details. `SendCopyTo` is a `adsml:RelaxedPartyType`.

The `MarkWith` element is used to record any text that the buyer requires the seller to put on the invoice in order to aid reconciliation once it reaches the payer.

The `Invoice` element captures special invoicing instructions in a `Payment` using the `adsml:Instructions` element, which is based on the `adsml:RequirementSpecType`.

Attributes

sendToPayer (fixed: 'true')

Defines that an invoice should be sent to the party identified as payer. See `Payer`.

3.3.49 Element: IsStandAlone

The `IsStandAlone` element defines whether the placement can be treated as a standalone placement, or if it is related to one or more other placement(s).

When the value is `'true'`, the placement represents a single individual advertisement without links to or from other placements. In any preceding or following booking, there **MUST NOT** be any `LinkedPlacement` element referring to this placement.

If the value is `'false'`, it is not stand alone and the type of relationship **SHOULD** be defined using the `LinkedPlacement` element. Only in the case where the related placement(s) will arrive later and their identifiers are thus still unknown should the `LinkedPlacement` not be used.

Note that the fact that a set of placements appear in the same booking, or placement group, is not a criteria for being related. If not related via usage of the `LinkedPlacement` structure, `IsStandAlone` should be `"false"`.

Attributes

No attributes.

3.3.50 Element: LastPossibleTime

See `SchedulingType` for more information.

Attributes

No attributes.

3.3.51 Element: LetterOfCredit

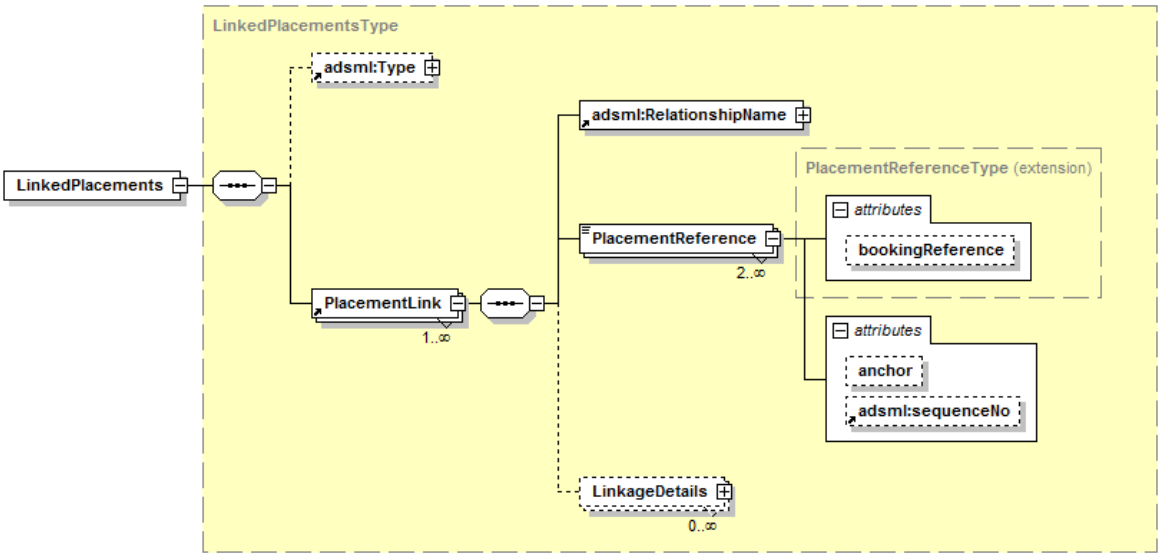
`LetterOfCredit` is used within the Payment `Mechanism` element to indicate that there has been an agreement recorded between the Buyer or Payer and the Seller that records that the Payer already has paid or has the ability to pay not otherwise recorded elsewhere. It consists of a `Reference` element.

Attributes

No attributes.

3.3.52 Element: LinkedPlacement

The element `LinkedPlacements` can be used to declare any type of relationship between placements in a booking. Its main purpose is to handle linked ads that appear as separate placements within a booking, or even as placements in different bookings. Examples include so-called "umbrella" ads in which a container ad includes space for other ads that can arrive later, often from other advertisers, or groups of ads that are requested to appear together or not at all.



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

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

See the *Linked Placement* section in *AdsMLBookings 2.5 Part 1, Usage Rules & Guidelines* for more details.

Attributes

No attributes.

3.3.53 Element: MarkWith

The `MarkWith` element is for text that the buyer asks the seller to put on the invoice to assist in the proper handling of that invoice once it goes to the payer. This might be explicitly for reconciliation or any other purpose that the buyer has in mind.

See `Invoice` for further information.

Attributes

No attributes.

3.3.54 Element: MaterialsExpectations.Insert

The `MaterialsExpectations.Insert` element is an extension of `adsml-ma:MaterialsExpectations` adding the `RequiredOverage` element, used for Inserts.

See `Placement.Insert` for further information.

Attributes

No attributes.

3.3.55 Element: MaterialsReference

The `MaterialsReference` element is a reference to an `adsml-ma:AdContent` instance used for doing pickups.

See `PickUp` for further information.

Attributes

No attributes.

3.3.56 Element: MaximumEventCount

The `MaximumEventCount` element is used in several contexts to record a maximum number of events.

See for instance `PlacementTarget` or `Cap` for further information.

Attributes

No attributes.

3.3.57 Element: Mechanism

The `Mechanism` element is used to indicate by which method the payment is/has been received. It is a choice between the following elements:

- `adsml:CreditCard`
- `Invoice`
- `Cash`
- `Check`
- `LetterOfCredit`
- `AccountDebit`
- `Contract`
- `OtherMechanism`

See `Payment` for further information.

Attributes

No attributes.

3.3.58 Group: MessageFooterGroup

The MessageFooterGroup includes a set of elements used in all AdsMLBookings transactions messages.

The `adsml:DocumentRendering` element allows the sender of a business document to convey a digital rendering of the document either by containership (e.g. a PDF is embedded in the message) or reference (a URL or equivalent is provided so that the recipient can automatically retrieve the rendering). The

element may be repeated for documents in alternative languages, but **MUST NOT** be repeated for any other reason. The element supports the *adsml:i18nAttributes* group for language metadata. Data in these attributes refer to the language in the rendered document.

Additional requirements relating to the transaction as a whole can be recorded in the `adsml:SpecialRequirements` structure.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Human communication between e.g. buyer's and seller's sales rep's regarding the booking may be recorded using the `adsml:Notes` element. Although convenient, it is strongly **RECOMMENDED** not to use this element for important bookings data that can be recorded elsewhere in a more formal way.

Attributes

No attributes.

3.3.59 Element: MinimumEventCount

The `MinimumEventCount` element is used in several contexts to record a minimum number of events.

See for instance `PlacementTarget` for further information.

Attributes

No attributes.

3.3.60 Element: MinimumAdditionalPieces

See `RequiredOverage` for information.

Attributes

No attributes.

3.3.61 Element: MultipleAdContentHandling

The `MultipleAdContentHandling` element describes how placements involving more than one set of ad materials should be handled.

See also `ProductionDetail.NewspaperMagazine` or `ProductionDetail.Generic`.

Attributes

No attributes.

3.3.62 Element: NumberOfColors

The `NumberOfColors` records the number of colors in a color specification.

See Colors for more information.

Attributes

No attributes.

3.3.63 Element: NumberOfCopies

The `NumberOfCopies` captures a number of copies.

Attributes

No attributes.

3.3.64 Element: NumberOfPages

The `NumberOfPages` captures the number of pages in for instance an insert.

Attributes

No attributes.

3.3.65 Element: OtherMechanism

The `OtherMechanism` element specifies additional payment methods in a `Payment` using the general `adsml:Properties` structure.

See `Payment` and `adsml:Properties` for more information.

Attributes

No attributes.

3.3.66 Element: OveragePercentage

See `RequiredOverage` for information.

Attributes

No attributes.

3.3.67 Element: Page

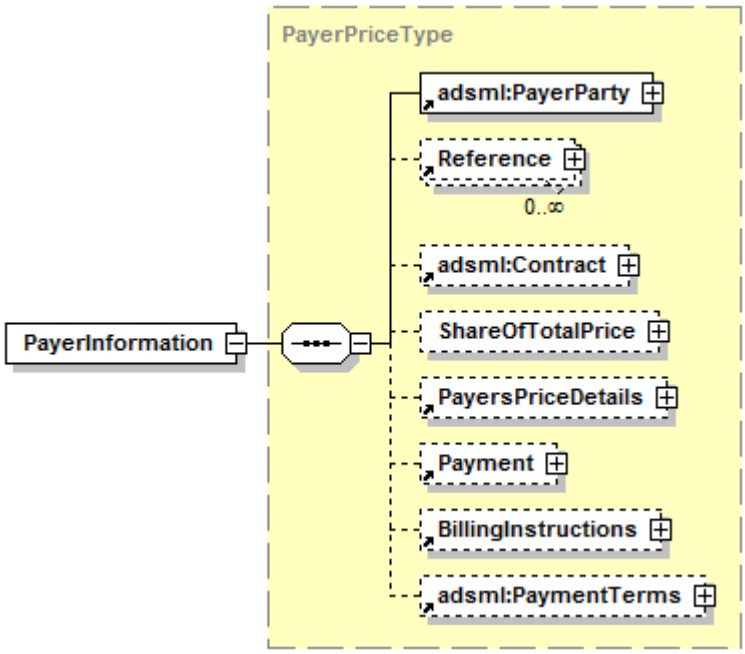
The `Page` element captures a page number.

Attributes

No attributes.

3.3.68 Element: PayerInformation

The `PayerInformation` structure defines the price for a specific payer including payment, billing instructions, payment terms and contact information.



The `adsml:PayerParty` element records the organization that should pay/have paid the price in the `Price` element. Note that `PayerInformation` is intended to be flexible and support several different use cases recording, for instance, a price to be invoiced by the publisher to a payer, a price already paid by the payer to the booking party and included only for reconciliation purposes, etc. The exact usage has to be mutually agreed by the trading partners.

The `Reference` element can be used to include a set of references for the payment, per payer. Note however that specific elements for references to Purchase Orders and Contracts are located elsewhere and **SHOULD** be used if per payer references are not needed.

The `adsml:Contract` element allows reference to a particular contract for the payer, including sections and rates within that contract.

In cases with multiple payers, the `ShareOfTotalPrice` specifies how the price is divided between the payers.

The price for this payer, including discounts and other price components, is specified in the `PayersPriceDetails` element.

The `Payment` element includes instructions on how to collect the payment from the payer.

In case a booking involves insertions during a longer period of time, it is possible to use the `BillingInstructions` elements to record how payments should be distributed over time, volume or other measures.

Detailed payment terms can be recorded in the `adsml:PaymentTerms` element, including due date, penalty periods and surcharges.

Note that the `PayerInformation` in this version of AdsMLBookings does not particularly support refunds and credits.

Attributes

No attributes.

3.3.69 Element: PayersPriceDetails

The `PayersPriceDetails` element is an optional element used in the `PayerInformation` element to record the price for a particular payer. It is optional to allow specification of just the payer organization, but not the price it will pay, in an `AdOrder` Request.

The `PayersPriceDetails` element can be used in several different contexts.

See also `adsml:CurrencyPriceDeclarationType` for more information about the usage of `PayersPriceDetails`.

Attributes

No attributes.

3.3.70 Element: Payment

The `Payment` element is used to provide payment details about the nature and mechanism of the payment associated with this booking or placement.

There are four elements within the `Payment` structure:

- An optional status (`adsml:Status`), which indicates the status of the payment (e.g. PrePaid, ToBeCollected)
- A `Mechanism`, which defines the means by which the payment either has or is to be collected.
- `Receipt` determines if the payer (or other third-party) requires a receipt to be issued, and where the receipt should be sent. This will mainly be used for pre-paid or credit card payments.
- Lastly, `AdditionalPaymentInstructions`, defined as an `adsml:RequirementsSpecType`, supports recording further instructions using codes or texts as appropriate.

Attributes

No attributes.

3.3.71 Element: Period

The `Period` element records a time period. It is defined on the `adsml:PeriodType`.

Attributes

No attributes.

3.3.72 Element: PickUp

The `Pickup` element records 'pickup' identifiers and instructions for existing ad content that is to be 'picked up' and reused. Data is recorded using a required `PlacementReference` element that must point to the placement where the ad content was used before. The optional `MaterialsReference` element can be used to further specify a reference to the ad content within that placement.

An optional `adsml:Instructions` element may be used to record additional information or instructions relevant to making the pickup.

See the `adsml:Instructions` element definition for more information.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

withChange (default: false)

A Boolean value recording whether the artwork identified by the pickup is to be changed or modified in some way by the recipient. When set to `'true'`, the artwork is to be changed or modified according to the instructions specified in the `adsml:Instructions` element child of the `Pickup` element.

3.3.73 Element: PickUp.Insert

The `Pickup.Insert` element records 'pickup' identifiers and instructions for existing materials that is to be 'picked up' and reused for an insert. Data is recorded using a required `PlacementReference` element that must point to the placement where the ad content was used before.

An optional `adsml:Instructions` element may be used to record additional information or instructions relevant to making the pickup.

See the `adsml:Instructions` element definition for more information.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

No attributes.

3.3.74 Element: Placement

The `Placement` element includes the media independent parts of a placement. It is an abstract element defined as a `PlacementType.Base` and can thus not be directly used in a message, but would have to be substituted with media-specific extensions such as `Placement.Generic` or `Placement.NewspaperMagazine`.

A placement must be given at least one primary identifier using the `PlacementIdentifier` element. The identifier is assigned by whoever first describes the placement in an AdsMLBookings message. Optional alternative identifiers can be provided using the `AuxiliaryBookingReferences` element.

The mandatory `IsStandAlone` element must be used to declare whether the placement is a standalone placement or a placement that is linked to or from other placement(s).

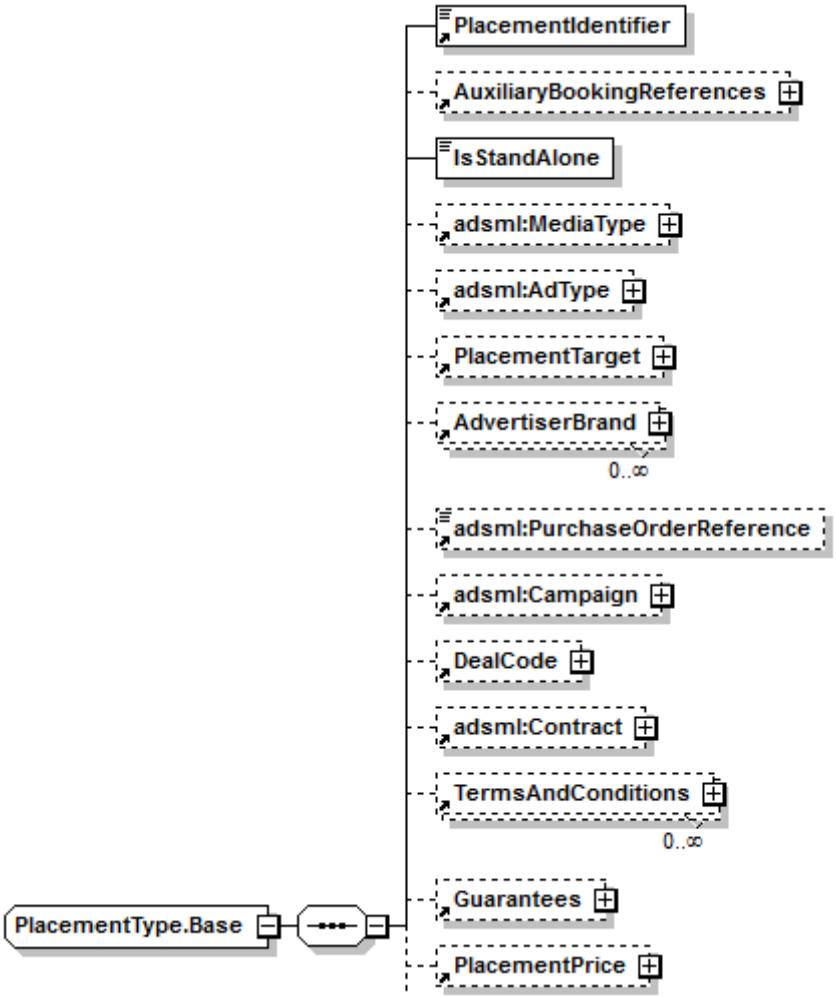
The `adsml:MediaType` may be used to indicate what type of advertising media the placement is specifying. Typical values are newspaper, outdoor, insert or magazine.

The `adsml:AdType` may be used to indicate what type of advertisement the placement is specifying. Typical values are insert, display, classified display or classified liner ad.

The role of the `PlacementTarget` element is to, when needed, explicitly define the *target event*, the "thing" booked and eventually paid for in a Placement. This is the primary goal of the order, the result for which the buyer is willing to pay a fee to the seller.

Based on type of advertisement and media, the target event is more or less implicitly understood. For instance, for a print display ad booking, the buyer books and pays for the insertion of an ad in the newspaper. But other media and advertisement types have less obvious targets, which is especially true within the interactive domain where the target might be "1 000 000 impressions" or "100 click-throughs" or "10 registrations".

With the `PlacementTarget` structure available across all Placement types, it is possible to define the target explicitly, including the target count of such events for this placement.



The advertisers and the brand(s) (or "products") they advertise in the ad can be listed using the repeatable `AdvertiserBrand` element.

A reference to a purchase order can be included using the optional `adsml:PurchaseOrderReference` element.

If the placement should be linked to a campaign (also known as an "estimate" in some regions), use the `adsml:Campaign` element to record a name for the campaign.

An optional `DealCode` element can be used to specify whether the placement is requested to be made according to terms in a pre-defined package, or deal.

Reference to a deal can be used to replace specific details regarding publication, production detail and/or insertion schedules.

An `adsml:Contract` structure allows for explicit identification of the "rate" or "level" that applies to a given Placement. It may also be specified at the booking and placement group levels.

The `TermsAndConditions` element holds a set of human-readable textual terms for aspects such as cancellations, claims and disclaimers and any other legalistic information that applies to the placement. `TermsAndConditions` may be repeated for information in alternative languages, but **MUST NOT** be repeated for any other reason.

The `Guarantees` element can be used to record that certain aspects of the placement have been "Guaranteed", and to convey the nature of that guarantee as a set of codes or text strings as appropriate.

The total price for the placement can be recorded in the `PlacementPrice` element. Also, the price to be paid by one or more payers can be declared using the `PayerInformation` element, which also includes structures for recording payment instructions.

If the placement is provided for free, the `CostExempt` element should be used to record the type and reason. The element **MUST** be omitted when the placement is not cost exempt.

A short descriptive text line intended for use as a human reference should be provided in the `adsml:DescriptionLine` element. It may be repeated to capture the description in alternative languages, but **MUST NOT** be repeated for any other reason.

The `PrePublicationProof` element allows specification of a set of parties that should receive proofs of the ad (i.e. approval copies) from the publisher before publication.

The `ProofOfPublication` element allows specification of a set of parties that should receive proof of publication (i.e. tearsheets) from the publisher after publication.

The current status of the placement within the sender's systems can be provided using the optional `adsml:Status` element. This element **SHOULD** only be included in response messages.

The `adsml:DocumentRendering` element allows the sender of a business document to convey a digital rendering of the `Placement` either by containership (e.g. a PDF is embedded in the message) or reference (a URL or equivalent is provided so that the recipient can automatically retrieve the rendering). The element may be repeated for documents in alternative languages, but **MUST NOT** be repeated for any other reason. The element supports the *`adsml:i18nAttributes`* group for language metadata. Data in these attributes refer to the language in the rendered document.

Additional requirement relating to the transaction as a whole can be recorded in the `adsml:SpecialRequirements` structure.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Human communication between e.g. buyer's and seller's sales rep's regarding the placement may be recorded using the `adsml:Notes` element. Although convenient, it is strongly **RECOMMENDED** not to use this element for important bookings data that can be recorded elsewhere in a more formal way.

Attributes

No attributes.

3.3.75 Element: Placement.Generic

The `Placement.Generic` element is an extension of the `PlacementType.Base` type, with generic placement data for any media type not directly supported by other media-specific placement elements.

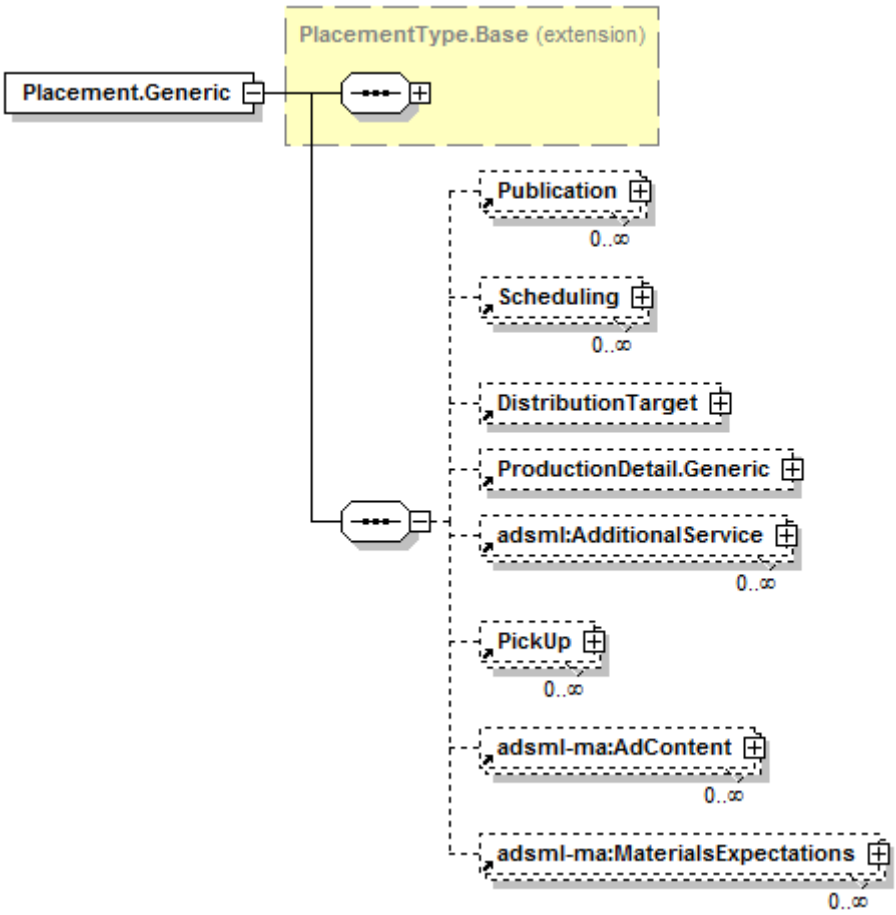
It extends the base placement the following main areas:

The repeatable `Publication` element defines the publication(s), sites(s) or program(s) where the ad should run. Note that if more than one `Publication` is recorded, all other information recorded within the `Placement` **MUST** be applicable to all of those publications.

`Scheduling` is a repeatable element that defines a time point or range in which the ad should run. The set of schedules are common to all instances of `adsml-ma:AdContent`. If a different `adsml-ma:AdContent` would be required for a specific schedule, this **MUST** be expressed using another `Placement.Generic`.

The `DistributionTarget` element can be used to record in which editions, regions, zones or other slice of the publication (or sub-publication) the ad should

run as well as the distribution count. It can also be used to specify demographic populations that this placement is intended to reach. If the child element `Targeting` is not present, the ad **MUST** be expected to run in all parts of the publication or sub-publication.



`ProductionDetail.Generic` includes a set of elements such as `Positioning` that defines other aspects of the ad's production process.

The `adsml:AdditionalService` element may be used to specify additional services that the publisher provides in relation to the ad insertion. The buyer may be invoiced for the service, and referenced from a `PriceComponent`.

The `PickUp` element may be used to specify that previously delivered artwork should be used in the placement.

The `adsml-ma:AdContent` element specifies the properties of the artwork that should be placed. It includes identification and naming of ad artwork files, as well as their physical properties, who should deliver the files, due dates for delivery etc. The `adsml-ma:AdContent` data includes the required data to perform matching of materials with bookings, including copy chasing if needed.

A `Placement.Generic` can include multiple `adsml-ma:AdContent` elements. Note however that they all need to share all other properties of the placement such as scheduling and production detail. If two `adsml-ma:AdContent` would require, for instance, different sizes, two separate placements **MUST** be provided.

The `adsml-ma:MaterialsExpectations` element can be used to record metadata relating to a set of ad materials that will be used for this placement but either is not available as part of the `adsml:AdContent` or was delivered earlier.

The structure contains some information that will typically be provided by an advertiser (or their agents) and other information typically provided by a publisher. See `adsml-ma:MaterialsExpectations` description for further information.

See also `Placement`.

Attributes

No attributes.

3.3.76 Element: Placement.Insert

The `Placement.Insert` element is an extension of the `PlacementType.Base` type, with media-specific content primarily for inserts describing the requirements on the ads production.

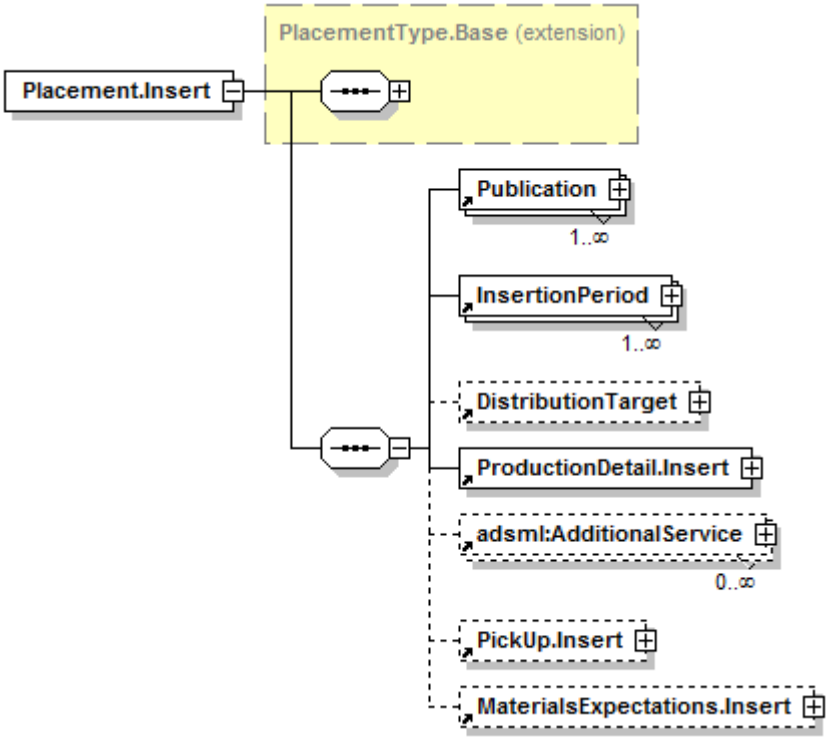
It extends the base placement in following areas:

The repeatable `Publication` element defines where the ad should run. Note that if more than one `Publication` is recorded, all other information recorded within the `Placement` **MUST** be applicable to all of those publications.

The `InsertionPeriod` is a repeatable element that defines time periods in which a specified number of insertions should take place.

The `DistributionTarget` element can be used to record in which editions, regions, zones or other slice of the publication (or sub-publication) the insert should be distributed as well as the distribution count. It can also be used to specify demographic populations that this placement is intended to reach. If the child element `Targeting` is not present, the ad **MUST** be expected to be distributed according to the full distribution of the publication or sub-publication.

`ProductionDetail.Insert` includes a set of elements such as `Size` and



`NumberOfPages` that defines other aspects of the insert's publication in a newspaper or magazine.

The `adsml:AdditionalService` element may be used to specify additional services that the publisher provides in relation to the ad insertion. The buyer may be invoiced for the service, and referenced from a `PriceComponent`.

The `PickUp.Insert` element may be used to specify that already delivered materials should be used in the placement.

`MaterialsExpectations.Insert` is an extension of the `adsml-ma:MaterialsExpectations` with a `RequiredOverage` element that can record the overage required to be available for distribution of Inserts.

See also `Placement`.

Attributes

No attributes.

3.3.77 Element: Placement.Interactive

The `Placement.Interactive` element is an extension of the `PlacementType.Base` type, with media-specific content primarily for interactive banner and rich media ads describing the requirements on the ads' production.

It extends the base placement the following main areas:

The repeatable `Publication` element defines the publication(s), sites(s) or program(s) where the ad should run. Note that if more than one `Publication` is recorded, all other information recorded within the `Placement` **MUST** be applicable to all of those publications.

`Scheduling` is a repeatable element that defines a time point or range in which the ad should run. The set of schedules are common to all instances of `adsml-ma:AdContent`. If a different `adsml-ma:AdContent` would be required for a specific schedule, this **MUST** be expressed using another `Placement.Interactive`.

The `DistributionTarget` element can be used to record in which editions, regions, zones or other slice of the publication (or sub-publication) the ad should run as well as the distribution count. It can also be used to specify demographic populations that this placement is intended to reach and the associated distribution count. If the child element `Targeting` is not present, the ad **MUST** be expected to run in all parts of the publication or sub-publication.

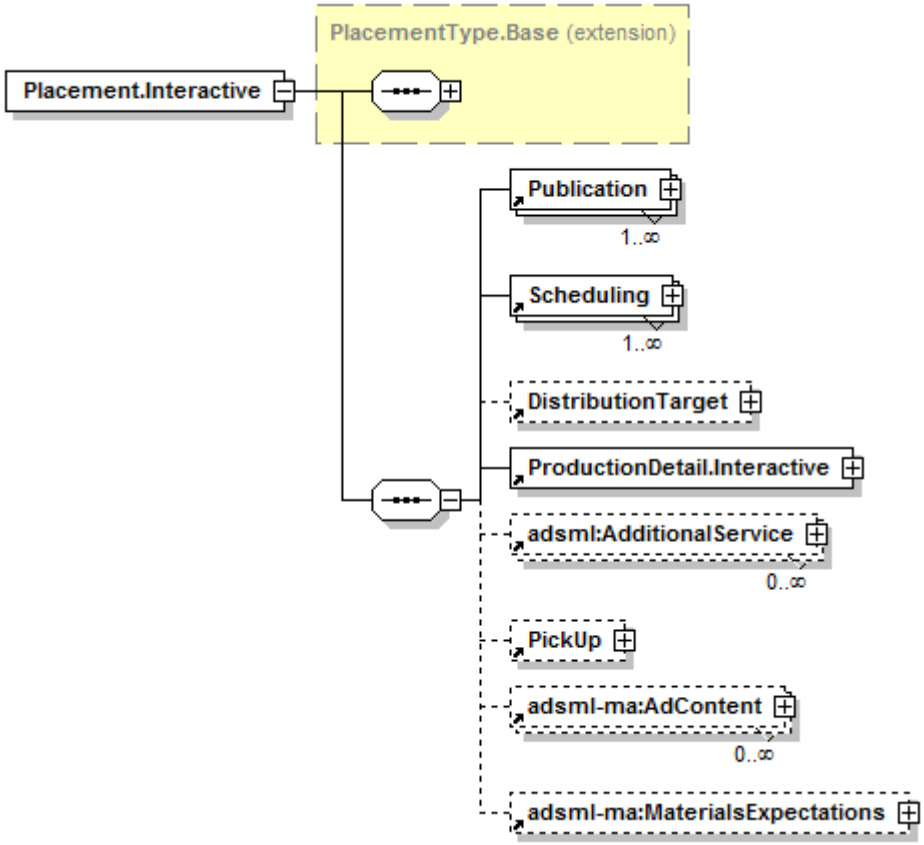
`ProductionDetail.Interactive` includes a set of elements such as `Positioning` and `CappingSpecification` that define other aspects of the ad's production process.

The `adsml:AdditionalService` element may be used to specify additional services that the publisher provides in relation to the ad insertion. The buyer may be invoiced for the service, and referenced from a `PriceComponent`.

The `PickUp` element may be used to specify that previously delivered artwork should be used in the placement.

The `adsml-ma:AdContent` element specifies the properties of the artwork that should be placed. It includes identification and naming of ad artwork files, as well as their physical properties, who should deliver the files, due dates for delivery

etc. The `adsml-ma:AdContent` data includes the required data to perform matching of materials with bookings, including copy chasing if needed.



A `Placement.Interactive` can include multiple `adsml-ma:AdContent` elements. Note however that they all need to share all other properties of the placement such as scheduling and production detail. If two `adsml-ma:AdContent` would require, for instance, different sizes, two separate placements **MUST** be provided.

The `adsml-ma:MaterialsExpectations` element can be used to record metadata relating to a set of ad materials that will be used for this placement but either is not available as part of the `adsml:AdContent` or was delivered earlier. The structure contains some information that will typically be provided by an advertiser (or their agents) and other information typically provided by a publisher. See `adsml-ma:MaterialsExpectations` description for further information.

See also `Placement`.

Attributes

No attributes.

3.3.78 Element: Placement.NewspaperMagazine

The `Placement.NewspaperMagazine` element is an extension of the `Placement` element with media specific content primarily for newspapers and magazines describing the requirements on the ads production.

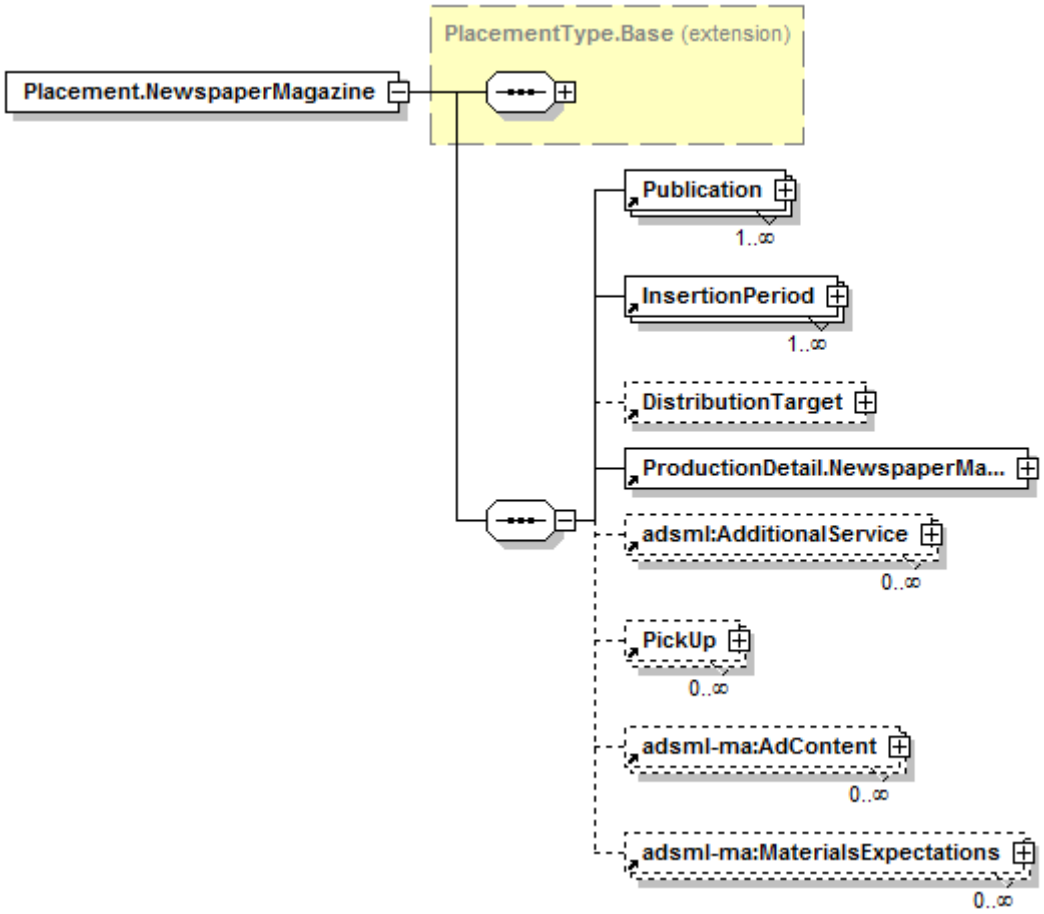
It extends the base placement in following areas:

The repeatable `Publication` element defines where the ad should run. Note that if more than one `Publication` is recorded, all other information recorded within the `Placement` **MUST** be applicable to all of those Publications.

The `InsertionPeriod` is a repeatable element that defines a time period in which a specified number of insertions should take place. The set of insertion dates are common to all instances of `adsml-ma:AdContent`. If a different `adsml-ma:AdContent` would be required for a specific insertion, this **MUST** be expressed using another `Placement.NewspaperMagazine`.

The `DistributionTarget` element can be used to record in which editions, regions, zones or other slice of the publication (or sub-publication) the ad should run as well as the distribution count. It can also be used to specify demographic populations that this placement is intended to reach. If the child element `Targeting` is not present, the ad **MUST** be expected to run in all parts of the publication or sub-publication.

`ProductionDetail.NewspaperMagazine` includes a set of elements such as `Size` and `Color` that defines other aspects of the ad's publication in newspapers and magazines.



The `adsml:AdditionalService` element may be used to specify additional services that the publisher provides in relation to the ad insertion. Blind boxes provided by the publisher could be such a service. The buyer may be invoiced for the service, and referenced from a `PriceComponent`.

The `PickUp` element may be used to specify that an already delivered artwork should be used in the placement.

The `adsml-ma:AdContent` element specifies the properties of the artwork that should be placed. It includes identification and naming of ad artwork files, as well as their physical properties, who should deliver the files, due dates for delivery etc. The `adsml-ma:AdContent` data includes the required data to perform matching of materials with bookings including copy chasing if needed.

A `Placement.NewspaperMagazine` can include multiple `adsml-ma:AdContent` elements. Note however that they all need to share all other properties of the placement such as insertion dates, size and other production detail. If two `adsml-ma:AdContent` would require, for instance, different sizes, two separate placements **MUST** be provided.

The `adsml-ma:MaterialsExpectations` element can be used to record metadata relating to a set of ad materials that will be used for this placement but either is not available as part of the `adsml:AdContent` or was delivered earlier. The structure contains some information that will typically be provided by an advertiser (or their agents) and other information typically provided by a publisher. See `adsml-ma:MaterialsExpectations` description for further information.

See also `Placement`.

Attributes

No attributes.

3.3.79 Element: PlacementGroup

The `PlacementGroup` element defines a group of placements within a booking. The main reason for creating a placement group as opposed to a plain list of placements is that the group can have a total price with associated campaign, purchase order, deal code and contract information for all placements in the group.

A placement group must be given an identifier using the `PlacementGroupIdentifier` element. Optional alternative identifiers can be provided using the `AuxiliaryBookingReferences` element.

A reference to a purchase order can be included using the optional `adsml:PurchaseOrderReference` element.

A reference to a campaign (also known as an "estimate" in some regions) that the placement group is linked to can be recorded as a formal code identifier using the `adsml:Campaign` element.

An optional `DealCode` element can be used to specify whether a placement group is requested according to terms in a pre-defined package, or deal. Reference to a deal can be used to replace specific details regarding publication, production detail and/or insertion schedules.

An `adsml:Contract` structure allows for explicit identification of the "rate" or "level" that applies to a given placement group. It may also be specified at the booking and placement levels.

The `TermsAndConditions` element holds a set of human-readable textual terms regarding cancellations, claims disclaimers and any other legalistic information that applies to the placement group as a whole. `TermsAndConditions` may be repeated for information in alternative languages, but **MUST NOT** be repeated for any other reason.

The `Guarantees` element can be used to record that certain aspects of the placement group as a whole have been "Guaranteed", and to convey the nature of that guarantee as a set of codes or text strings as appropriate.

The total price for the placement group can be recorded in the `PlacementGroupPrice` element. Also, the price to be paid by one or more payers can be declared using the `PayerInformation` element.

If the placement group is provided for free, the `CostExempt` element should be used to record the type and reason. The element **MUST** be omitted when the placement group is not cost exempt.

In case an additional breakdown of prices is needed, the `PlacementGroupSubPrice` can be used to provide price information for any group of Placements inside the `PlacementGroup`. The

`PlacementGroupSubPrice` has the same content model as the overall `PlacementGroupPrice`, extended with a repeatable `PlacementReference` element that should record the QID(s) of the Placement(s) in question. Note however that the *`bookingReference`* attribute in `PlacementReference` **MUST NOT** be used in this context since all Placements referenced **MUST** be child elements in the `PlacementGroup` structure.

A short descriptive text for the `PlacementGroup` may be added using the `adsml:DescriptionLine` element. It may be repeated to capture the description in alternative languages, but **MUST NOT** be repeated for any other reason.

The placements in the group are defined as a set of media specific placements such as `Placement.NewspaperMagazine`, which supports mainly (but is not limited to) newspapers and magazines.

The `adsml:DocumentRendering` element allows the sender of a business document to convey a digital rendering of the `Placement` either by containership (e.g. a PDF is embedded in the message) or by reference (a URL or equivalent is provided so that the recipient can automatically retrieve the rendering). The element may be repeated for documents in alternative languages, but **MUST NOT** be repeated for any other reason. The element supports the

adsml:i18nAttributes group for language metadata. Data in these attributes refer to the language in the rendered document.

Additional requirement relating to the transaction as a whole can be recorded in the adsml:SpecialRequirements structure.

Application-specific extensions can be included using the optional adsml:Properties element.

Human communication between e.g. buyer's and seller's sales rep's regarding the placement may be recorded using the adsml:Notes element. Although convenient, it is strongly **RECOMMENDED** not to use this element for important bookings data that can be recorded elsewhere in a more formal way.

Attributes

No attributes.

3.3.80 Element: PlacementGroupPrice

The PlacementGroupPrice element captures the total price for a complete placement group.

See PlacementGroup and adsml:CurrencyPriceDeclarationType for more information.

Attributes

No attributes.

3.3.81 Element: PlacementGroupSubPrice

The PlacementGroupPrice element records a sub total price for a subset of Placements in a PlacementGroup.

See PlacementGroup and adsml:CurrencyPriceDeclarationType for more information.

Attributes

No attributes.

3.3.82 Element: PlacementIdentifier

The PlacementIdentifier is a unique identifier for a placement which is assigned by whoever first describes the placement in an AdsMLBookings message.

See Placement and adsml:QIDtype for more information.

Attributes

No attributes.

3.3.83 Element: PlacementInBook

The PlacementInBook element records all information required to specify a page or a number of pages in the "book of pages" for newspaper/magazine media. The Placement can be specified using a choice of one or more

`PlacementCode` elements or an explicit `Page` number. In case of a classified ad, also a set of `ClassifiedPlacementCode` elements may be given.

In case of more than one `PlacementCode`, logical 'AND' **MUST** be applied, i.e. all requirements expressed by the placement codes are requested to be fulfilled together.

The scope of the placement instructions is related to the Publication and possibly Subpublication specified in the `Publication` element that is part of the `Placement.NewspaperMagazine` structure higher up in the booking structure. Within this `Publication` structure, an optional `SubPublicationCode` may be used to identify a meaningful subpart of the overall publication. In addition, within the `PlacementInBook` element described here, an optional `SectionCode` can be used to further specify a subsection of the Publication (or of the sub publication if used). All other placement instructions **MUST** be interpreted within the context specified by these elements. For example, if we have a `Publication "Stockholmstidningen"` with a `SectionCode` of `"BUSINESS"` and then a `PlacementCode` of `"FirstPage"`, this must be interpreted as "`the first page of the business section of Stockholmstidningen`".

Attributes

No attributes.

3.3.84 Element: PlacementLink

See `LinkedPlacement` for more information.

Attributes

No attributes.

3.3.85 Element: PlacementPrice

The `PlacementPrice` element captures the total price for a complete placement.

See `Placement` and `adsml:CurrencyPriceDeclarationType` for more information.

Attributes

No attributes.

3.3.86 Element: PlacementReference

The `PlacementReference` is used in several contexts to reference a placement identifier. If case the placement is from another booking, the *`bookingReference`* attribute should be used to record the booking identifier of that booking.

Attributes

bookingReference (optional)

A reference to a `BookingIdentifier` in a booking

3.3.87 Element: PlacementTarget

The role of the `PlacementTarget` element is to describe the target event, the "thing" booked and eventually paid for in a Placement. This is the primary goal of the order, the result for which the buyer is willing to pay a fee to the seller.

The mandatory `EventType` should be used to record the type of event such as "impressions" or "registrations". Then, the also mandatory `EventCount` holds the target count of the event specified.

In addition to the absolute target count given in `EventCount`, `MinimumEventCount` and `MaximumEventCount` may also be used to record an acceptable range. The value in `EventCount` is expected to be within the given min/max range.

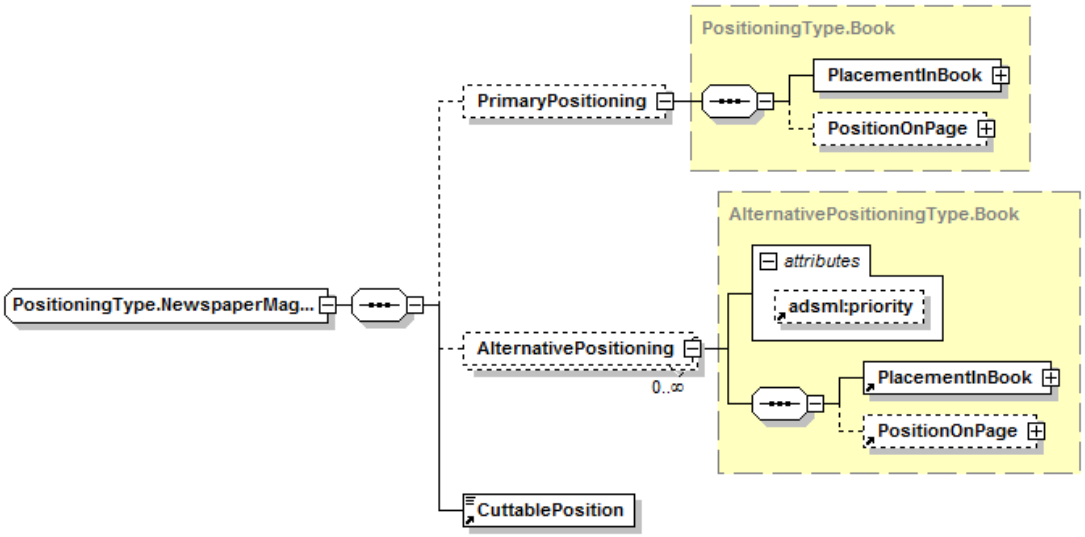
Attributes

No attributes.

3.3.88 Element: Positioning

3.3.88.1 Context: Placement.NewspaperMagazine

The `Positioning` element in the `Placement.NewspaperMagazine` context captures requirements on positioning the advertisement in the publication.



Positioning can be described using `PrimaryPositioning` or optionally one or more `AlternativePositioning` elements, both having the same content model. The alternatives may be ordered using the *`adsml:priority`* attribute. The `PrimaryPositioning` element has default semantics of being the first preferred choice, while alternatives should be evaluated according to the *`adsml:priority`* attribute. Note that it is possible for alternatives to have the same priority, leaving the choice between them to the publisher.

Within these elements, further details are expressed using `PlacementInBook` (i.e. which page in the book of pages) and `PositionOnPage` elements. Instructions within these elements should be considered as having a logical 'AND' relationship.

The PlacementInBook may also include a classification code for a classified ad using the ClassifiedPlacementCode element. It is based on the adsml:CodeType and may use a controlled vocabulary.

If a cuttable position is required, the mandatory element CuttablePosition should must be set to *'true'*, otherwise *'false'*. A "cuttable position" is a position that allows an ad to be cut out from a publication's page without any interference with other cuttable ads on the other side of that page.

Attributes

No attributes.

3.3.88.2 Context: Placement.Generic

The Positioning element in the Placement.Generic context captures requirements on positioning.

Positioning can be described using a PrimaryPositioning or optionally one or more AlternativePositioning elements. The alternatives may be ordered using the *adsml:priority* attribute. The PrimaryPositioning element has default semantics of being the first preferred choice, while alternatives should be evaluated according to the *adsml:priority* attribute. Note that it is possible for alternatives to have the same priority, leaving the choice between them to the publisher.

Instructions within these elements should be considered as having a logical 'AND' relationship.

Attributes

No attributes.

3.3.89 Element: PositionOnPage

The PositionOnPage element is defined as adsml:PositionOnPageType.

A position can be expressed as a set of codes and possibly textual human-oriented descriptions. The interpretation of these requirements **MUST** be logical 'AND'.

If required, the position on the page in exact x-y coordinates can be specified using the adsml:AbsolutePosition element.

Attributes

No attributes.

3.3.90 Element: PreDefinedPeriod

The PreDefinedPeriod element is used within the InsertionPeriod element to capture a predefined insertion period such as the name, number or ID of an issue in which an ad should run, or a name of special package such as 'March classifieds' or 'Back to School issue'. Its values can be validated by a controlled vocabulary.

Attributes

No attributes.

3.3.91 Element: PrePublicationProof

The `PrePublicationProof` element should be used to specify a party that should receive a proof of the printed ad, before publication.

The `SendTo` element, defined as an `adsml:RelaxedPartyType`, defines the intended receiver of the proof.

The mandatory `ProofType` element defines the type of proof. It is defined as `adsml:CodeType` and it is **RECOMMENDED** to use a controlled vocabulary.

The `ProofingParty` is the party that will distribute the proof to the receiver (or has business responsibility for the proofing as a whole). This is usually the Publisher. The `adsml:ProvenanceParty` is another party, often an external third party, that is responsible for (or certifies) the proofing information, e.g. Tearsheet. Usually these will be the same party, but not always. If the `adsml:ProvenanceParty` is not recorded, it should be assumed that the `ProofingParty` is taking that role.

The `NumberOfCopies` element defines the number of copies of the proof that should be sent to the receiver.

See also `Placement`.

Attributes

No attributes.

3.3.92 Element: PrimaryPositioning

See `Positioning`.

Attributes

No attributes.

3.3.93 Element: ProductionDetail.Generic

The `ProductionDetail.Generic` element includes a set of elements that define certain aspects of the ad's publication.

The size of the ad booked can be recorded in the `Size` element. The `Size` element is repeatable to support multiple specifications of the same size using alternative units of measure. The `adsml:Duration` element plays the same role for time based advertisements and captures the booked duration.

In cases where a placement involves more than one set of ad materials, it is necessary to declare how the multiple instances should be managed and produced. Such instructions should be captured using the `MultipleAdContentHandling` element.

The position of the ad within the identified target media is expressed using the `Positioning` element.

If a particular technical ad format is required, it can be identified as a code using the `TechnicalAdFormat` element.

All other requirements regarding the ads production can be described in the `adsml:Specifications` element.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

No attributes.

3.3.94 Element: ProductionDetail.Insert

The `ProductionDetail.Insert` element includes a set of elements that define certain aspects of the insert's production.

The `NumberOfPages` defines the expected number of pages of each insert. Other physical properties of the insert can be provided in the `Thickness` and `Weight` elements. `Width` and `Height` or an `AdSizeCode` can be expressed inside the `Size` element. The `Size` element is repeatable to support multiple specifications of the same size using alternative units of measure.

All other requirements regarding the ads production can be described in the `adsml:Specifications` element.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

No attributes.

3.3.95 Element: ProductionDetail.Interactive

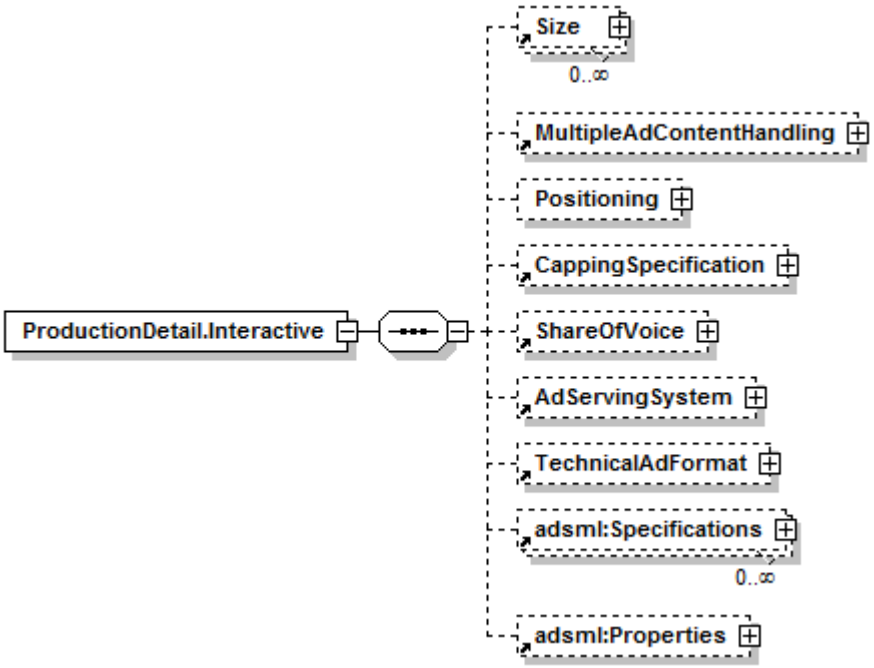
The `ProductionDetail.Interactive` element includes a set of elements that define certain aspects of the ad's publication.

The booked size of the ad is captured in the `Size` element. It is repeatable to support multiple specifications of the same size using alternative units of measure. The `adsml:Duration` element plays the same role for time based advertisements and captures the booked duration.

In cases where a placement involves more than one set of ad materials, it is necessary to declare how the multiple instances should be managed and produced, such as a rotation. Such instructions should be captured using the `MultipleAdContentHandling` element.

The position of the ad within an identified publication is expressed using the `Positioning` element.

Capping is a set of constraints which limit the number of times an advertisement may be published within a defined scope. Capping instructions are recorded in the `CappingSpecification` element where they can be expressed either as purely textual instructions in an `adsml:Description` element, or as a stack of one or more machine-processable `Cap` elements.



It is possible in an interactive order to purchase a "share of voice" of the target publication. This may be expressed as a percentage in the `ShareOfVoice` element.

The `AdServingSystem` element is used to record a code identifying the system that will be used to manage the run of the ad.

The `TechnicalAdFormat` element can be used to record a specifically named advertising format of which there exists a variety within the interactive domain.

Special requirements regarding the ad's production that cannot be expressed using other structures in the message can be described in the `adsml:SpecialRequirements` element.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

No attributes.

3.3.96 Element: ProductionDetail.NewspaperMagazine

The `ProductionDetail.NewspaperMagazine` element includes a set of elements that define certain aspects of the ad's publication.

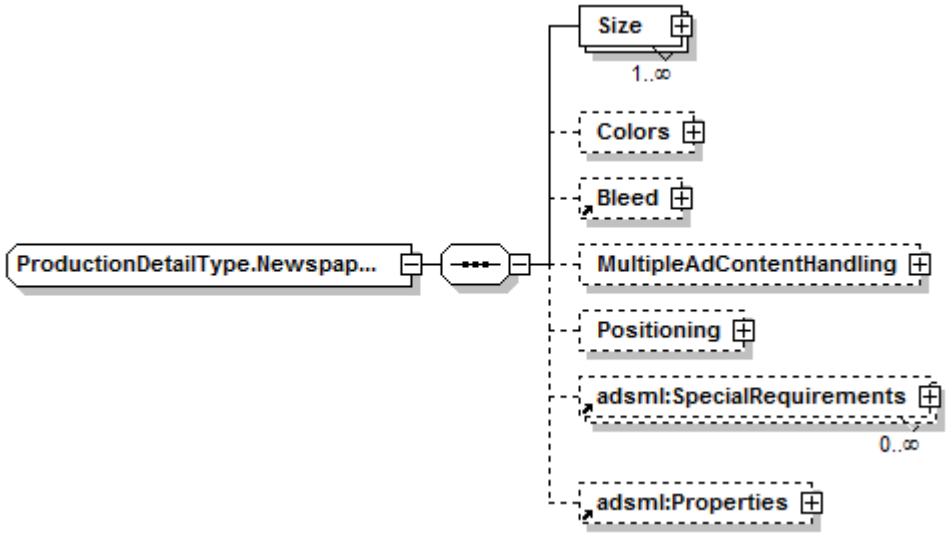
Depending on the type of ad at hand, not all parts of the production detail need to be specified. For instance, color can often be omitted in the case of classified ads.

The booked size and colors of the ad are captured in the `Size` and `Colors` elements respectively. It is repeatable to support multiple specifications of the same size using alternative units of measure.

The type of bleed required, if any, can be expressed using the `Bleed` element. It is based on the `adsml:CodeType` and may use a controlled vocabulary. Note also that `Gutter` specifications are part of the `Size` structure.

In cases where a placement involves more than one set of ad materials, it is necessary to declare how the multiple instances should be managed and produced. An example is the so called 'A/B split' where every second copy of the publication should have a specific artwork printed. Such instructions should be captured using the `MultipleAdContentHandling` element.

The position of the ad within an identified publication is expressed using the `Positioning` element.



Special requirements regarding the ad's production that cannot be expressed using other structures in the message can be described in the `adsml:SpecialRequirements` element.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

No attributes.

3.3.97 Element: ProofingParty

The `ProofingParty` element is defined as an `adsml:RelaxedPartyType`. See `ProofOfPublication` or `PrePublicationProof` for further information.

Attributes

No attributes.

3.3.98 Element: ProofOfPublication

The `ProofOfPublication` element should be used to specify a party that should receive a proof of publication, i.e. a proof that the ad has run (and possibly what it looked like and the context in which it appeared) and information about when, where and to whom it was distributed. Examples from the print world are tearsheets and affidavits. Multiple proof recipients can be specified by including more than one instance of `ProofOfPublication`.

The `SendTo` element, defined as an `adsml:RelaxedPartyType`, defines the intended receiver of the proof.

The mandatory `ProofType` element defines the type of proof. It is defined as `adsml:CodeType` and it is **RECOMMENDED** to use a controlled vocabulary.

The `ProofingParty` is the party that will distribute the proof to the receiver (or has business responsibility for the proofing as a whole). This is usually the Publisher. The `adsml:ProvenanceParty` is another party, often an external third party, that is responsible for (or certifies) the proofing information, e.g. the tearsheet, affidavit or distribution information. Usually these will be the same party, but not always. If the `adsml:ProvenanceParty` is not recorded, it should be assumed that the `ProofingParty` is taking that role.

The `NumberOfCopies` element defines the number of copies of the proof that should be sent to the receiver.

Attributes

No attributes.

3.3.99 Element: ProofType

The `ProofType` element is defined as an `adsml:CodeType`. See `ProofOfPublication` or `PrePublicationProof` for further information.

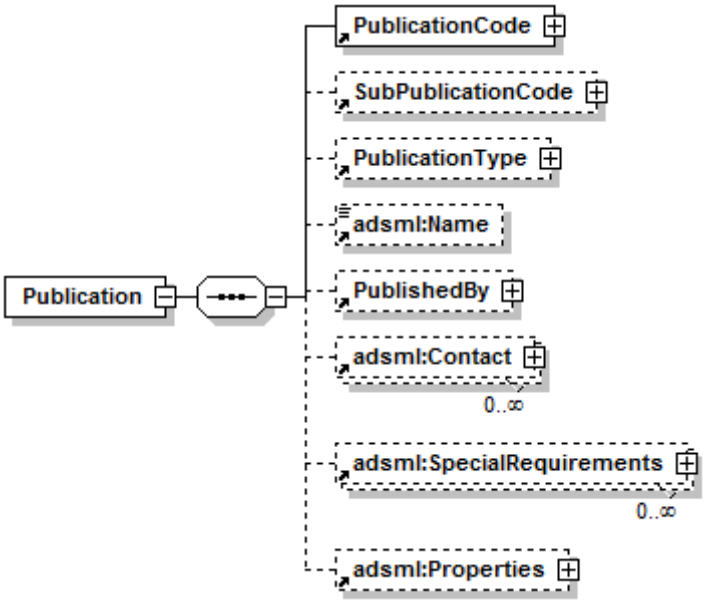
Attributes

No attributes.

3.3.100 Element: Publication

The `Publication` element is used to specify the publication in which an ad should run. It includes a single mandatory `PublicationCode` element that holds a code identifying the publication.

The `PublicationCode` can use a controlled vocabulary for validation.



The `SubPublicationCode` element can be used to define a part of a publication by a machine readable code.

The type of publication can be recorded using the `PublicationType` element.

It is **RECOMMENDED** to use the `adsml:Name` element to provide a human readable name of the publication.

The `PublishedBy` element is defined on the `adsml:RelaxedPartyType` type and can be used to capture the publishing party

The `adsml:SpecialRequirements` can handle any other requirements on how to define a publication.

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

No attributes.

3.3.101 Element: PublicationCode

See `Publication`.

Attributes

No attributes.

3.3.102 Element: PublicationType

See `Publication`.

Attributes

No attributes.

3.3.103 Element: PublishedBy

The `PublishedBy` element is used to define the publisher of a publication. It is based on the `adsml:PartyType` type.

Attributes

No attributes.

3.3.104 Element: QuotationDate

The `QuotationDate` element allows the recording of a business significant date for the quotation. This date may be used in addition to other dates such as *`messageAssembledDateTime`* and *`transmissionDateTime`* which records dates that may be different from the quotation date from a business standpoint.

Attributes

No attributes.

3.3.105 Element: QuotationIdentifier

The `QuotationIdentifier` element declares the primary identifier for the quotation. It is defined as an `adsml:QIDType`. See the Section *Message*

References – Booking and Quotation Identifiers for detailed information about the use of identifiers.

Attributes

No attributes.

3.3.106 Element: RecurrencePattern

See `SchedulingType`.

Attributes

No attributes.

3.3.107 Element: Receipt

The `Receipt` element is used to capture the details of where to send receipts for a payment. A receipt may be issued where, for example, the payer has paid by credit card in advance, but requires the details of the transaction to be acknowledged by the seller.

The Boolean attribute *`sendToPayer`* indicates if the receipt should only be sent to the payer (when 'true'). Additional copies of the receipt should be sent to the parties identified by the optional `SendCopiesTo` element.

See `Payment` for more information.

Attributes

sendToPayer (required)

Defines that a receipt should be sent or not to the party identified as payer. See also `Payer`.

3.3.108 Element: Reference

The `Reference` element is used for general references to business documents or related information, for example, a contract or voucher number. It is based on the `LabeledIDType`.

The semantics of the element are decided by the context.

Attributes

No attributes.

3.3.109 Element: ReservationReference

The `ReservationReference` element provides a pointer to a reservation, when used inside an `AdOrder` element.

See `AdOrder` for more information.

Attributes

No attributes.

3.3.110 Element: RequiredOverage

The `RequiredOverage` element is used in `MaterialsExpectations.Insert` to record the overage required to be available for distribution of Inserts. The overage can be given as a percentage or as an absolute value.

Attributes

No attributes.

3.3.111 Element: ScheduleEntryIdentifier

See `SchedulingType`.

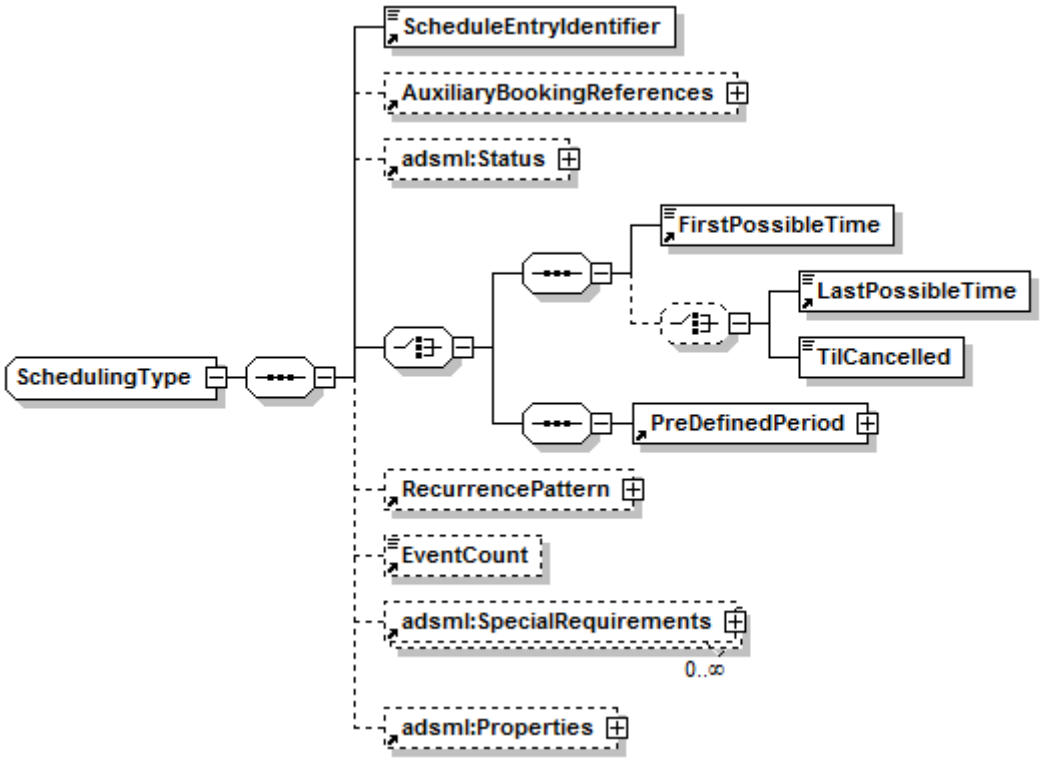
Attributes

No attributes.

3.3.112 Type: SchedulingType

The `SchedulingType` captures requirements on dates and date intervals, i.e. when an ad should occur in a publication.

Every element defined as `SchedulingType` must be able to be identified using the mandatory `SchedulingEntryIdentifier` element. In addition, an `AuxiliaryBookingReferences` element is available. These identifiers are typically used at later stages in the advertising lifecycle, for example, to support the process of reconciling items on invoices with the specific orders that initiated them.



The current status of the ad occurrence within the sender's systems can be provided using the optional `adsml:Status` element. This element **SHOULD** only be included in response messages.

The occurrence period can be defined using one of three alternative models:

- ***Absolute time interval*** - A time interval is defined using the `FirstPossibleTime` and `LastPossibleTime` elements. The required number of occurrences during that period may be defined using the `EventCount` element. For the common case when an ad should occur once on a specific day, `FirstPossibleTime` and `LastPossibleTime` must have identical values and the `EventCount` (if used) must be '1'.
- ***Predefined periods*** – Codified values for predefined periods can be given using the `PreDefinedPeriod` element. It is based on the `adsml:CodeType`. A typical value could be "November issue" or "Back to School issue" of a magazine.
- ***Recurring occurrences*** – Recurring occurrences during a period can be specified using the `RecurrencePattern` element. The start and end points of an overall period during which the events are requested to happen periodically are provided by the same means as described above. The `RecurrencePeriod` starts at a specific time (`FirstPossibleTime`) and either ends at a specific time (`LastPossibleTime`), or after a given number of events (`EventCount`), or is open ended (`TilCancelled`). The `RecurrencePattern` is specified using a code or a text description according to the `RequirementSpecType`, for example: "daily", "every Monday", "Mondays and Wednesdays", etc.

The `adsml:SpecialRequirements` can handle any other requirements on when to publish the ad (or not to).

Application-specific extensions can be included using the optional `adsml:Properties` element.

Attributes

3.3.113 Element: Scope

See `Cap` for information.

Attributes

No attributes.

3.3.114 Element: SectionCode

The `SectionCode` element identifies a section in a publication.

Attributes

No attributes.

3.3.115 Element: SendTo

The `SendTo` element is based on the `adsml:RelaxedPartyType` and defines a receiver.

Attributes

No attributes.

3.3.116 Element: SendCopyTo

The `SendCopyTo` element is based on the `adsml:RelaxedPartyType` and defines a receiver of a copy of something.

Attributes

No attributes.

3.3.117 Element: ShareOfVoice

The `ShareOfVoice` element can be used to specify an ad's share of a particular advertising channel. It is primarily used within the interactive domain.

See `ProductionDetail.Interactive` for further information.

Attributes

No attributes.

3.3.118 Element: SharedPriceComponent

See `ShareOfPrice`.

Attributes

No attributes.

3.3.119 Element: ShareOfTotalPrice

The `ShareOfTotalPrice` element is used for specification of how the costs for an ad should be split between payers. Every instance of the element specifies the share of the total price a particular payer is associated with.

The total price referenced by `ShareOfPrice` is always the instance of `TotalPrice` which is at the same level of the booking message as the `ShareOfPrice` element in question. For example, if `ShareOfPrice` is inside a placement, then the relevant `TotalPrice` element will be the one in the `TotalPlacementPrice` structure for that placement. Conversely, if `ShareOfPrice` is in the `PayerInformation` structure at the root of the `AdMessageRequestModule` (and therefore outside of any particular placement or placement group), then the relevant `TotalPrice` element will be the one in the `TotalBookingPrice` structure which is also at that level.

The `ShareOfTotalPrice` includes a set of `SharedPriceComponents` where each component is specified using a `Name` and a choice between a `Percentage` or `adsml:Amount`. In case of a fixed price amount, the `adsml:CurrencyCode` may also be specified.

See also the `PayerInformation` element.

Attributes

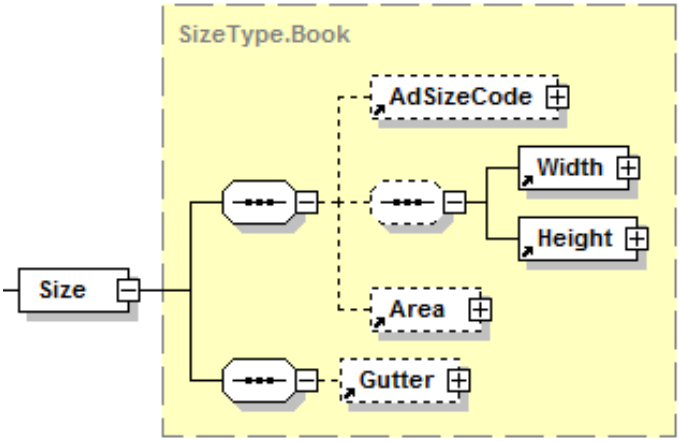
No attributes.

3.3.120 Element: Size

A size can be specified using either explicit measures with either `Height` and `Width` elements or `Area`, or using a specific code captured in the `AdSizeCode` element. The code can be controlled using a controlled vocabulary.

Depending on the context in which the `Size` element is used, more or less restricted versions are defined. The image below shows the full content model of the element as used in the NewspaperMagazine context as expressed in `SizeType.Book`.

The `Area` element is available in all usage contexts, but note that it is an optional element and can be omitted in many contexts.



For print media, the optional `Gutter` element should be used to record size related booking data about gutters. A gutter is defined as the space between two pages and can be booked as an ad position. A gutter affects the actual size of the ad when the size is given as column by width. The gutter may be expressed as an absolute value, but should always have the required included attribute set to true or false depending on if the gutter size is included in the ad size or not.

See also `Bleed`.

Attributes

No attributes.

3.3.121 Element: SubPublicationCode

See `Publication`.

Attributes

No attributes.

3.3.122 Element: Target

The `Targeting` element can be used by the buyer to restrict the distribution of the advertisement to a subset of the audience that the publication is capable of reaching.

See `DistributionTarget` for further information.

3.3.123 Element: Targeting

The `Targeting` element can be used by the buyer to restrict the distribution of the advertisement to a subset of the audience that the publication is capable of reaching.

See `DistributionTarget` for further information.

Attributes

No attributes.

3.3.124 Element: TechnicalAdFormat

The `TechnicalAdFormat` element is defined as an `adsml:CodeType` and should be used to record an identifying code for a technical advertising format. This is particularly common in the interactive domain.

See `ProductionDetail.Interactive` and `ProductionDetail.Generic` for further information.

3.3.125 Element: TermsAndConditions

The `TermsAndCondition` element is used to record a set of free texts describing different types of terms and conditions intended for human users. It includes child elements `adsml:DisclaimerText`, `Claims` and `Cancellations` for particular types of terms and conditions as suggested by the element names, as well as a generic `General` element for other terms.

An example of a buyer's disclaimer text in a booking could be: '`In the event of non-payment by the advertiser, Ad Buyers is liable only for 10% of the seller's placement price. Ad Buyers reserves the right to recover any payments previously advanced. Publication will not be paid until Ad Buyers receives full payment from advertiser`'.

An example of a seller's disclaimer text could be: '`Upon acceptance of this space reservation request, the New York Herald will make best efforts to place the ad on the requested insertion date. The New York Herald does not guarantee the buyer's requested position in the absence of a position fee. Materials must be received by the material deadline in the required digital format and meet the New York Herald's printing specifications in order for the buyer to receive a make good in the event of a DNR (Did Not Run)`'.

When both buyer and seller provide text of a given type, such as the two disclaimer examples above, they can be concatenated in the same element.

It should however be noted that detailed legal constructions should be agreed as part of the TPA set up phase.

The `TermsAndCondition` structure also includes the option to reference or convey the full Terms and Conditions that apply through the `adsml:TermsAndConditionsDetails`, for example as an embedded PDF or as a URL pointing to the web page where they can be found

Attributes

adsml:i18nAttributes (optional)

The *`i18nAttributes`* group supports internationalization by providing attributes to record language, directionality and source.

3.3.126 Element: Thickness

The `Thickness` records a thickness value as a decimal together with a unit of measure. It is defined as an `adsml:DecimalMeasurementType`.

Attributes

No attributes.

3.3.127 Element: TotalBookingPrice

The `TotalBookingPrice` element captures the total price for the complete booking.

See `AdMessageRequestModule` and `adsml:CurrencyPriceDeclarationType` for more information.

Attributes

No attributes.

3.3.128 Element: TotalDistributionCount

The `TotalDistributionCount` element is used to specify a distribution target within a `Distribution` structure.

See `Distribution` for more information.

Attributes

No attributes.

3.3.129 Element: Weight

The `Weight` records a weight value as a decimal together with a unit of measure. It is defined as an `adsml:DecimalMeasurementType`.

Attributes

No attributes.

3.3.130 Element: Width

The `Width` records a width value as a decimal together with a unit of measure. It is defined as an `adsml:DecimalMeasurementType`.

Attributes

No attributes.

Appendix A: Acknowledgment for contributions to this document

This document is a product by the AdsML technical working group. In addition to the main authors listed above, the following members of the technical working group have provided major feedback contributions and know-how:

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

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

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

