

---

For consideration at the August 6, 1998 meeting of the NAA Classified Advertising Standards Task Force

# The New York Times Advertising Markup Prototype

ZEDAK CORP.  
400 COLUMBUS AVE  
VALHALLA, NY 10595-1335

PHONE 1. 800.314.1592 • FAX 1.914.773.6504

<http://www.zedak.com/admarkup>



*USAGE NOTE:*

*This document and the accompanying machine readable material (code) is the intellectual property of The New York Times (The Times) and is copyright The New York Times, 1998. The Times and Zedak Corporation are providing this document and code to the NAA as source-material for the development of an XML standard for classified advertising. The NAA is free to use and incorporate part or all of this material for a standard and may distribute and reproduce this document as it sees fit for that purpose. These materials are a work-in-progress.*

*The Times and Zedak Corporation make no representation that the materials are complete, accurate, or useful. The New York Times and Zedak Corporation make no warranty of any kind, expressed or implied with regard to the code or the instructions contained in this document. The New York Times and Zedak shall not be liable in any event for any damages, including incidental or consequential damages, in connection with or arising out of, the furnishings, performance or use of this code and/or instruction.*

*All company names and logos used in this document are registered trademarks of their respective companies and/or owners.*

*Copyright 1998, The New York Times.*

# Table of Contents

- Introduction..... 1
  - The Players in Electronic Ad Sharing..... 1
  - Markup Roles in *The New York Times* Prototype..... 2
  - Ad Transfer Illustrated..... 3
- XML Overview..... 6
  - Why Use XML for Classified Ad Markup?..... 6
  - The Data Type Definition..... 7
  - Well-Formed vs. Valid XML..... 7
- Public Markup: Ad Transfer Information*..... 8
  - Ad Text..... 8
    - Reply Forwarding..... 9
  - Coding..... 9
    - Contact..... 11
    - Keywording..... 11
  - Publication..... 14
    - Pub\_Alias..... 15
    - Pub\_Options..... 15
    - Zone..... 17
    - Class..... 18
  - Unique Identifier..... 19
  - Expiration..... 19
  - Advertisement..... 20
- Private Inbound Markup: Ad Origination*..... 21
  - Contact..... 22
  - Advertiser..... 23
    - Account..... 24

Contact..... 24

Payment..... 25

Comment..... 27

Reference..... 28

Publication..... 28

    Claim..... 28

    Columns..... 29

    Sortkey..... 29

*Private Outbound Markup: Ad Feedback*..... 30

    Status..... 30

    Source..... 31

        Updated..... 32

        Created..... 32

        Base..... 33

    Charge\_Authorization..... 33

    Pub\_Price..... 34

    Rate..... 34

    Message..... 35

    Lines..... 35

    Instance..... 36

        Edition..... 36

        Section..... 37

        Page..... 37

        Column..... 37

        Offset..... 37

Ad Formatting..... 38

    CENTER..... 38

    FONT..... 38

    GLYPH..... 39

    IMAGE..... 39

KEYWORD..... 39

LEFT..... 40

LINE..... 40

MAILBOX..... 40

MARGIN..... 41

REPLY..... 41

RIGHT..... 41

SPACE..... 42

TAB..... 42

Unique Ad ID..... 43

    Generating Universal Ids..... 43

Classification Mapping..... 44

    Tight Mapping..... 45

    Loose Mapping..... 45

Sample Ad..... 46

The Document Type Definition (DTD)..... 51

    Advertisement.dtd..... 51

Glossary..... 68

## Introduction

As an agent of *The New York Times*, Zedak Corp. has developed an advertising-submission system, in place at *The Times* and at numerous ad agencies since 1985. While engaged in the development of an updated system supporting Internet connectivity and a browser interface, we have encountered many of the ad-entry, markup, and transmission issues currently under discussion by the standards task force. We have addressed these issues, and others, in prototype development. We herein offer—for the sake of discussion and in an effort to bring some obscure issues to light—solutions our developers have incorporated in *The Times*' prototype.

We anticipate that members of the standards task force will find this effort useful as they consider ways to resolve markup issues related to publishing in media-independent formats. We encourage members to engage in discussion of the enclosed draft XML DTD and its application in markup code. As the prototype evolves it continues to undergo broad examination. We welcome member feedback!

## The Players in Electronic Ad Sharing

The players involved in classified ad publishing and transferring include:

- ◆ the *advertiser* who wants to place the ad;
- ◆ the *ad taker* who accepts and generates the ad in XML markup, whether that ad comes directly from the advertiser or from an upline user;
- ◆ the *ad publisher* who originally publishes the ad and transfers basic ad content expressed in XML—along with other required public markup information—to downline users;
- ◆ the *upline user*—who may be either the original ad publisher transferring public markup downline, or a publisher to whom an ad has been transferred and who will, in turn, transfer it downline to additional publishers;
- ◆ the *downline user* who publishes the XML markup in a different media; and
- ◆ the *ad aggregator* who takes collections of ads and arranges them together.

## Markup Roles in *The New York Times* Prototype

*The New York Times* advertisement markup prototype includes three markup roles:

- ◆ *Public Markup: Ad Transfer* information: data to be forwarded to downline publishers. This information includes the ad content itself, searchable terms to be used for indexing, placement information, a unique identifier, an optional expiration date, mailbox, and an action to be taken by the publisher (create, kill, update, or preview).
- ◆ *Private Inbound Markup: Ad Origination* information: data required for billing—*not* forwarded downline to other publishers. This information includes account information, comments, and publication-specific claim, sort, width, and zone directions.
- ◆ *Private Outbound Markup: Ad Feedback* information: data returned by the publisher to the ad's originator—*not* forwarded downline to other publishers. This information includes status, authorization, version, price, rate breakdown, and positioning information—as well as warning and confirmation messages.

### Ad Transfer Illustrated

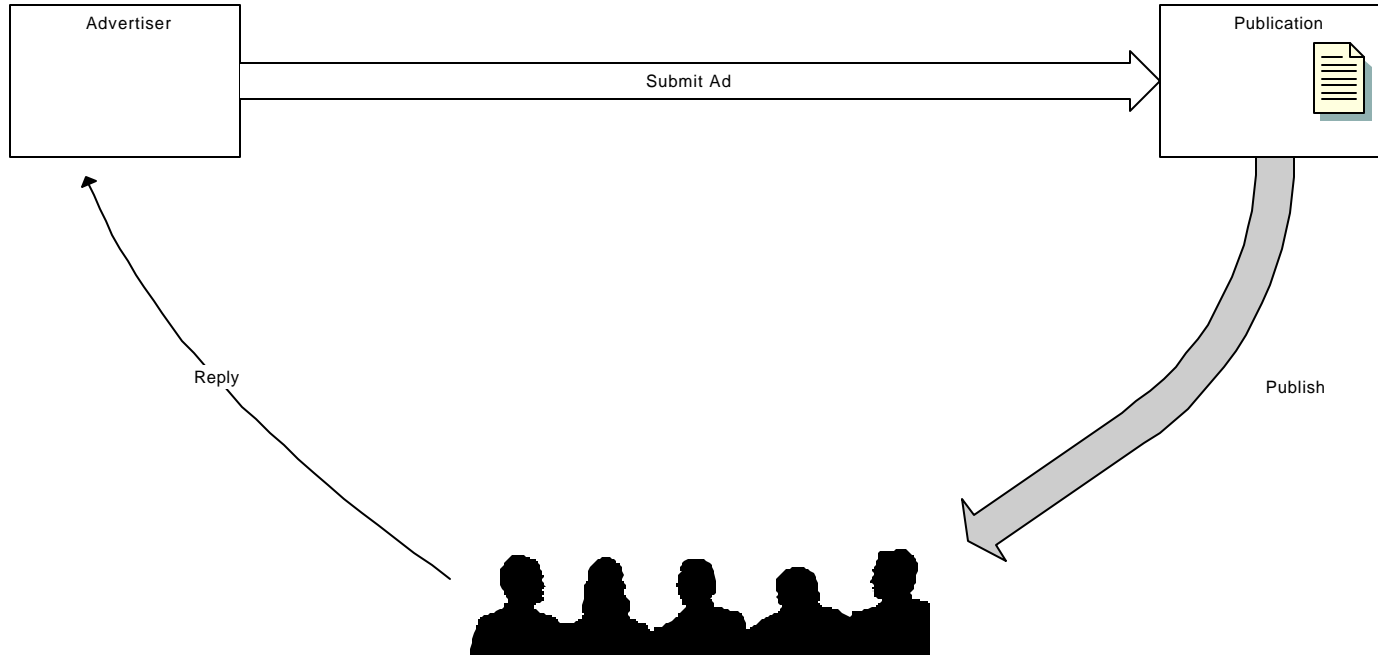


Figure 1: *Classified Advertising*

In Figure 1, the advertiser provides both public and private information to the publisher. The private information includes payment arrangements, for example. The public information includes the ad content, formatting, coding, and classification. This information is passed on to the publication’s audience.

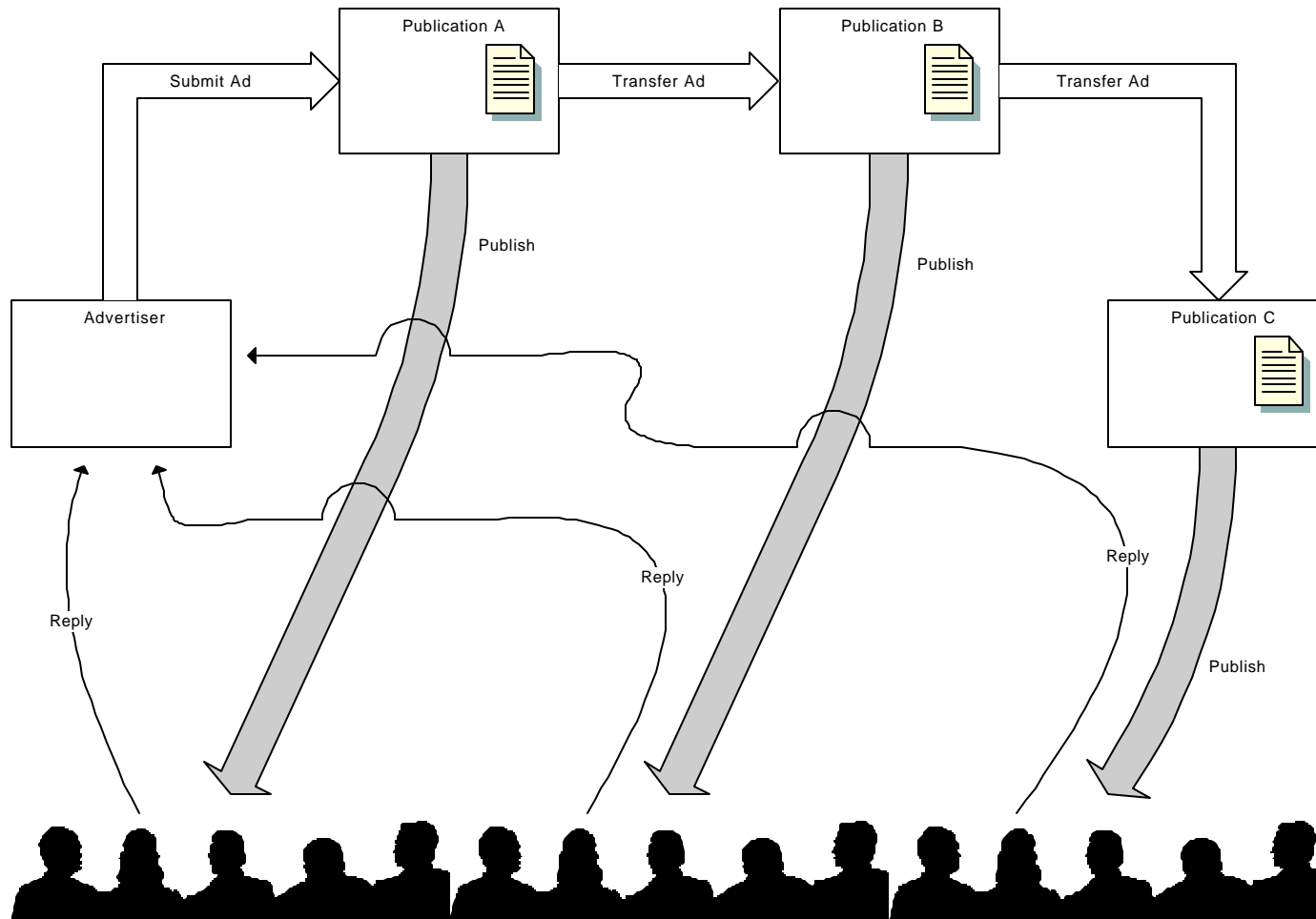


Figure 2: *Electronic Transfer of Classified Ads*

In Figure 2, the downline user (Publication B, Publication C) receives Public Markup (ad transfer information) from an upline user (Publication A, Publication B) for re-publication. Downline user Publication B becomes an upline user when transferring the Public Markup on to additional downline users.

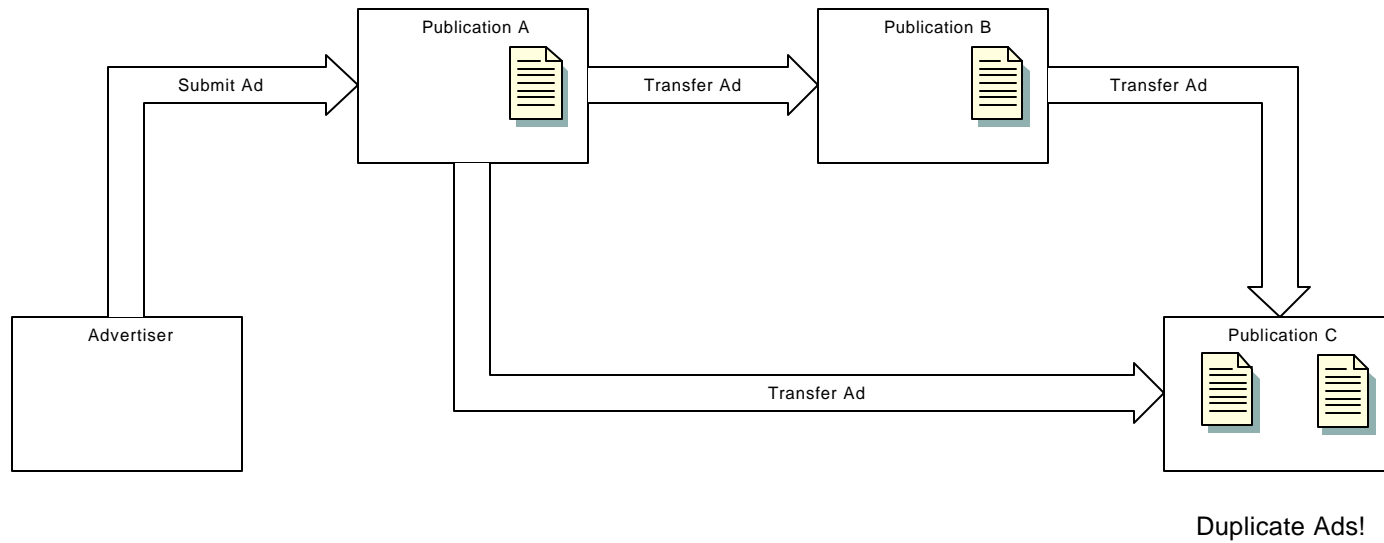


Figure 3: *Stereo Transfer*

In Figure 3, downline user Publication C receives Public Markup (ad transfer information) *for the same ad* from more than one upline user (Publication A and Publication B). This redundancy can be resolved because each ad is assigned its own globally-unique identifier.

## XML Overview

### Why Use XML for Classified Ad Markup?

XML (Extensible Markup Language) is a system for defining markup languages. It provides a common syntax for structuring data according to its content or meaning. And it provides a means to deliver and exchange structured data between applications and electronic systems for media-independent display and manipulation.

Instead of providing display instructions (which can only be handled by the device for which it was designed), each XML element identifies a particular *kind* of information. Any system receiving the XML markup can interpret the elements and determine how to display them appropriately. Using XML markup, classified ad takers define the structure of ad text, searchable terms, placement information, unique ad identifiers, and so forth. Once structured, the ad can be handled by—and tailored for—any device with an XML browser.

“By separating structure and content from presentation, the same XML source document can be written once, then displayed in a variety of ways: on a computer monitor, within a cellular-phone display, translated into voice on a device for the blind, and so forth. It’ll work on any communications devices that might be developed; an XML document can thus outlive the particular authoring and display technologies available when it was written.

So XML will have a life outside of the Internet, serving the publishing industry at large, for example, and especially people who produce documents intended to appear across multiple media.”<sup>1</sup>

Finally, because XML is extensible, it can be made to handle every requirement for classified ad publishing across various media.

---

<sup>1</sup> Reprinted from CNET’s Builder.Com site, “Twenty Questions on XML”.

## The Data Type Definition

A Document Type Definition (DTD) defines syntax rules for a common XML tag set. It defines the tags that can be used in a document, the relationships between the tags and their order, which tags can be nested, which have attributes, and so forth. In essence, the DTD is an agreement between people exchanging documents on how they will use XML to describe a common document architecture.

The DTD ensures uniform structure and desired order, required for accurate data exchange and interpretation. Because XML does not have its own universal DTD—as does HTML—each industry that wants to use XML for exchanging data must define its own DTDs.

This document includes an appendix with a draft DTD. This DTD was developed for the prototype ad submission system being engineered for *The New York Times*. See “The Document Type Definition (DTD)” on page 51.

## Well-Formed vs. Valid XML

XML documents are either “well-formed” or “valid”.

A *well-formed* XML document conforms to the rules of XML syntax, including:

- ◆ XML markup always starts with a left-angle bracket or an ampersand;
- ◆ XML data must always be closed with an end tag as in the tag pair `<text> </text>;`
- ◆ Elements and attributes are case-sensitive;
- ◆ Attributes require quotation marks.

A *valid* XML document conforms to a specific Document Type Definition (DTD).

Systems generating XML markup for classified ads must confirm that the ad markup is valid. A valid document is inherently well-formed.

## Public Markup: Ad Transfer Information

Public Markup includes those essential bits of the ad that must be transferred when publication in additional newspapers or alternative media is required. Public Markup includes:

- ◆ Ad text
- ◆ Searchable terms used for indexing
- ◆ Placement information
- ◆ Unique identifier for the ad
- ◆ Optional expiration date for the ad
- ◆ Action—create, kill, update, preview

### Ad Text

Given an auto-for-sale ad appearing in a newspaper like this:

Saab 900SE  
1997 yellow convertible, 14k  
miles, Auto, PL, PW, AC, power  
leather Seats Showroom  
cond. Assume lease.  
Call 212-333-3333

the ad text would be marked up for transfer inside the `<text>` `</text>` tags as follows:

```
<text>  
  <font size="10"><center>  
    SAAB 900SE  
  </center></font>  
  1997 yellow convertible, 14k miles,  
  Auto, PL, PW, AC, power leather Seats  
  Showroom cond. Assume lease.  
  <center>  
    Call 212-333-3333  
  </center>  
</text>
```

*Ad text—including formatting elements—appears inside text tags.*

*Here, the first and last lines are set “centered”. The first line is set at 10 point type; the remainder of the text at default font size. Where not centered, text is justified—when set, lines will wrap as required.*

Refer to “Ad Formatting” on page 38 for formatting markup.

## Reply Forwarding

If the advertiser requests a box and mail-forwarding service, the ad text includes a `<mailbox>` element. The mailbox element is usually nested in a `<reply>` `</reply>` element.

```
<text>
  <font size="10"><center>
    SAAB 900SE
  </center></font>
  1997 yellow convertible, 14k miles,
  Auto, PL, PW, AC, power leather Seats
  Showroom cond. Assume lease.
  <center>
    <reply>
      Send replies to NY Times Box
      <mailbox>
        T1234
      </mailbox>
    </reply>
  </center>
</text>
```

*The reply element instructs the publisher to substitute text from the forwarding element<sup>2</sup>, if applicable—or to use the text appearing inside the reply tag—in the ad’s text. The mailbox element instructs the publisher to substitute the assigned forwarding mailbox.*

In the case of ads created for more than one publication, when reply instructions must differ for each publication, the reply instruction verbiage is specified in the forwarding element. (See “Forwarding” on page 15.) When a downline user is capable of handling the substitution, the entire text string from the forwarding element will replace text entered inside the `<reply>` `</reply>` tags.

## Coding

Coding is a rich, structured representation of the ad content. Coding also supports ad searching. Finally, where the classification handling is insufficient to identify the ad class, coding assists downline users in classifying the ad.

The values inside the `<coding>` `</coding>` element are standardized for data storage and comparison, not formatted for presentation. For example, the telephone number inside the `<phone>` `</phone>`

---

<sup>2</sup> Forwarding is an element of the `<pub_options>` element of the `<publisher>` element.

element is represented as a string of digits without punctuation. If the same information is also included in the text element, it would be formatted for presentation there.

Inside the coding element, the ad classification is specified with its appropriate tags; e.g., `<automotive>` `</automotive>`. Refer to the Task Force's "Data Elements" publication for elements appropriate to each ad classification.

```
<coding>
  <automotive>
    <auto_side value="sell">sell</auto_side>
    <auto_category value="used">used</auto_category>
    <auto_year>1991</auto_year>
    <auto_make>Saab</auto_make>
    <auto_model>900 Convertible</auto_model>
    <auto_mileage>72000</auto_mileage>
    <auto_price>$13,900</auto_price>
    <auto_exterior>white</auto_exterior>
    <auto_interior>gray leather</auto_interior>
    <auto_body value="convertible">convertible</auto_body>
    <auto_vin>372AB918098910X</auto_vin>
  </automotive>
  <contact>
    <name></name>
    <phone>19085551212</phone>
  </contact>
</coding>
```

*Values for searchable terms are drawn from the ad text and then placed inside the `<coding>` `</coding>` tags. Elements available for automotive ads are nested in the automotive element.*

### Contact

Coding may also include a contact name, box number, or phone number for ad reader response. Nested in the coding element, contact information is supplied inside the `<contact>` `</contact>` tags.

```
<coding>
  <automotive>
    <auto_side value="sell">sell</auto_side>
    <auto_category value="used">used</auto_category>
    ...
  </automotive>
  <contact>
    <name></name>
    <phone>19085551212</phone>
  </contact>
</coding>
```

*Contact information relates to the recipient of ad replies.*

### Keywording

The ad text in the text element can be expanded to include keywords. Keywording is a way to bind the ad text to the coding. Some downline users may wish to ignore these tags and use just their contents, while others may make clever use of the tags.

The keyword element takes the following attributes:

- ◆ *name* (required): must match a coding element used in the same ad
- ◆ *punct* (optional): suffix characters that are added to the formatted value when the formatted length is greater than zero.
- ◆ *format* (optional): provides a pattern for expressing the raw coding value into a form suitable for ad text. For example, “m/yy” is a format that might be used for a date.
- ◆ *scale* (optional): if present, the coding value is divided by scale before formatting. A scale of “1000” is appropriate to express mileage as ‘k miles’.

```
<text>
  <font size="10">
    <center>
      <keyword name="auto_make" punct=" ">SAAB</keyword>
      <keyword name="auto_model" punct=" ">900SE</keyword>
    </center>
  </font>
  <keyword name="auto_year" punct=" ">1997</keyword>
  <keyword name="auto_exterior" punct=" ">yellow</keyword>
  <keyword name="auto_body" punct=",">convertible</keyword>
  <keyword name="auto_mileage" format="9'k miles"
    scale="1000" punct=",">14k miles
</keyword>
Auto, PL, PW, AC, power leather Seats
Showroom cond. Assume lease.
<center>
  Call
  <keyword name="phone" format="T999-999-9999"
    punct=" ">212-333-3333
  </keyword>
</center>
</text>
```

*Via keywording, the word SAAB is bound to the coding element "auto\_make"; the word 900SE is bound to the coding element "auto\_model", and so on.*

*The punct attribute allows for spaces, commas, etc. The format and scale attributes allow for mileage expressed as "k miles".*

The downline user can ignore the <keyword> </keyword>tags themselves, but not the contents. To a downline user ignoring keywording the text would be treated like:

```
<text>
  <font size="10"><center>
    SAAB 900SE
  </center></font>
  1997 yellow convertible, 14k miles,
  Auto, PL, PW, AC, power leather Seats
  Showroom cond. Assume lease.
  <center>
    Call 212-333-3333
  </center>
</text>
```

*With keyword elements filtered out, downlines users handle the ad as formatted text.*

For downline users who cannot recognize keyword tags, they can be filtered out. For those who can recognize and ignore keyword tags, filtering is not required. And downline users who can recognize and process keyword tags may wish to make sophisticated use of them

Keywording allows an editor to propagate coding changes into the ad text automatically. This ensures that the text and coding agree, despite changes, without requiring human labor. For example, suppose an ad's text includes an abbreviated year; e.g., '96. The year may be defined in the coding as 1996 since four digits offer improved searching. When keywords are used to create a tight link between the ad text and the coding, a user who updates the year in coding to 1997 does not need to remember to update it in the ad text—the change will occur automatically.

Keywording also allows a search facility to highlight the ad text terms that match the searcher's criteria even if it doesn't exactly match the search parameter. For example, the ad text could read '97 in an auto ad selling a 1997 car. The '97 would be tagged:

```
<keyword name="auto_year">
```

Since the search facility knows the search is by year, it can highlight this keyword's interior even though '97 doesn't match 1997.

## Publication

Information specific to each publication is supplied inside `<publication>` `</publication>` tags. `Publication` `<option>` `</option>`, `<class>` `</class>`, and `<rundate>` `</rundate>` elements are each nested inside the publication element. Each unique publication requires its own publication element.

The publication element takes the *name* attribute, accepting the names of publications.

```
<publication name="nytimes">
  <pub_alias>
    981011301
  </pub_alias>
  <pub_options>
    <forwarding collect="email">
      Please email replies to
      <mailbox>
        T1234
      </mailbox>
      @nytimes.com
    </forwarding>
    <tearsheet>
    </tearsheet>
    <shading>
    </shading>
  </pub_options>
  <class>
    3720
    <title>Autos/Vans/Sports Utilities</title>
    <classword>Automotive</classword>
    <classword>For Sale</classword>
    <classword>Used</classword>
    <zone>
      M
      <title>Full Run</title>
    </zone>
    <rundate>
      19980719
    </rundate>
    <rundate>
      19980724
    </rundate>
  </class>
</publication>
```

*Name is an attribute to the publication element. Pub\_alias is an ad-identifier assigned locally by the publication (not a universal identifier across systems).*

*The mailbox number is also substituted in the ad text.*

*The presence of tearsheet and shading elements—even when empty as they are here—indicates a request for a tearsheet and for shading behind the ad.*

*Class and zone are publication-dependent. The classification code here reflects that of The New York Times automotive ads. To assist downline users to determine the appropriate class, class keywords, called “classwords”, are supplied.*

*Ads may be assigned more than one class. Zone and Rundate elements are nested inside class to allow variations by class.*

#### Pub\_Alias

The publication `<pub_alias>` `</pub_alias>` element is an ad-identifier assigned locally by the publication. The `pub_alias` is *not* used across systems as a universal identifier.

#### Pub\_Options

Inside the publication element, options offered by particular publications—such as box requests, tear sheets, and shading—may be specified. These elements are nested inside the `<pub_options>` `</pub_options>` element.

### FORWARDING

The `<forwarding>` `</forwarding>` element is used to request a box and forwarding service, and optionally to specify text with reply instructions. Use the `<mailbox>` `</mailbox>` element as a callout for the actual forwarding mailbox assigned by the publication.

Forwarding takes the attributes *collect* and *forward*.

*Collect* accepts the values: mail, voice, email, none. The value of *collect* determines the kind of mailbox the advertiser requires—U.S. mail, voicemail, email, or none.

*Forward* accepts the values: same, hold, mail, or fax. The value of *forward* determines how the advertiser wants the replies forwarded—by the same method they were received, hold for pickup, U.S. mail, or fax. *Forward* choices are limited depending on the value of *collect*. `Forward="fax"` is only available when `collect="email"`, and `forward must = "fax"` when `collect="voice"`. *Forward* defaults to "same".

`forward="none"` may be used when forwarding substitutions are required for one publication but not for another. In the later case, `forward="none"` is used in the forwarding element nested inside that specific publication element.

If no reply instructions appear inside `<reply>` `</reply>` tags in the text element, text for each publication’s unique reply instructions should be included in the forwarding element; then called with the text element’s `<reply>` `</reply>` tags. For example:

*Forwarding Markup in the Publication Element:*

```
<publication name="nytimes">
...
  <pub_options>
    <forwarding collect="email">
      Please email replies to
      <mailbox>
        T1234
      </mailbox>
      @nytimes.com
    </forwarding>
  ...
</pub_options>
...
</publication>
```

*Reply instructions—unique to each specific publication—are inserted inside the forwarding element. The mailbox element is used to specify insertion of the forwarding mailbox.*

*The forwarding collection method is defined here as “email”. Because the forward attribute wasn’t included in the forwarding element, forward defaults to “same”. Therefore, the emailed responses from ad-respondents will be emailed to the contact’s email address—specified in the contact element.*

*Reply Text Called Out in the Text Element*

```
<text>
...
  <center>
    <reply>
      Email replies to the Times
      <mailbox>
        T1234
      </mailbox>
      @nytimes.com
    </reply>
  </center>
</text>
```

*When downline users cannot handle substitutions, the text string defined here in the text element, “Email replies to the Times T1234@nytimes.com”, will be centered below the ad text.*

*When downline users can handle substitutions, the text string “Please email replies to T1234@nytimes.com”—defined by the forwarding element—will be centered below the ad text.*

TEAR SHEETS

Use of the `<tearsheet>` `</tearsheet>` element establishes a tearsheet request.

## SHADING

Use of the `<shading>` `</shading>` element establishes a request for shading behind the ad.

### Zone

Inside the publication element, use of the `<zone>` `</zone>` element specifies the publication's code for the selected edition of the paper.

Nested inside zone is the `<title>` `</title>` element.

```
<zone>
  M
  <title>
    Full Run
  </title>
</zone>
```

*The specified zone code is specified as M—its corresponding title is “Full Run”.*

### TITLE

The `<title>` `</title>` element is used to identify a meaningful name equivalent to the zone code.

## Class

Inside the publication element, use of the `<class>` `</class>` element specifies *The New York Times'* code for this classification. Additionally, the class element provides class keywords, facilitating classification mapping for downline users.

Nested inside the class element are the classword, title, and rundate elements.

```
<publication name="nytimes">
  <pub_options>
    ...
  </pub_options>
  <class>
    3720
    <title>
      Auto/Van/Sports Utilities
    </title>
    <classword>Automotive</classword>
    <classword>For Sale</classword>
    <classword>Used</classword>
    <rundate>
      19980719
    </rundate>
    <rundate>
      19980724
    </rundate>
  </class>
</publication>
```

*Class is the publication's own classification code. Title is the class code's equivalent in English.*

*Classwords help downline users to map the classification to their own classification scheme.*

## CLASSWORD

The `<classword>` `</classword>` element is used to specify a series of class keywords. Downline users may use classwords to aid in mapping the classification to their own classification scheme.

## TITLE

The `<title>` `</title>` element is used to identify the class title.

## RUNDATE

A unique `<rundate>` `</rundate>` element is used to specify each of the ad's insertion dates for the particular publication.

Since ads may be assigned more than one class, insertion dates are nested inside the class element to allow class-specific insertion dates.

### Unique Identifier

The required `<id>` `</id>` element is used to specify a unique identifier for the ad.

The id element takes an attribute called *version*—an optional characteristic that can be used to distinguish between different versions of an ad which has undergone edit changes.

```
<id version="2">  
  NYT.19980701.12345.107  
</id>
```

*Universally-unique identifier required for each ad. Version is specified as "2".*

See “Unique Ad ID” on page 43 for a discussion of unique identifiers.

### Expiration

The `<expiration>` `</expiration>` element is optional. An ad may include an expiration date (or date/time) inside the expiration element.

```
<expiration>  
  19980731  
</expiration>
```

*Optional expiration date upon which the ad should be pulled.*

*COMMENT: Publishers should pull the ad as soon as practical after the date or date/time passes.*

## Advertisement

The `<advertisement>` `</advertisement>` element—specifying what action should be taken for the transmission—wraps around every other element in the ad markup.

The advertisement element takes an attribute called *action*—accepting the values: create, update, kill, preview.

```
<advertisement action="create">  
  <id>  
    ...  
  </id>  
  <expiration>  
    ...  
  </expiration>  
  <coding>  
  <contact>  
    ...  
  </contact>  
    ...  
  </coding>  
  <text>  
    ...  
  </text>  
  <publication>  
    ...  
  </publication>  
</advertisement>
```

*The entire ad markup is nested inside the advertisement element. Action is an attribute of the advertisement element—“create” specifies that the transmission is for a new ad.*

## *Private Inbound Markup: Ad Origination*

Private Inbound Markup supplies information required for billing by the original publication. This information is *not* transferred to downline users.

Private Inbound Markup includes:

- ◆ advertiser account, contact, and payment information
- ◆ optional reference information to appear on the ad invoice
- ◆ optional comments from the ad taker to the ad publisher
- ◆ publication-specific claim, column, and sort key specifications

## Contact

The `<contact>` `</contact>` element specifies name, address, phone, email, and other information either entered (for a transient) or affiliated with an existing account. There may be more than one contact: one for billing, one for mail forwarding, and so forth. A new contact element is required for each contact, and each is assigned its own unique id.

Nested inside the contact element are the `<name>` `</name>`, `<address>` `</address>`, `<phone>` `</phone>`, `<fax>` `</fax>`, `<email>` `</email>`, and `<url>` `</url>` elements—required for transient advertisers and/or for ads with box and mail-forwarding requests.

Elsewhere in the ad, references are made to the appropriate contact by id. In the example below, the id is specified as “contact1”. When referring to this contact throughout the ad, the `<contact_ref>` `</contact_ref>` element is used as a reference. The *link* attribute is set to the value of some contact’s id, thus linking the `contact_ref` to a particular contact.

```
<contact id="contact1">
  <name>
    John Smyth
  </name>
  <address>
    <address_line>c/o Bat Accessories, Inc.</address_line>
    <address_line>Hitchcock Building, 80th Floor</address_line>
    <address_line>1313 Mockingbird Lane</address_line>
    <city>New York</city>,
    <state>NY</state>
    <postal>10000-1234</postal>
    <country>USA</country>
  </address>
  <phone>
    19085551212
  </phone>
  <fax>
    19085551213
  </fax>
  <email>
    jsymth@batacc.com
  </email>
  ...
</contact>
```

*This contact is assigned an id of “contact1”.*

*Address, city, state, postal, country, phone, fax, email, and url elements specify the contact’s location and so forth.*

*Elsewhere in the ad, this contact is referenced by the `<contact_ref link="contact1">` `</contact_ref>` element.*

## Advertiser

The `<advertiser>` `</advertiser>` element is used to specify the company or ad agency paying for the ad, as well as the arrangement for payment. Nested inside the advertiser element are `<account>` `</account>`, `<contact_ref>` `</contact_ref>`, and `<payment>` `</payment>` elements.

```
<advertiser>
  <account type="transient">
    19085551212-1
  </account>
  <contact_ref link="contact1"></contact_ref>
  <payment>
    <charge>
      <charge_card brand="amex"></charge_card>
      <charge_account>3710-111111-99995</charge_account>
      <charge_expiration>19991231</charge_expiration>
      <contact_ref link="contact1"></contact_ref>
      <charge_authorization status="allowed">
        28
      </charge_authorization>
    </charge>
  </payment>
</advertiser>
```

*The account element identifies the account number. The type attribute identifies the type of account.*

*The first contact element specifies that the account contact is contact1.*

*The payment element specifies how the ad will be paid—necessary only for transient advertisers. The second contact element specifies that the billing name and address information is that of contact1.*

### Account

Use the optional `<account>` `</account>` element to specify the advertiser’s account identifier and the account type.

The account element takes the attribute *type*. Type can be:

- ◆ “transient” (default): advertiser without a contract
- ◆ “agency”: advertiser with an agency contract
- ◆ “direct”: advertiser with a non-agency contract
- ◆ “miscellaneous”: transient ad paid for by an agency

```
<advertiser>
  <account type="transient">
    19085551212-1
  </account>
  ...
</advertiser>
```

*This advertiser has no contract—the advertiser is assigned the transient account number 19085551212-1.*

### Contact

Use the `<contact_ref>` `</contact_ref>` element to refer to contact information—defined in the `<contact>` `</contact>` element—for the advertiser.

```
<advertiser>
  ...
  <contact_ref link="contact1"></contact_ref>
  ...
</advertiser>
```

*Name, address, phone, fax, email, and url are all specified for contact1 in the <contact> element, elsewhere in the ad. This <contact\_ref> element links to that previously-defined information..*

## NAME

Use the `<name>` `</name>` element to specify the contact name.

## ADDRESS

Use the `<address>` `</address>` element to specify the contact address.

## PHONE

Use the `<phone>` `</phone>` element to specify the contact phone number.

## FAX

Use the `<fax>` `</fax>` element to specify the contact fax number.

## EMAIL

Use the `<email>` `</email>` element to specify the contact email address.

## URL

Use the `<url>` `</url>` element to specify the contact url.

## Payment

Use the optional `<payment>` `</payment>` element to specify payment information, including payment type and credit card information, if applicable. Default arrangements vary by advertiser and class.

Nested inside the payment element are the `<charge>` `</charge>`, `<cash>` `</cash>`, `<check>` `</check>`, and `<internal>` `</internal>` elements.

Nested inside the charge element are `<charge_card>` `</charge_card>`, `<charge_account>` `</charge_account>`, `<charge_expiration>` `</charge_expiration>`, `<contact_ref>` `</contact_ref>`, and `<charge_authorization>` `</charge_authorization>` elements.

Nested inside the internal element are `<account>` `</account>` and `<reason>` `</reason>` elements.

```
<payment>
  <charge>
    <charge_card brand="amex"></charge_card>
    <charge_account>3710-111111-99995</charge_account>
    <charge_expiration>19991231</charge_expiration>
    <contact_ref link="contact1"></contact_ref>
    <charge_authorization status="allowed">
      28
    </charge_authorization>
  </charge>
</payment>
```

*Charge is the specified payment. Charge-specific elements are nested inside, including a contact reference.*

### CHARGE

#### *charge\_card*

Use the `<charge_card>` `</charge_card>` element to specify the card name, account number, expiration date, etc. for charge payments. `Charge_card` takes the attribute *brand*, assigned the card name; e.g., amex

#### *charge\_account*

Use the `<charge_account>` `</charge_account>` element to specify the billing account number for charge payments. E.g., 3710-111111-99995.

#### *charge\_expiration*

Use the `<expiration>` `</expiration>` element to specify the billing account expiration date for credit-card payments. E.g., 19991231.

#### *contact\_ref*

Use the `<contact_ref>` `</contact_ref>` element to link to a contact defined in the contact element.

## CASH

Use the `<cash>` `</cash>` element to specify payment by cash.

## CHECK

Use the `<check>` `</check>` element to specify payment by check.

## INTERNAL

Use the `<internal>` `</internal>` element to specify account number and so forth for an internal adjustment; e.g., credit for a defective ad.

## Comment

Use the `<comment>` `</comment>` element to include a message from the advertiser to the ad taker, or to otherwise record side notes.

```
<comment>
  Up sold to add Friday repeat.
</comment>
```

*This comment does not appear in the published ad.*

## Reference

Use the `<reference>` `</reference>` element to include optional reference information for the ad invoice.

```
<reference>
  Ad to sell Linda's car.
</reference>
```

*This comment appears on the ad invoice, not in the published ad.*

## Publication

The Publication section, introduced in “*Private Inbound Markup: Ad Origination*” on page 21, also includes the `<pub_options>` `</pub_options>` element for specifying claim, columns, and optional sortkey information.

```
<publication name="nytimes">
  ...
  <pub_options>
    <claim>
      7
    </claim>
    <columns>
      1
    </columns>
    ...
  </pub_options>
  <class>
    ...
    <sortkey>
      SAAB91900
    </sortkey>
    ...
  </class>
</publication>
```

*Claim: although this particular version of the ad text does not run seven times, the advertiser has simultaneously placed a similar ad. The number of run dates totals seven, and billing is based on this claim.*

*The ad is one-column wide.*

*Sort key specifies how the ad is sorted. It is limited to specific values in some classes.*

## Claim

Use the `<claim>` `</claim>` element to specify the total number of inserts for the ad in the specific publication—used when more than one version of the ad text is to run on different dates.

#### Columns

Use the optional `<column>` `</column>` element to specify the width across columns for the ad in the specific newspaper. Ads are one column by default.

#### Sortkey

Use the optional `<sortkey>` `</sortkey>` element to specify the sort key override for the ad in the specific publication.

*COMMENT: Sort key is limited to specific values in some classes, and is publication-specific.*

## Private Outbound Markup: Ad Feedback

The system generates code to return to the ad taker. This information is *not* transferred to other publishers.

Private Outbound Markup includes:

- ◆ ad status,
- ◆ authorization,
- ◆ version,
- ◆ price,
- ◆ rate breakdown,
- ◆ positioning information,
- ◆ warning messages.

### Status

The `<status> </status>` element returns the ad's status.

The status element takes the attribute *value* which may be: accepted, rejected, partial, finished, killed, or killed\*. The default for *value* is "accepted".

```
<status value="accepted"/>
```

## Source

The optional `<source>` `</source>` element returns ad tracking information.

Nested inside the source element are the elements `<updated>` `</updated>`, `<created>` `</created>`, and `<base>` `</base>`.

```
<source>
  <updated>
    <timestamp>
      19980701 12290200
    </timestamp>
    <userid>
      JK1892
    </userid>
  </updated>
  <created>
    <timestamp>
      19980701 12225800
    </timestamp>
    <userid>
      JK1892
    </userid>
  </created>
  <base version="1">
    NYT.19980621.90810.98
  </base>
</source>
```

*The ad-update timestamp is 19980701 12290200.*

*The user updating the ad is identified as JK1892.*

*The ad-creation time stamp is 19980701 12225800.*

*The user (ad taker) generating the ad is identified as JK1892.*

*The ad from which the new ad was picked up is identified as version 1 of ad # NYT.19980621.90810.98.*

### Updated

The `<updated>` `</updated>` element returns date/time and user information related to the current ad update.

Nested inside the updated element are the elements `<timestamp>` `</timestamp>` and `<userid>` `</userid>`.

```
<source>
  <updated>
    <timestamp>
      19980701 12290200
    </timestamp>
    <userid>
      JK1892
    </userid>
  </updated>
  ...
</source>
```

*The ad-update timestamp is 19980701 12290200.*

*The user updating the ad is identified as JK1892.*

### Created

The `<created>` `</created>` element returns date/time and user information related to the original ad creation.

Nested inside the created element are the elements `<timestamp>` `</timestamp>` and `<userid>` `</userid>`.

```
<source>
  ...
  <created>
    <timestamp>
      19980701 12225800
    </timestamp>
    <userid>
      JK1892
    </userid>
  </created>
  ...
</source>
```

*The ad-creation timestamp is 19980701 12225800.*

*The user (ad taker) generating the ad is identified as JK1892.*

## Base

If the ad is a pickup, the `<base>` `</base>` element identifies the original ad's id. Base is not used if the ad was not picked up.

The base element takes the attribute *version*, specifying which version of the picked-up ad was used.

```
<source>
  ...
  <base version="1">
    NYT.19980621.90810.98
  </base>
  ...
</source>
```

*The ad from which the new ad was picked up is identified as version 1 of ad # NYT.19980621.90810.98.*

## Charge\_Authorization

The `<charge_authorization>` `</charge_authorization>` element is nested inside the charge element (which is nested inside the payment element, in turn nested inside the advertiser element).

Authorization returns the credit-card authorization number, and takes the attribute *status*, accepting the value "allowed".

```
<advertiser>
  <payment>
    <charge>
      ...
      <charge_authorization status="allowed">
        4561
      </charge_authorization>
      ...
    </charge>
    ...
  </payment>
  ...
</advertiser>
```

*The charge authorization is approved and assigned the authorization code 4561.*

## Pub\_Price

The `<pub_price>` `</pub_price>` element returns the ad's publication price.

```
<publication name="nytimes">
  ...
  <pub_price>
    $128.00
  </pub_price>
  ...
</publication>
```

*The ad's publication price is \$128.00.*

## Rate

The `<rate>` `</rate>` element returns the payment rate for various publication options including forwarding, tearsheet, shading, and rundate.

The rate element accepts the attributes:

- ◆ *basis*: forwarding service charge, tearsheet service charge, shading premium, Full run, etc.
- ◆ *unit*: ad, recipient, standard.

```
<publication name="nytimes">
  ...
  <pub_options>
    <forwarding>
      ...
      <rate basis="Email forwarding service charge--Full run"
        unit="ad">
          $25.00
        </rate>
      ...
    </forwarding>
    ...
  </pub_options>
  ...
</publication>
```

*The rate basis for the box and mail-forwarding service is \$25.00.*

## Message

The `<message>` `</message>` element returns a message from the publisher in response to numerous elements throughout the ad, at all levels.

The message may be a warning; e.g., *“The ad includes a forwarding charge, but the mailbox isn’t included in the ad text.”* Or the message may be a confirmation, e.g., *“Invoice will include one copy of the published ad, torn from the paper.”*

```
<publication name="nytimes">
  ...
  <pub_options>
    ...
    <tearsheet>
      ...
      <message>
        Invoice will include one copy of the published ad,
        torn from the paper.
      </message>
    ...
  </tearsheet>
  ...
</pub_options>
...
</publication>
```

*The system responds to the tearsheet request with a confirmation message.*

## Lines

The `<lines>` `</lines>` element returns the number of lines in which the ad will be set. In some cases set lines and billed lines differ. When they do, a "billed\_lines" tag shows lines billed.

```
<publication name="nytimes">
  ...
  <class>
    3720
    ...
    <lines>
      4
    </lines>
    ...
  </class>
  ...
</publication>
```

*The ad is set in 4 lines for this classification.*

## Instance

The `<instance>` `</instance>` element returns positioning information. If the ad is placed in more than one edition of the publication, each placement is specified with its own unique instance element.

Nested inside the instance element are `<edition>` `</edition>`, `<section>` `</section>`, `<page>` `</page>`, `<column>` `</column>`, and `<offset>` `</offset>` elements.

```
<publication name="nytimes">
  ...
  <rundate>
    ...
    <instance>
      <edition>BASE</edition>
      <section>12</section>
      <page>22</page>
      <column>9</column>
      <offset>17.85</offset>
    </instance>
    <instance>
      <edition>LI</edition>
      <section>12</section>
      <page>18</page>
      <column>9</column>
      <offset>17.85</offset>
    </instance>
    <instance>
      <edition>NJ</edition>
      <section>12</section>
      <page>18</page>
      <column>9</column>
      <offset>17.85</offset>
    </instance>
  ...
</rundate>
...
</publication>
```

*This ad is placed in three different editions of the publication—Base, Long Island, and New Jersey. In each edition, the ad appears on a different page. The section, column and offset remain the same for each edition.*

## Edition

The `<edition>` `</edition>` element identifies the edition for this instance.

Section

The `<section>` `</section>` element identifies the section for this instance.

Page

The `<page>` `</page>` element identifies the page for this instance.

Column

The `<column>` `</column>` element identifies the column for this instance.

Offset

The `<offset>` `</offset>` element identifies the offset for this instance.

## Ad Formatting

### CENTER

Ad body is centered. Without the `<center>` and `</center>` tags, the ad body would be fully justified. The text between the tags flows to produce the minimum number of lines.

```
<text>
  <left>
    <font size="10">
      SAAB 900SE
    </font>
  </left>
  <center>
    1997 yellow convertible, 14k miles,
    Auto, PL, PW, AC, power leather seats
    Showroom cond. Assume lease.
  </center>
  <right>
    Call 212-333-3333
  </right>
</text>
```

### FONT

Outside of `<font>` `</font>` tags, text is set as agate (the smallest available type size). In this example, the first line of the ad is set in 10 point type.

```
<text>
  <left>
    <font size="10">
      SAAB 900SE
    </font>
  </left>
  ...
</text>
```

### GLYPH

Glyph specifies a special character that can't be represented normally. The ad below contains an em dash (a dash with the width of a capital M) between "showroom cond" and "Assume lease".

```
<text>
  ...
  Auto, PL, PW, AC, power leather seats
  Showroom cond<glyph name="dash"/>Assume lease.
  ...
</text>
```

### IMAGE

Image adds a graphic to the ad. The graphic fills the entire ad width and occupies a specified depth in agate lines.

```
<text>
  <image name="saablogo" depth="6"/>
  <left>
    <font size="10">
      SAAB 900SE
    </font>
  </left>
  ...
</text>
```

### KEYWORD

Keyword binds ad text to ad coding. In the example below, the make and model are marked in the ad text.

```
<text>
  <center>
    <keyword name="make" punct=" ">
      SAAB
    </keyword>
    <keyword name="model" punct=" ">
      900SE
    </keyword>
  </center>
  ...
</text>
```

### LEFT

The first ad line is left-justified.

```
<text>
  <left>
    <font size="10">
      SAAB 900SE
    </font>
  </left>
  ...
</text>
```

### LINE

The first line is stretched using equal spacing between the three words. Line can be supplemented with `<space>` or `<tab>` for other effects.

```
<text>
  <font size="14">
    <line>
      Yellow Saab 900SE
    </line>
  </font>
  ...
</text>
```

### MAILBOX

The assigned forwarding mailbox appears inside the `<mailbox>` `</mailbox>` tags, placed where they are to indicate where the mailbox should appear.

```
<text>
  ...
  <center>
    <reply>
      Email replies to NY Times Box
      <mailbox>
        T1234
      </mailbox>
      @nytimes.com
    </reply>
  </center>
</text>
```

MARGIN

The ad body has an indented first line. Subsequent lines are full measure.

```
<text>
...
<margin left="2" right="0" hang="0">
  1997 yellow convertible, 14k miles,
  Auto, PL, PW, AC, power leather seats
  Showroom cond. Assume lease.
</margin>
...
</text>
```

REPLY

The reply instructions appear inside the <reply> </reply> element. This element is placed to mark the location for replacement reply instructions.

```
<text>
...
<center>
  <reply>
    Email replies to NY Times Box
    <mailbox>
      T1234
    </mailbox>
    @nytimes.com
  </reply>
</center>
</text>
```

RIGHT

The last line of the ad is right-justified.

```
<text>
...
  <right>
    Call 212-333-3333
  </right>
</text>
```

SPACE

The first line is stretched using equal spacing between the three phrases (Yellow, Saab 900SE, and Convertible). Without the `<space/>` elements, there would also be extra space between Saab and 900SE.

`<space/>` is only allowed inside a `<line>` `</line>` element.

```
<text>
  <font size="10">
    <line>
      Yellow<space/>Saab 900SE<space/>Convertible
    </line>
  </font>
  ...
</text>
```

TAB

The ad body contains tabular data. Each line is marked and the columns are `<tab/>` separated. With `flow="no"`, overset lines are split into two lines---the left text is set left-justified on the first line and the right text is set right justified on the next line. With `leader="no"`, there is no leading between the left and right text. The `flow` attribute has no effect unless line is overset. In that case, `flow="yes"` causes the left text to be flowed over two lines to create a more tabular appearance.

```
<text>
  ...
  <line>
    TRUSTS AND ESTATES<tab flow="no" leader="no"/>$40K
  </line>
  <line>
    MUTUAL FUNDS<tab flow="no" leader="no"/>$38K
  </line>
  <line>
    INVESTMENT BANKING DIVISION<tab flow="no" leader="no"/>$36K
  </line>
  <line>
    ENTRY LEVEL OPPORTUNITY<tab flow="no" leader="no"/>$OPEN
  </line>
</text>
```

## Unique Ad ID

While publishers may assign ad ids locally, each ad's Public Markup must include a globally-unique identifier. As ads are transferred from one media to another, publishers must be able to reference each ad by an unambiguous id for maintenance purposes.

They must, for example, be able to track down and kill an ad already transited.

Furthermore, downline users receiving an ad from more than one source must have a means to recognize duplicates in order to avoid publishing more than one copy. Stereo transfer occurs when the same ad is forwarded to one downline user from multiple upline sources. The download user uses the id to filter out duplicate ads. For an illustration of stereo transfer and the potential for ad redundancy, see figure 3 in "Ad Transfer Illustrated" on page 1.

## Generating Universal Ids

Possibilities for assigning identifiers unique across the internet include:

- ◆ Several sources can be established, each with a unique prefix. Suppose *The New York Times* takes an ad, issuing the local identifier 1.2.3.4—an id unique to *The Times*. The ad might be transferred from *The Times* to downline users with the prefix "NYT". *NYT.1.2.3.4* becomes the ad's unique identifier.

A different ad taken by the *Boston Globe* may also be assigned the local identifier 1.2.3.4—an id unique to the *Globe*, but duplicating that of the ad taken by *The Times*. The *Globe's* ad might be transferred from the *Globe* to downline users with the prefix "BG". *BG.1.2.3.4* becomes the ad's unique identifier, distinguishing it from the ad taken by *The Times*.

- ◆ A universal service may be used to ensure that no identifier is ever issued more than once. Who runs that service? And how is the number generated?

Microsoft uses a method of generating 128-bit Globally Unique Identifiers (GUIDs), building them from three components:

- ◆ a unique base—the PC's unique TCP/IP address;
- ◆ the current date and time; and
- ◆ a randomized number

## Classification Mapping

Different publications use different classification schemes. A typical classification scheme assigns classification numbers in a hierarchy. For example, real estate classifications are grouped together and subdivided by type of offering (e.g. rental, sale, land) and by neighborhood. Different publications' classification schemes are likely to differ in:

- ◆ *Classification numbers*—1235 Auto or 5642 Auto?
- ◆ *Classification titles*—Land for Sale or Lots for Sale?
- ◆ *Classification hierarchy*—Is type of offering below or above neighborhood?
- ◆ *Rules for determining where an ad should appear*—Is a bicycle sold under a transportation class or under a general merchandise class?

When an ad is transferred from one publication to another, the new publication is likely to use a different classification scheme. Given this, what classification should the new publication use? Two approaches may be taken: Tight Mapping or Loose Mapping.

## Tight Mapping

Devise a universal classification hierarchy. Require conformant publications to create a two-way mapping between their classification scheme and the universal hierarchy. When publishing ads, use the local to universal mapping to represent the universal classification. Note that the publisher may need to select one of many universal classification that map to just one local classification. Downline users use the universal to local mapping to represent the universal classification in local terms. Manual disambiguation may be required on this end as well.

Tight Mapping results in fewer classification errors than loose mapping, but is probably impractical. It requires:

- ◆ Initial development of the universal taxonomy
- ◆ Creation and maintenance of maps by each publication
- ◆ Coordinated ongoing adjustment as, for example, new classifications emerge.

## Loose Mapping

As with tight mapping, devise a universal hierarchy. Keyword each universal classification and publish the keywords. Each publication keywords their classifications as well (instead of creating a two-way map). When publishing an ad, include the class keywords with the ad. Downline users determine their classifications by:

- ◆ Making a “best fit” match on keywords; Best fit is highest number of matching keywords
- ◆ When necessary, refine the class selection by reference to the ad’s coding
- ◆ Otherwise fallback to a general classification

Loose mapping requires less labor to establish, maintain, and use but is less accurate for some ads.

## Sample Ad

*White = Public Markup*

*Light Gray = Inbound Private Markup*

*Dark Gray = Outbound Private Markup*

```
<!DOCTYPE advertisement SYSTEM "advertisement.dtd">
<advertisement action="update">
  <id version="2">
    NYT.19980701.12345.107
  </id>
  <status value="accepted"></status>
  <expiration>
    19980731
  </expiration>
  <reference>
    Ad to sell Linda's car.
  </reference>
  <comment>
    Up sold to add Friday repeat.
  </comment>
  <contact id="contact1">
    <name>
      John Smyth
    </name>
    <address>
      <address_line>c/o Bat Accessories, Inc.</address_line>
      <address_line>Hitchcock Building, 80th Floor</address_line>
      <address_line>1313 Mockingbird Lane</address_line>
      <city>New York</city>,
      <state>NY</state>
      <postal>10000-1234</postal>
      <country>USA</country>
    </address>
    <phone>
      19085551212
    </phone>
    <fax>
      19085551213
    </fax>
    <email>
      jsymth@batacc.com
    </email>
```

```
<url>
  http://www.batacc.com/~smyth
</url>
</contact>
<source>
  <updated>
    <timestamp>
      19980701 12290200
    </timestamp>
    <userid>
      JK1892
    </userid>
  </updated>
  <created>
    <timestamp>
      19980701 12225800
    </timestamp>
    <userid>
      JK1892
    </userid>
  </created>
  <base version="1">
    NYT.19980621.90810.98
  </base>
</source>
<advertiser>
  <account type="transient">
    19085551212-1
  </account>
  <contact_ref link="contact1"></contact_ref>
  <payment>
    <charge>
      <charge_card brand="amex"></charge_card>
      <charge_account>3710-111111-99995</charge_account>
      <charge_expiration>19991231</charge_expiration>
      <contact_ref link="contact1"></contact_ref>
      <charge_authorization status="allowed">4561</charge_authorization>
    </charge>
  </payment>
</advertiser>
<coding>
  <automotive>
    <auto_side value="sell">sell</auto_side>
    <auto_category value="used">used</auto_category>
    <auto_year>1991</auto_year>
    <auto_make>Saab</auto_make>
    <auto_model>900 Convertible</auto_model>
```

```
<auto_mileage>72000</auto_mileage>
<auto_price>$13,900</auto_price>
<auto_exterior>white</auto_exterior>
<auto_interior>gray leather</auto_interior>
<auto_body value="convertible">convertible</auto_body>
<auto_vin>372AB918098910X</auto_vin>
</automotive>
<contact>
  <name></name>
  <phone>19085551212</phone>
</contact>
</coding>
<text>
  <font size="10">
    <center>
      <keyword name="auto_make" punct=" ">SAAB </keyword>
      <keyword name="auto_model" punct=" ">900SE </keyword>
    </center>
  </font>
  <keyword name="auto_year" punct=" ">1997 </keyword>
  <keyword name="auto_exterior" punct=" ">yellow </keyword>
  <keyword name="auto_body" punct=", ">convertible, </keyword>
  <keyword name="auto_mileage" format="9'k miles" scale="1000"
    punct=", ">14k miles, </keyword>
  Auto, PL, PW, AC, power leather Seats
  Showroom cond. Assume lease.
  <center>
    Call
    <keyword name="phone" format="T999-999-9999" punct=" ">
      212-333-3333
    </keyword>
  </center>
</text>
<publication name="nytimes">
  <pub_alias>
    981011301
  </pub_alias>
  <pub_price>
    $128.00
  </pub_price>
  <pub_options>
    <claim>
      7
    </claim>
    <columns>
      1
    </columns>
  </pub_options>
</publication>
```

```
<forwarding collect="email">
  Please email replies to <mailbox>T1234</mailbox>@nytimes.com
  <rate basis="Email forwarding service charge--Full run"
    unit="ad">$25.00
  </rate>
</forwarding>
<tearsheet>
  <rate basis="Tear sheet service charge"
unit="recipient">$20.00</rate>
</tearsheet>
<shading>
  <rate basis="Shading premium" unit="standard">20%</rate>
</shading>
</pub_options>
<class>
  3720
  <title>Autos/Vans/Sports Utilities</title>
  <classword>Automotive</classword>
  <classword>For Sale</classword>
  <classword>Used</classword>
  <lines>
    4
  </lines>
  <sortkey>
    SAAB91900
  </sortkey>
  <zone>
    M
    <title>Full Run</title>
  </zone>
  <rundate>
    19980719
    <rate basis="Automotive, Open, Sunday NY Region"
      unit="line">$23.10
    </rate>
    <instance>
      <edition>BASE</edition>
      <section>12</section>
      <page>22</page>
      <column>9</column>
      <offset>17.85</offset>
    </instance>
    <instance>
      <edition>LI</edition>
      <section>12</section>
      <page>18</page>
      <column>9</column>
```

```
        <offset>17.85</offset>
    </instance>
    <instance>
        <edition>NJ</edition>
        <section>12</section>
        <page>18</page>
        <column>9</column>
        <offset>17.85</offset>
    </instance>
    <instance>
        <edition>NY/LI</edition>
        <section>12</section>
        <page>18</page>
        <column>9</column>
        <offset>17.85</offset>
    </instance>
    <instance>
        <edition>WC</edition>
        <section>12</section>
        <page>18</page>
        <column>9</column>
        <offset>17.85</offset>
    </instance>
```

```
</rundate>
<rundate>
    19980724
```

```
<rate basis="Automotive, Open, Weekday NY Region--
    Sunday ad repeated on Friday (within 7 days)"
    unit="line">$8.90
</rate>
<rate basis="Automotive, Open, Weekday NY Region"
    unit="line" type="comparison">$15.20
</rate>
<instance>
    <edition>METRO</edition>
    <section>6</section>
    <page>14</page>
    <column>6</column>
    <offset>5.15</offset>
</instance>
```

```
</rundate>
</class>
</publication>
</advertisement>
```

# The Document Type Definition (DTD)

## Advertisement.dtd

```
<!-- ===== -->
<!-- ADVERTISEMENT - entire ad -->
<!-- -->
<!-- action means: -->
<!-- -->
<!--     create  - ad is new; id should not already exist -->
<!--     update  - replace ad that matches on id -->
<!--     kill    - withdraw ad immediately -->
<!--     inquiry - report ad status -->
<!--     preview - validate, but don't publish ad -->
<!-- -->
<!-- ===== -->
<!ELEMENT advertisement
(id,status?,expiration?,reference?,comment?,contact*,source?,advertiser,coding,text
,publication+)>
<!ATTLIST advertisement
  action (create|update|kill|inquiry|preview) "create"
>

<!-- ***** -->
<!--             ADMINISTRATIVE INFORMATION -->
<!-- ***** -->

<!-- ===== -->
<!-- ID - globally unique identifier for ad -->
<!-- -->
<!-- version can be used to distinguish different versions of -->
<!-- the same ad. -->
<!-- ===== -->
<!ELEMENT id (#PCDATA)>
<!ATTLIST id
  version CDATA ""
>

<!-- ===== -->
<!-- STATUS - system state of ad, as described below -->
```

```

<!-- -->
<!-- state means: -->
<!-- -->
<!-- accepted - ad is accepted for publication -->
<!-- rejected - ad not accepted; see <message> tags -->
<!-- partial - ad is incomplete so it can't be processed -->
<!-- finished - ad has run its entire schedule -->
<!-- killed - ad was killed before ever running -->
<!-- killed_x - ad was killed after at least one publication -->
<!-- valid - preview ad passes validation -->
<!-- unknown - inquiry ad does not exist -->
<!-- -->
<!-- ===== -->
<!ELEMENT status EMPTY>
<!ATTLIST status
  value (accepted|rejected|partial|finished|killed|killed_x|valid|unknown)
"partial"
>

<!-- ===== -->
<!-- EXPIRATION - date or date/time ad should be withdrawn -->
<!-- ===== -->
<!ELEMENT expiration (#PCDATA)>

<!-- ===== -->
<!-- REFERENCE - text carried to advertiser's invoice -->
<!-- ===== -->
<!ELEMENT reference (#PCDATA)>

<!-- ===== -->
<!-- COMMENT - advertiser's notes for publisher -->
<!-- ===== -->
<!ELEMENT comment (#PCDATA)>

<!-- ***** -->
<!-- CONTACTS -->
<!-- ***** -->

<!-- ===== -->
<!-- CONTACT - contact info for parties related to this ad -->
<!-- -->
<!-- id is used to uniquely identify each contact. contract_ref -->
<!-- tags that appear later reference a specific contact by -->
<!-- assigning the contact's id value to their link attribute. -->
<!-- ===== -->
<!ELEMENT contact (name,(address|phone|fax|email|url)*)>

```

```
<!ATTLIST contact
  id ID #IMPLIED
>

<!-- ===== -->
<!-- NAME - contact name -->
<!-- ===== -->
<!ELEMENT name (#PCDATA)>

<!-- ===== -->
<!-- ADDRESS - contact address -->
<!-- ===== -->
<!ELEMENT address (#PCDATA|address_line|city|state|postal|country)*>
<!ELEMENT address_line (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT postal (#PCDATA)>
<!ELEMENT country (#PCDATA)>

<!-- ===== -->
<!-- MISCELLANEOUS CONTACT INFO - phone, fax, email, url -->
<!-- ===== -->
<!ELEMENT phone (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT url (#PCDATA)>

<!-- ===== -->
<!-- CONTACT_REF - contact reference -->
<!-- link gets a value that equals the id of a <contact> -->
<!-- ===== -->
<!ELEMENT contact_ref (#PCDATA)>
<!ATTLIST contact_ref
  link IDREF #IMPLIED
>

<!-- ***** -->
<!-- TRACKING INFORMATION -->
<!-- ***** -->

<!-- ===== -->
<!-- SOURCE - ad origination and update record -->
<!-- ===== -->
<!ELEMENT source (updated,created,base?)>
```

```
<!-- ===== -->
<!-- UPDATED - time and agent of latest ad update -->
<!-- ===== -->
<!ELEMENT updated (timestamp,userid)>

<!-- ===== -->
<!-- CREATION - time and agent of original ad store -->
<!-- ===== -->
<!ELEMENT created (timestamp,userid)>

<!-- ===== -->
<!-- BASE - ad pickup source -->
<!-- ===== -->
<!-- version can be used to denote the specific version that -->
<!-- was picked up. -->
<!-- ===== -->
<!ELEMENT base (#PCDATA)>
<!ATTLIST base
  version CDATA ""
>

<!-- ===== -->
<!-- MISCELLANEOUS SOURCE INFO - timestamp, userid -->
<!-- ===== -->
<!ELEMENT timestamp (#PCDATA)>
<!ELEMENT userid (#PCDATA)>

<!-- ***** -->
<!-- ADVERTISER -->
<!-- ***** -->

<!-- ===== -->
<!-- ADVERTISER - identity and payment information -->
<!-- ===== -->
<!ELEMENT advertiser (account,contact_ref,payment)>

<!-- ===== -->
<!-- ACCOUNT - private ad originator account reference -->
<!-- ===== -->
<!-- type means: -->
<!-- ===== -->
<!-- transient - direct advertiser w/o account relationship -->
<!-- agency - contract advertiser via agency -->
<!-- direct - contract advertiser w/o agency -->
<!-- misc - advertiser w/o account via agency -->
<!-- ===== -->
```

```
<!-- ===== -->
<!ELEMENT account (#PCDATA)>
<!ATTLIST account
  type (transient|agency|direct|misc) "transient"
>

<!-- ===== -->
<!-- PAYMENT - payment arrangements -->
<!-- ===== -->
<!ELEMENT payment (cash|charge|check|invoice|no_charge)>

<!-- ===== -->
<!-- CASH -->
<!-- ===== -->
<!ELEMENT cash (#PCDATA)>

<!-- ===== -->
<!-- CHARGE -->
<!-- ===== -->
<!ELEMENT charge
(charge_card,charge_account,charge_expiration,contact_ref,charge_authorization)>
<!ELEMENT charge_card (#PCDATA)>
<!ATTLIST charge_card
  brand (amex|discover|master_card|visa) #REQUIRED
>
<!ELEMENT charge_account_number (#PCDATA)>
<!ELEMENT charge_expiration (#PCDATA)>
<!ELEMENT charge_authorization (#PCDATA)>
<!ATTLIST charge_authorization
  status CDATA #REQUIRED
>

<!-- ===== -->
<!-- CHECK -->
<!-- ===== -->
<!ELEMENT check (check_number,check_account,contact_ref)>
<!ELEMENT check_number (#PCDATA)>
<!ELEMENT check_account (#PCDATA)>

<!-- ===== -->
<!-- INVOICE -->
<!-- ===== -->
<!ELEMENT invoice (#PCDATA)>

<!-- ===== -->
<!-- NO_CHARGE -->
<!-- ===== -->
```

```
<!-- offset account is the internal account that is charged -->
<!-- for the cost of the ad. -->
<!-- -->
<!-- reason means: -->
<!-- -->
<!-- house_ad - publisher is paying for ad -->
<!-- credit_for_error - credit, explained in PCDATA -->
<!-- special - special, explained in PCDATA -->
<!-- other - other, explained in PCDATA -->
<!-- -->
<!-- ===== -->
<!ELEMENT no_charge (#PCDATA)>
<!ATTLIST no_charge
  offset_account CDATA #REQUIRED
  reason (house_ad|credit_for_error|special|other) #REQUIRED
>

<!-- ***** -->
<!-- CODING -->
<!-- ***** -->

<!-- ===== -->
<!-- CODING -->
<!-- -->
<!-- Note that coding includes an actual <contact> instead of -->
<!-- a <contact_ref>. This is to allow the advertiser to -->
<!-- control which contact details are to be published. -->
<!-- ===== -->
<!ELEMENT coding ((automotive|employment|real_estate),contact)>

<!-- ===== -->
<!-- AUTOMOTIVE -->
<!-- ===== -->
<!ELEMENT automotive
(auto_side?,auto_category?,auto_year?,auto_make?,auto_model?,auto_mileage?,auto_pri
ce?,auto_exterior?,auto_interior?,auto_body?,auto_vin?)>
<!ELEMENT auto_side (#PCDATA)>
<!ATTLIST auto_side
  value (sell|buy) "sell"
>
<!ELEMENT auto_category (#PCDATA)>
<!ATTLIST auto_category
  value (unknown|new|used|antique) "unknown"
>
<!ELEMENT auto_year (#PCDATA)>
<!ELEMENT auto_make (#PCDATA)>
```

```
<!ELEMENT auto_model (#PCDATA)>
<!ELEMENT auto_mileage (#PCDATA)>
<!ELEMENT auto_price (#PCDATA)>
<!ELEMENT auto_exterior (#PCDATA)>
<!ELEMENT auto_interior (#PCDATA)>
<!ELEMENT auto_body (#PCDATA)>
<!ATTLIST auto_body
  value (unknown|sedan|coupe|convertible|suv|4x4|pickup|van|minivan) "unknown"
>
<!ELEMENT auto_vin (#PCDATA)>

<!-- ===== -->
<!-- EMPLOYMENT -->
<!-- ===== -->
<!ELEMENT employment
(empl_side?,empl_category?,empl_experience?,empl_salary?,empl_basis?,empl_skills?,e
mpl_benefits?)>
<!ELEMENT empl_side (#PCDATA)>
<!ATTLIST empl_side
  value (offered|wanted) "offered"
>
<!ELEMENT empl_category (#PCDATA)>
<!ELEMENT empl_title (#PCDATA)>
<!ELEMENT empl_experience (#PCDATA)>
<!ELEMENT empl_salary (#PCDATA)>
<!ATTLIST empl_salary
  unit (unknown|annual|hourly|weekly|monthly) "unknown"
>
<!ELEMENT empl_basis (#PCDATA)>
<!ATTLIST empl_basis
  value (unknown|full_time|part_time|contract) "unknown"
>
<!ELEMENT empl_skills (#PCDATA)>
<!ELEMENT empl_benefits (#PCDATA)>

<!-- ===== -->
<!-- REAL ESTATE -->
<!-- ===== -->
<!ELEMENT real_estate
(real_side?,real_category?,real_year_built?,real_class?,real_price?,real_bedrooms?,
real_bathrooms?,real_location?)>
<!ELEMENT real_side (#PCDATA)>
<!ATTLIST real_side
  value (sell|buy|rent|to_rent) "sell"
>
<!ELEMENT real_category (#PCDATA)>
<!ATTLIST real_category
```

```
value (unknown|new_construction|resale|land) "unknown"
>
<!ELEMENT real_year_built (#PCDATA)>
<!ELEMENT real_class (#PCDATA)>
<!ATTLIST real_class
  value
  (unknown|single_family|multiple_family|apartment|condominium|residential_building|c
ommercial_building|residential_property|commercial_property) "unknown"
>
<!ELEMENT real_price (#PCDATA)>
<!ELEMENT real_bedrooms (#PCDATA)>
<!ELEMENT real_bathrooms (#PCDATA)>
<!ELEMENT real_location (address|real_neighborhood)>
<!ELEMENT real_neighborhood (#PCDATA)>

<!-- ***** -->
<!-- AD TEXT -->
<!-- ***** -->

<!ENTITY % inline "#PCDATA|font|glyph|image|keyword|mailbox|margin">
<!ENTITY % spacer "space|tab">
<!ENTITY % flow "center|left|line|right">

<!-- ===== -->
<!-- TEXT - ad Text including formatting -->
<!-- ===== -->
<!ELEMENT text (%inline;|%flow;|reply)*>

<!-- ===== -->
<!-- CENTER - center text within prevailing margins -->
<!-- -->
<!-- center causes a break. -->
<!-- Interior content is flowed to best fit. -->
<!-- ===== -->
<!ELEMENT center (%inline;|reply)*>

<!-- ===== -->
<!-- FONT - select font size -->
<!-- -->
<!-- size is the new font size. agate is a synonym for the -->
<!-- smallest available font size. The sizes allowed vary for -->
<!-- different ads, so some sizes allowed in the DTD may be -->
```

```
<!-- adjusted to the closest size allowed for the ad. -->
<!-- -->
<!-- font causes a break because only one font size per line is -->
<!-- allowed. -->
<!-- ===== -->
<!ELEMENT font (%inline;|%flow;|reply)*>
<!ATTLIST font
  size (agate|5|6|10|12|13|14|18|24|30|31|36|48|60|72) "agate"
>

<!-- ===== -->
<!-- GLYPH - call for a special (non-ASCII) character -->
<!-- -->
<!-- name selects the desired glyph. Values allowed: -->
<!-- -->
<!-- en - space the width of a capital N -->
<!-- em - space the width of a capital M -->
<!-- thin - minimal (non-stretchable) space -->
<!-- figure - space the width of a digit -->
<!-- dash - dash the width of a capital M -->
<!-- open - open double quotation mark -->
<!-- close - close double quotation mark -->
<!-- fractions: 1/8, 3/8, 5/8, 7/8, 1/4, 3/4, 1/3, 2/3, 1/2 -->
<!-- expressed: 1-8, 3-8, 5-8, 7-8, 1-4, 3-4, 1-3, 2-3, 1-2 -->
<!-- -->
<!-- glyph is non-breaking. -->
<!-- ===== -->
<!ELEMENT glyph EMPTY>
<!ATTLIST glyph
  name (en|em|thin|figure|dash|open|close|1-8|3-8|5-8|7-8|1-4|3-4|1-3|2-3|1-2)
#REQUIRED
>

<!-- ===== -->
<!-- IMAGE - include a logo, halftone, or other graphic -->
<!-- -->
<!-- depth is the image depth measured in agate lines. -->
<!-- -->
<!-- name can be blank to set vertical space in ad. Otherwise, -->
<!-- name and depth must match a library image that is available -->
<!-- to advertiser for this ad (by class, etc.). -->
<!-- -->
<!-- generic="yes" limits name matching to generic images. -->
<!-- -->
<!-- image causes a break images are required to occupy the -->
```

```
<!-- entire ad width. Note that images are not subject to -->
<!-- settings. -->
<!-- ===== -->
<!ELEMENT image EMPTY>
<!ATTLIST image
  depth CDATA #REQUIRED
  generic (no|yes) "no"
  name CDATA ""
>
```

```
<!-- ===== -->
<!-- KEYWORD - bind ad text to ad coding -->
<!--
<!-- name specifies the tag for a coding element, which must be -->
<!-- present in the same ad. -->
<!--
<!-- format gives rule for expressing coding value as text. -->
<!-- For example, format="m/dd/yyyy" could be used for a date. -->
<!--
<!-- scale is used to adjust the coding value before formatting. -->
<!-- For example scale="1000" is used to express mileage as -->
<!-- as 'k miles' because the coding value is divided by 1000. -->
<!--
<!-- punct is appended to the formatted keyword value, but only -->
<!-- when the value has a non-zero length. Otherwise, punct is -->
<!-- omitted. -->
<!--
<!-- NOTICE. The interior of the keyword element is pre-filled -->
<!-- with the formatted coding value, so downline users don't -->
<!-- need special handling for keyword. -->
<!--
<!-- keyword is non-breaking. -->
<!-- ===== -->
<!ELEMENT keyword (#PCDATA)>
<!ATTLIST keyword
  format CDATA ""
  name CDATA #REQUIRED
  punct CDATA ""
  scale CDATA ""
>
```

```
<!-- ===== -->
<!-- LEFT - left-justify text within prevailing margins -->
<!--
<!-- left causes a break. -->
```

```
<!-- Interior content is flowed to best fit. -->
<!-- ===== -->
<!ELEMENT left (%inline;|reply)*>

<!-- ===== -->
<!-- LINE - mark off a line for special spacing or tabulation -->
<!-- -->
<!-- The line interior is set to full justification to the -->
<!-- prevailing margins. The <space/> or <tab/> elements, if -->
<!-- present, influence how the line is justified. Absent these -->
<!-- tags, excess line space is divided evenly among the -->
<!-- interword spaces. See <space/> and <tab/> for comments on -->
<!-- how they affect <line>. -->
<!-- -->
<!-- line causes a break. -->
<!-- ===== -->
<!ELEMENT line (%inline;|%spacer;|reply)*>

<!-- ===== -->
<!-- MAILBOX - publication-assigned reply forwarding code -->
<!-- -->
<!-- agent is the identifier for the forwarding agent. -->
<!-- -->
<!-- If the advertiser has arrange for reply forwarding, it is -->
<!-- necessary to include a forwarding code in the ad. This -->
<!-- allows the reply forwarder to match a reply with the -->
<!-- appropriate forwarding information. The advertiser uses -->
<!-- <mailbox> to mark the location where the forwarding code -->
<!-- is to appear in the ad. -->
<!-- -->
<!-- NOTICE. The interior of the mailbox element is pre-filled -->
<!-- with the assigned code, so downline users don't need -->
<!-- special handling for mailbox. Despite this, downline -->
<!-- users, if they accept ads that use forwarding should -->
<!-- provide forwarding agent-specific reply instructions. -->
<!-- -->
<!-- mailbox is non-breaking. -->
<!-- ===== -->
<!ELEMENT mailbox (#PCDATA)>
<!ATTLIST mailbox
  agent CDATA ""
>

<!-- ===== -->
```

```

<!-- MARGIN - establish margins -->
<!--
<!-- Margin units are agate en spaces. -->
<!--
<!-- hang is the hanging indent margin. That is, the left -->
<!-- margin for all but the first line. If hang="none", then -->
<!-- the left margin prevails for all lines. Otherwise the -->
<!-- left margin prevails for the first line and the hang -->
<!-- margin prevails for all other lines. -->
<!--
<!-- left is the left margin. If left="same", the left margin -->
<!-- in effect prior to <margin> is used. -->
<!--
<!-- right is the right margin. If right="same", the right -->
<!-- margin in effect prior to <margin> is used. -->
<!--
<!-- The initial margins are left="0", right="0", and -->
<!-- hang="none". -->
<!--
<!-- margin causes a break. -->
<!-- ===== -->
<!ELEMENT margin (%inline;|%flow;|reply)*>
<!ATTLIST margin
  hang (none|0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15) "none"
  left (same|0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15) "same"
  right (same|0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15) "same"
>

```

```

<!-- ===== -->
<!-- REPLY - reply request phrase -->
<!--
<!-- agent is the identifier for the forwarding agent. -->
<!--
<!-- <reply> is a generalization of <mailbox> for the benefit -->
<!-- of advertisers who a) user reply forwarding and b) target -->
<!-- ads to multiple publications. These advertisers can -->
<!-- different reply instructions for each publication. The -->
<!-- <reply> is used to mark the location in the ad text where -->
<!-- the publisher-specific reply instructions are to appear. -->
<!--
<!-- NOTICE. The interior of the reply element is pre-filled -->
<!-- with the text assigned for a particular publisher, so -->
<!-- downline users can avoid special handling for reply. -->
<!-- Despite this, downline users, if they accept ads that use -->
<!-- forwarding should provide forwarding agent-specific reply -->
<!-- instructions. A more sophisticated downline user could act -->

```

```
<!-- as forwarding agent and/or substitute different reply -->
<!-- instructions. -->
<!-- -->
<!-- reply is non-breaking. -->
<!-- ===== -->
<!ELEMENT reply (%inline;)*>
<!ATTLIST reply
  agent CDATA ""
>
```

```
<!-- ===== -->
<!-- RIGHT - right-justify text within prevailing margins -->
<!-- -->
<!-- right causes a break. -->
<!-- Interior content is flowed to best fit. -->
<!-- ===== -->
<!ELEMENT right (%inline;|reply)*>
```

```
<!-- ===== -->
<!-- SPACE - mark point for justification space in a line -->
<!-- -->
<!-- The <space> element is only allowed interior to a <line> -->
<!-- element. Furthermore, if one or more <space> elements are -->
<!-- present in a particular <line> element, no <tab> elements -->
<!-- are allowed. -->
<!-- -->
<!-- The <line> contents are set to full justification at the -->
<!-- prevailing margins. Excess space is distributed evenly -->
<!-- among all interior <space> elements. -->
<!-- -->
<!-- space is non-breaking. -->
<!-- ===== -->
<!ELEMENT space EMPTY>
```

```
<!-- ===== -->
<!-- TAB - split a line into two into left and right halves -->
<!-- -->
<!-- The <tab> element is only allowed interior to a <line> -->
<!-- element. Furthermore, at most one <tab> element is allowed -->
<!-- in a particular <line> element, and no <space> elements -->
<!-- are allowed. -->
<!-- -->
<!-- leader="yes" requests dot leadering between the left and -->
<!-- right text. -->
```

```
<!-- -->
<!-- flow="yes" requests that the left text be flowed to a -->
<!-- second line if the whole <line> is overset. With flow="no", -->
<!-- the left text is set on one line and the right text on the -->
<!-- next when the <line> is overset. -->
<!-- -->
<!-- tab is non-breaking. -->
<!-- ===== -->
<!ELEMENT tab EMPTY>
<!ATTLIST tab
  leader (no|yes) "no"
  flow (no|yes) "no"
>

<!-- ***** -->
<!-- PUBLICATION INFORMATION -->
<!-- ***** -->

<!-- ===== -->
<!-- PUBLICATION - placement data -->
<!-- -->
<!-- name identifies a particular publisher. Interior values, -->
<!-- such as class and zone are in the publication's name space -->
<!-- ===== -->
<!ELEMENT publication (pub_alias?,pub_price?,pub_options?,class*)>
<!ATTLIST publication
  name CDATA #REQUIRED
>

<!-- ===== -->
<!-- PUB_ALIAS - alternate ad id; local to publication -->
<!-- ===== -->
<!ELEMENT pub_alias (#PCDATA)>

<!-- ===== -->
<!-- PUB_PRICE - total ad cost -->
<!-- ===== -->
<!ELEMENT pub_price (#PCDATA)>

<!-- ===== -->
<!-- PUB_OPTIONS - special ad features (usually at extra cost) -->
<!-- ===== -->
<!ELEMENT pub_options (claim?,columns?,forwarding?,tearsheet?,shading?)>

<!-- ===== -->
<!-- CLAIM - base pricing on a different number of inserts -->
```

```
<!-- -->
<!-- Multiple ads ordered at once for the same class can be -->
<!-- counted together as the basis for a combination ad rate. -->
<!-- If claim exceeds the number of inserts ordered, a lower -->
<!-- price may be quoted based on the claimed combination. -->
<!-- Claim does not affect actual billing, so the advertiser -->
<!-- pays the regular rate if the other ad(s) don't run. -->
<!-- ===== -->
<!ELEMENT claim (#PCDATA)>

<!-- ===== -->
<!-- COLUMNS - specify a multi-column ad -->
<!-- ===== -->
<!ELEMENT columns (#PCDATA)>

<!-- ===== -->
<!-- FORWARDING - request ad reply forwarding service -->
<!-- -->
<!-- collect means: -->
<!-- -->
<!-- mail - assign mailbox for us mail replies -->
<!-- voice - assign voice mailbox for telephone replies -->
<!-- email - assign mailbox id for email replies -->
<!-- none - no forwarding -->
<!-- -->
<!-- forward means: -->
<!-- -->
<!-- same - forward replies like they were collected -->
<!-- -->
<!-- hold - hold mail or print and hold email -->
<!-- requires collect="mail" or collect="email" -->
<!-- -->
<!-- mail - forward mail or print and mail email -->
<!-- requires collect="mail" or collect="email" -->
<!-- -->
<!-- fax - forward email via fax -->
<!-- requires collect="email" -->
<!-- -->
<!-- ===== -->
<!ELEMENT forwarding (%inline;|contact_ref|rate)*>
<!ATTLIST forwarding
  collect (mail|voice|email|none) "mail"
  forward (same|hold|mail|fax) "same"
>

<!-- ===== -->
<!-- TEARSHEET - request copy of published ad (torn from paper) -->
```

```
<!-- ===== -->
<!ELEMENT tearsheet (#PCDATA|contact_ref|rate)*>

<!-- ===== -->
<!-- SHADING - request ad background shading -->
<!-- ===== -->
<!ELEMENT shading (rate*)>

<!-- ===== -->
<!-- CLASS - set classification & publishing schedule -->
<!-- ===== -->
<!ELEMENT class (#PCDATA|title|classword|lines|billed_lines|sortkey|zone|rundate)*>

<!-- ===== -->
<!-- CLASSWORD - keywords for loose classification remapping -->
<!-- ===== -->
<!ELEMENT classword (#PCDATA)>

<!-- ===== -->
<!-- LINES - agate lines ad set to for this classification -->
<!-- ===== -->
<!ELEMENT lines (#PCDATA)>

<!-- ===== -->
<!-- BILLED_LINES - agate lines ad billed for this class -->
<!-- ===== -->
<!-- Note: This element is omitted if, as usual, billed lines -->
<!-- equals set lines. They only differ if ad is too small -->
<!-- compared to the minimum size allowed in the class. -->
<!-- ===== -->
<!ELEMENT billed_lines (#PCDATA)>

<!-- ===== -->
<!-- SORTKEY - ad sort key; some classes restrict key choices -->
<!-- ===== -->
<!ELEMENT sortkey (#PCDATA)>

<!-- ===== -->
<!-- ZONE - zone code to select ad audience -->
<!-- ===== -->
<!ELEMENT zone (#PCDATA|title)*>

<!-- ===== -->
<!-- RUNDATE - yyyyymmdd date for add plus publication record -->
<!-- ===== -->
<!ELEMENT rundate (#PCDATA|rate|instance)*>
```

```

<!-- ===== -->
<!-- INSTANCE - publication record and related details -->
<!-- ===== -->
<!ELEMENT instance (edition,section,page,column,offset)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT section (#PCDATA)>
<!ELEMENT page (#PCDATA)>
<!ELEMENT column (#PCDATA)>
<!ELEMENT offset (#PCDATA)>

```

```

<!-- ***** -->
<!-- MISCELLANEOUS ELEMENTS -->
<!-- ***** -->

```

```

<!-- ===== -->
<!-- RATE - record of pricing rule as it relates to this ad -->
<!-- -->
<!-- basis give invoice explanation for rate. -->
<!-- -->
<!-- unit means: -->
<!-- -->
<!-- line - rate quoted is per agate line billed -->
<!-- ad - rate quoted is once per ad -->
<!-- recipient - rate is per tearsheet recipient -->
<!-- standard - percentage increase of all standard charges -->
<!-- -->
<!-- type means: -->
<!-- -->
<!-- actual - rate is applied to final ad price -->
<!-- comparison- rate is not applied; ad earns discount rate -->
<!-- -->
<!-- ===== -->
<!ELEMENT rate (#PCDATA)>
<!ATTLIST rate
  basis CDATA #REQUIRED
  unit (line|ad|recipient|standard) #REQUIRED
  type (actual|comparison) "actual"
>

```

```

<!-- ===== -->
<!-- TITLE - the looked-up title of a coded value (e.g. zone) -->
<!-- ===== -->
<!ELEMENT title (#PCDATA)>

```

## Glossary

### Attribute

In XML, a source of additional information about an element. For example, the *forwarding* element takes the *collect* attribute:

```
<forwarding collect="email">
  Please email replies to <mailbox>T1234</mailbox>@nytimes.com
</forwarding>
```

### Element

In XML, a fully-formed application of a tag. For example:

```
<class>
  3720
</class>
```

Elements may be nested, where appropriate. For example:

```
<class>
  3720
  <date>
    19980719
  </date>
</class>
```

*Not to be confused with NAA's use of the term Data Element.*

## Entity

In XML, a remote specification. For example:

```
<!ENTITY % inline "#PCDATA|font|glyph|image|keyword|mailbox|margin">
<!ELEMENT center (%inline;|reply)*>
```

is equivalent to:

```
<!ELEMENT center (#PCDATA|font|glyph|image|keyword|mailbox|margin|reply)*>
```

## Tag

In XML, a piece of markup. Tags may be *opening* tags, such as `<class>` or *closing* tags, such as `</class>`.