SAS® Intelligent Advertising for Publishers: User's Guide

SAS® Documentation
March 29, 2018

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2014. *SAS® Intelligent Advertising for Publishers: User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Intelligent Advertising for Publishers: User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

2015

P1:iapug

Contents

Part 1

Introduction to SAS Intelligent Advertising for Publishers

Chapter 1

About SAS Intelligent Advertising for Publishers

What Is SAS Intelligent Advertising for Publishers?

SAS Intelligent Advertising for Publishers is a complete online publishing solution that includes sales order management, simulation-based forecasting, ad serving, and business intelligence capabilities to help you make better-informed decisions faster across the entire business. SAS Intelligent Advertising for Publishers gives publishers complete control of the online advertising process by providing a comprehensive solution that supports multiple platforms, including mobile, display, and video output devices.

SAS Intelligent Advertising for Publishers enables digital publishers to perform the following actions:

- manage sales orders
- simulate and forecast inventory levels
- optimize pricing and target ad delivery
- provide billing and invoicing reports
- visualize and analyze ad delivery details

Chapter 2

Setting up Single Sign-On (SSO) with the Identity Provider

Introduction

Single sign-on (SSO) provides a way of authenticating users in SAS Intelligent Advertising for Publishers through a third-party identity provider (IDP). The communication between SAS Intelligent Advertising for Publishers and the IDP uses Security Assertion Markup Language (SAML). To configure single sign-on, you must set up the IDP and the Service Provider (SP) (that is, SAS Intelligent Advertising for Publishers) to exchange required data.

Here are the terms used for the SSO process:

Identity Provider (IDP)
: Provides identities for users who need to interact with a system. The IDP is usually the host of the user information repository that handles authentication and password management.

Security Assertion Markup Language (SAML)
: Enables "single sign on" by allowing a single-user authorization service (or identity provider) to grant access to any number of third-party or remote applications. For example, when SAML is set up and configured, the identity provider can provide credentials to log in to applications such as Airbrake, PagerDuty, or Slack.

Service Provider (SP)
: Provides features or functionality to users. The IDP, rather than the SP, manages user authentication.

User Agent
: Usually refers to a web browser.

IDP Data that is Required by the SP

The SP must receive the following information from the IDP:

- the "IDP single sign on" URL to redirect users for authentication
- the IDP issuer URL
- a copy of the IDP X.509 certificate

These values can be entered manually or provided by a metadata XML file to the SP. Also, the Security Assertion Markup Language (SAML) uses the **Name ID format** value to enable the IDP and SP to exchange information about a user, and to identify the user. SASIA expects the Name ID to be **Persistent**. (The Name ID can be configured to have a transient value).

SP Data that is Required by the IDP

The IDP must receive the following information from the SP:

- the SP URL to post authentication information, that is, the SP "consumer" or "SP single sign on" URL
- the default SP landing page, that is, the "Audience URI" or "SP Entity ID". This is typically the home page for the website
- authentication user payload information that is sent to the IDP in the format required by the SP. An SP defines the user attributes (that is, “Claims”) that are acceptable. For example, the attributes might be "login" instead of "username", "first_name" instead of "FirstName", or "email" instead of "email_address". If necessary, the IDP transforms its user data to the expected format before posting it to the SP consumer URL.

  Note: SAS Intelligent Advertising for Publishers expects to receive the following attributes from the IDP (case sensitive):

  - FirstName
  - LastName
  - Email
  - Login

  SAS Intelligent Advertising for Publishers requires the four attributes and the persistent name ID from the IDP when posting to the SSO consumer.

Configuring the SSO in SAS Intelligent Advertising for Publishers

Note: Because the IDP configuration requires information from the SP, and the SP requires information from the IDP during setup, you might want to initially create the SSO configuration with only the name and default role and blank URL fields before you configure the IDP. Configuring the SSO in this manner enables you to have the required information before you get started.

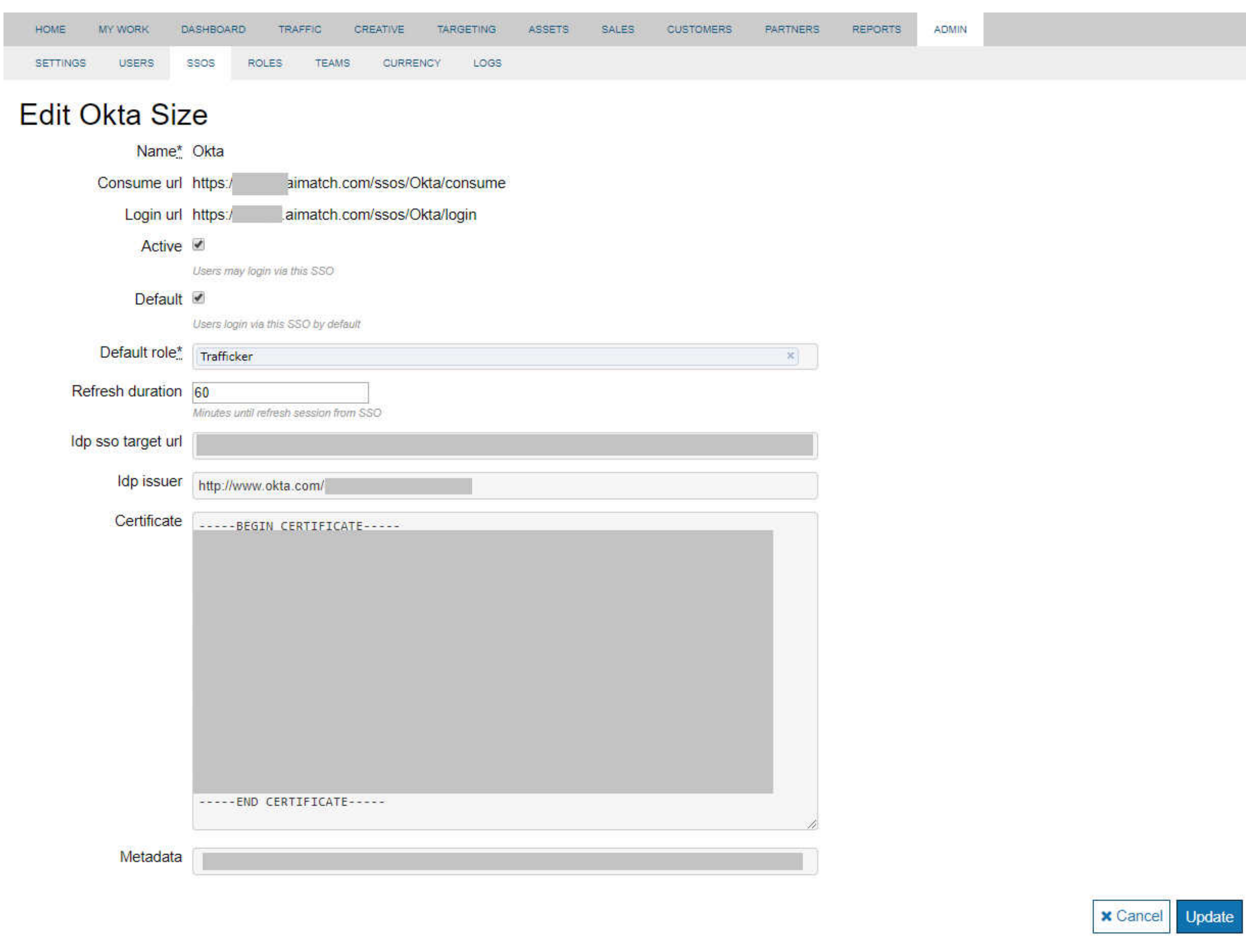
In SAS Intelligent Advertising for Publishers, SSO is configured in the **SSOs** panel on the **Admin** tab. The fields in the **SSOs** panel are automatically populated after metadata information is copied into the **Metadata** field that appears when you first select **Admin** ⇨ **SSOs**. After an SSO configuration has been created and activated, navigating to the URL of the UX will redirect to the IDP's login screen (if the user is not logged in).

After logging in through the IDP, a new user account is created in SAS Intelligent Advertising for Publishers if the email address provided by the IDP for the user does not exist in SAS Intelligent Advertising for Publishers. If the email address does exist, the existing user account in SAS Intelligent Advertising for Publishers is converted to an SSO account and now must be used in conjunction with the SSO. In other words, a user account that has logged in through the SSO service can no longer log in to SAS Intelligent Advertising for Publishers in the traditional manner.

As a result, it is important to create at least one user account with Administrator access that is NOT used for SSO. Creating a user account with Administration access ensures that if the IDP is down, at least one account is still able to login.

Here are the fields in the **SSO** panel:

Figure 2.1 *SSO Panel*



Name

The unique name for the SASIA installation. The name must be URL safe because the SSO name is part of the "SP single sign on URL" that is provided to the IDP

Note: This field is Read-only after you create the SSO

Active
: Select to indicate that users can authenticated with the SSO.

Default
: Select to indicate that anonymous users will be authenticated with this SSO by default instead of another SSO or the application local application login.

Note: If **Default** is deselected, users can only log in using the local application.

Default role
: Specify the default role for new SAS Intelligent Advertising for Publishers users signing in from the SSO. Groups, roles, and permissions are not declared by the SSO and must be managed within SAS Intelligent Advertising for Publishers.

Refresh duration
: Specify the amount of time that SAS Intelligent Advertising for Publishers waits before re-verifying the authentication information with the SSO. An employee who has left the company will be inauthenticated at the end of this time duration.

Note: The minimum duration is 15 minutes.

Idp sso target url
: This information is provided by the IDP. For more information, see “IDP Data that is Required by the SP” on page 6.

Idp issuer
: This information is provided by the IDP. For more information, see “IDP Data that is Required by the SP” on page 6.

Certificate
: This information is provided by the IDP. For more information, see “IDP Data that is Required by the SP” on page 6.

Metadata (optional)
: The URL this is used to upload IDP information, if the SSO record was created by that method.

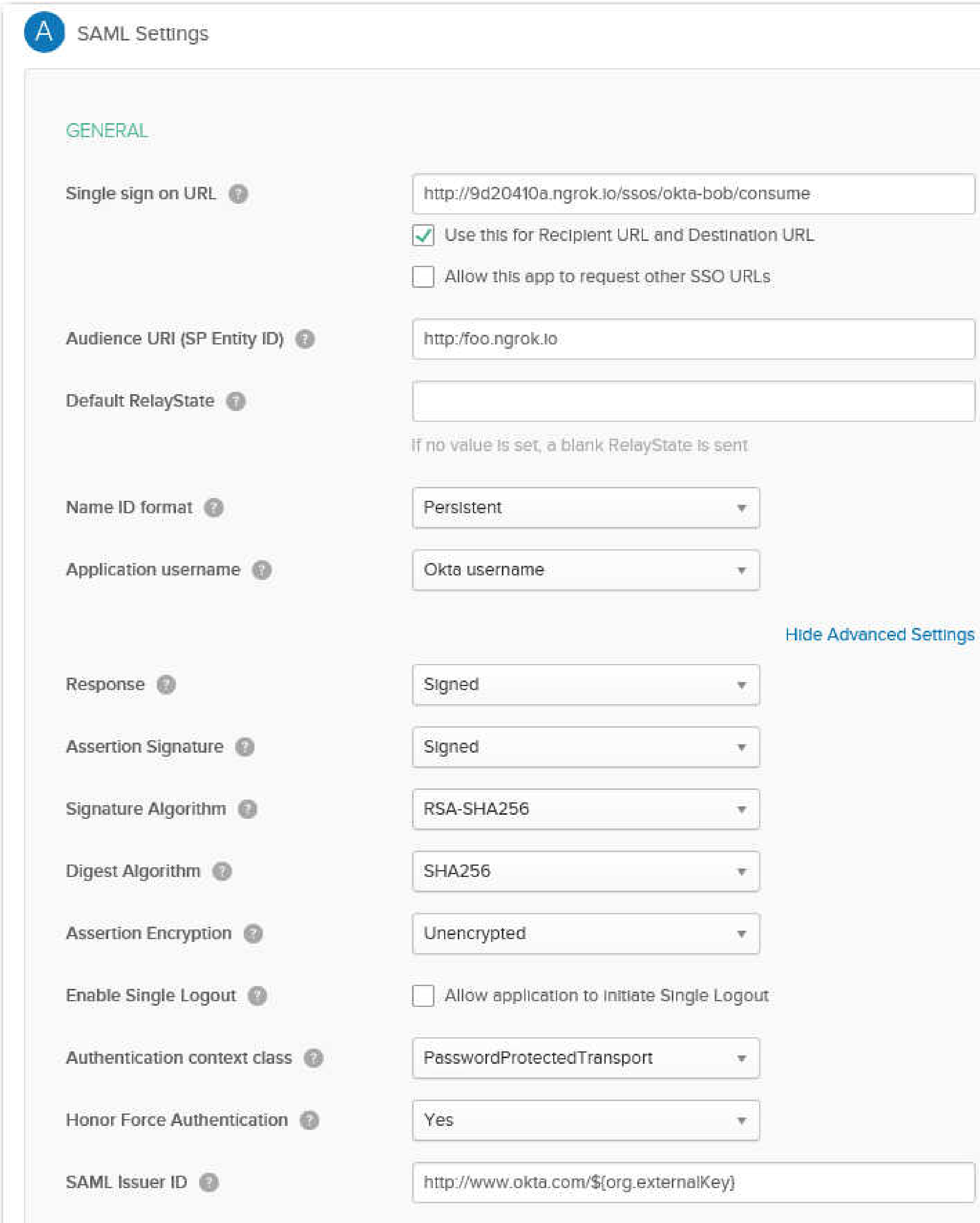
Click **Update** after making desired changes. After the IDP and the SP are configured and enabled, authentication by the SSO starts automatically.

Setting up Okta as the IDP

To set up Okta as the IDP:

1. Log into Okta.
2. Select **Add Applications** in the**Shortcut** menu on the dashboard.
3. Click **Create New App**.
4. Click **Show Advanced Settings** to display all the fields in the **SAML Settings** panel.

Figure 2.2 *SAML Settings Panel*



5. In the **SAML Settings** panel, follow these steps:

   a. Enter information in the **Single sign on URL** field.

      Note: The value in this field must match the value in the **Consume URL** field in SAS Intelligent Advertising for Publishers

To set the same URL as a recipient and destination, select **Use this for Recipient URL and Destination URL**. If **Use this for Recipient URL and Destination URL** is deselected, continue with step b. Otherwise, skip to step d.

b. Enter information in the **Recipient URL** field. For example, the value for SAS Intelligent Advertising for Publishers is https://signonuser.aimatch.com/ssos/Okta/consume/.

c. Enter information in the **Destination URL** field. For example, the value for SAS Intelligent Advertising for Publishers is https://signonuser.aimatch.com/ssos/Okta/consume/.

d. Enter information in the **Audience URI (SP Entity)** field.

   Note: The value in the **Audience URI (SP Entity)** field must be the base user interface URL for the SP with no trailing slash. For example, the value for SAS Intelligent Advertising for Publishers is https://signonuser.aimatch.com.

e. (Optional) Enter information in the **Default RelayState** field. If no value is set, a BlankRelay state is sent.

f. Select **Persistent** from the **Name ID** format list.

g. Select **Email** from the **Application username** list

h. Select **Signed** from the **Response** list

i. Select **Signed** from the **Assertion Signature** list.

j. Select **RSA-SHA256** from the **Signature Algorithm** list

k. Select **SHA256** from the **Digest Algorithm** list.

l. Select **Unencrypted** from the **Assertion Encryption** list.

m. Deselect **Allow application to initiate Single Logout** in the **Enable Single Logout** field.

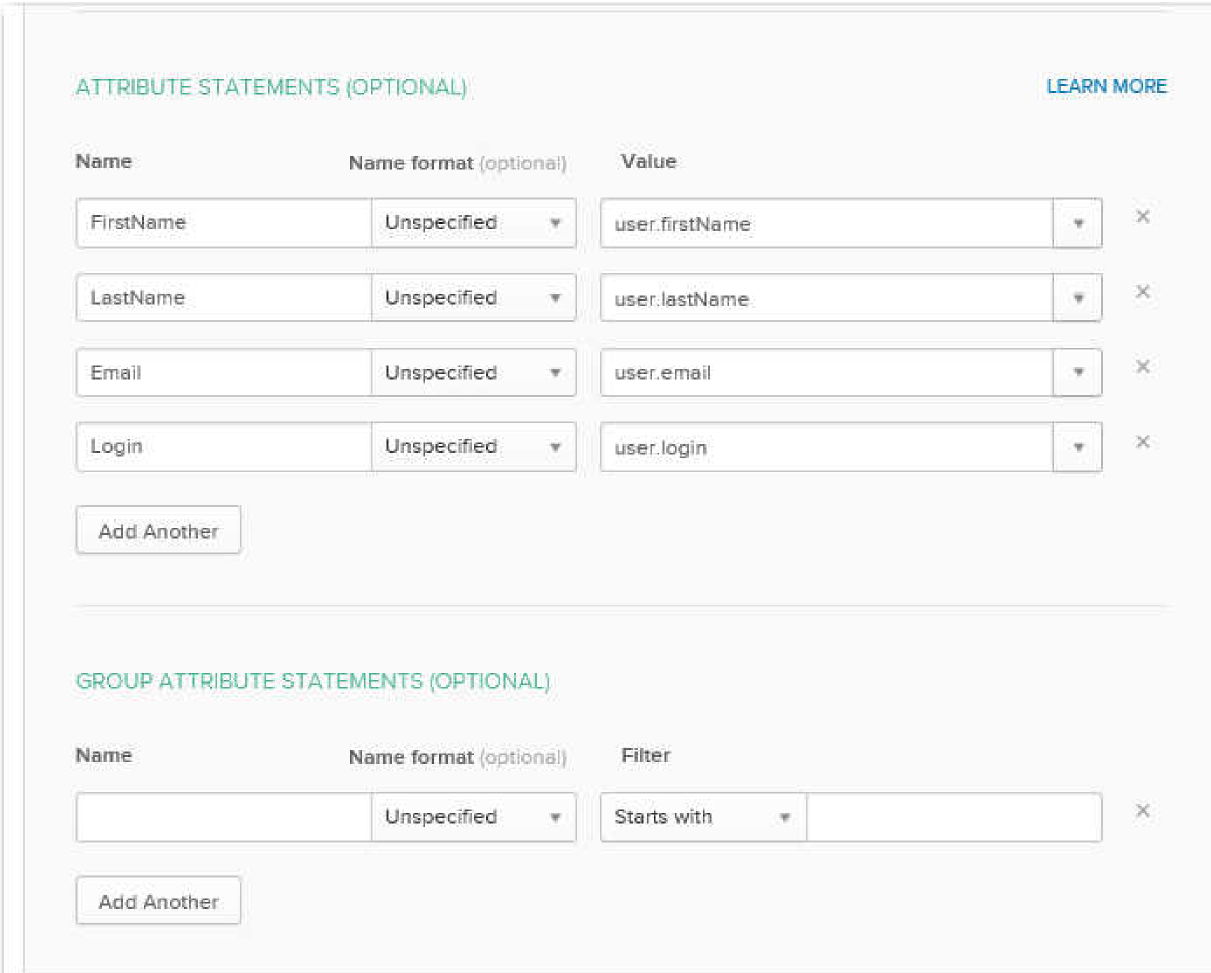
n. Select **PasswordProtectedTransport** from the **Authentication context class** list

o. Select **Yes** in the **Honor Force Authentication** list.

p. Enter *http://www.okta.com/${org.externalKey* in the **SAML Issuer ID** field.

q. (Optional) On the **ATTRIBUTE STATEMENTS** panel, select name formats and values for the attributes. For each additional user that you want to add, click **Add Another** and enter values as desired.

Figure 2.3 *Optional SAML Attributes*



r. (Optional) On the **GROUP ATTRIBUTE STATEMENTS** panel, enter information and select from the lists. For each additional group that you want to add, click **Add Another** and enter values as desired.

s. Click **Next**.

t. Click **Finish**.

Chapter 3

Navigating the User Interface

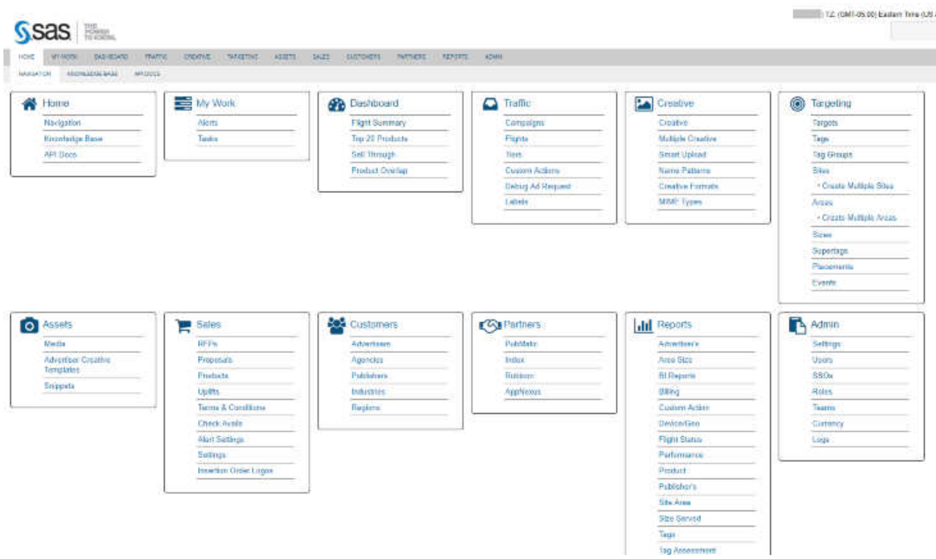
Log In to SAS Intelligent Advertising for Publishers

1. In the address bar of your web browser, enter the URL for SAS Intelligent Advertising for Publishers (for example, http://shrtnme.aimatch.com where shrtnme is the name of your company) and press **Enter**. The Login page appears.

   Note: Contact your system administrator if you need the URL for SAS Intelligent Advertising for Publishers.

2. Enter a **Login** ID and **Password**.

3. Click **Login**. The Home screen appears.



Navigating the User Interface

After you log on, the Home page for SAS Intelligent Advertising for Publishers appears. The tabs at the top of the page are the main navigational links that appear on every page. You might have a Quick Start panel to the far right containing a Quick Start tutorial, a link to the Knowledge Base, and a Processing Status section. Any field that requires input is denoted by an asterisk. All other fields are optional.

Home Tab

The **Home** tab appears by default when you log on to SAS Intelligent Advertising for Publishers. The following table lists and describes the tabs on the **Home** tab:

Table 3.1 *Overview of the Home Tab*

Tab	Description
Navigation	Displays links to all the tabs in the application.
Knowledge Base	Provides access to the Knowledge Base. You can click **Open knowledge base in a separate window** to open the Knowledge Base in a separate window.
API Docs	Provides links to information about the XML REST API documentation for SAS Intelligent Advertising for Publishers. For more information, see Appendix 1, “XML API Examples,” on page 195.

My Work Tab

You use the **My Work** tab to display your associated alerts and ongoing and completed tasks. Alerts can be created for price, inventory, change order, or credit rules. For more information, see “About Alerts” on page 151.

From the **My Work** tab, you can filter to view particular alerts, view only your alerts, or view all team alerts. Select the check boxes beside multiple alerts and click the **Multi Edit** button to update or clear the selected alerts. You can also click at the end of the row of an alert to clear an alert. You can either request approval for the proposal or line item change if you are the requester, or approve (or reject) the request if you are the approver. The approver is specified on the **Alert Settings** subtab under the **Sales** tab. Once approval is requested or approved, the button is no longer displayed. Alternatively, the alert can be cleared by clicking the **Just clear it** button. Notes that are saved with the alert can be added whether approval is requested or it is cleared.

Tasks can be viewed in the **Tasks** tab of the My Work section. Click **View My Background Tasks** to view the tasks that are being processed in the background.

Dashboard Tab

The **Dashboard** tab displays performance charts and graphs that are automatically generated for each client. The following table lists and describes the tabs on the **Dashboard** tab:

Table 3.2 *Overview of the Dashboard Tab*

Tab	Description
Flight Summary	Displays an at-a-glance analysis of flight progress. You can also drill down on a flight in the chart. The chart is updated every half-hour. Flights that have an end date and a status of Open or Paused are displayed in the chart. Flights that have no end date are not displayed in order to reduce clutter in the chart. The two dashed, vertical red lines represent the OnSchedulePriority value (default value = 1.0) and 75% of that value. The solid, vertical red line indicates the point of time at which the flight is 80% complete. Flights that appear in the upper left quadrant of the chart are the first flights that you should examine because they are at or near completion and behind schedule. *Note:* Flights whose end date exceeds the end date of inventory projection forecasts display inaccurate projection numbers.
Top 20 Products	Displays the top 20 products with the most inventory available, along with the product sales history.

Tab	Description
Sell Through	Illustrates and compares trends in revenue, impressions, customers, and daily effective Cost Per Thousand (eCPM) rate for all flights of a selected flight type.
Product Overlap	Displays the top products with revenue at risk based on the opportunity cost (sales rate booked versus floor rate specified for the product). The dashboard enables you to select one of those products and drill down to view products that are overlapping and consuming inventory based on flights that serve to the same inventory.

Traffic Tab

The **Traffic** tab contains all trafficking-related functions. The **Campaigns** tab is selected by default when the **Traffic** tab is selected. The following table lists and describes the tabs on the **Traffic** tab:

Table 3.3 *Overview of the Traffic Tab*

Tab	Description
Campaigns	Displays all the campaigns that are loaded in the system, including the inactive ones. Campaigns contain flights. The list can be filtered by Name, Status, and Advertiser. For more information, see “About Campaigns” on page 27.
Flights	Displays all the flights that are loaded in the system. The list can be filtered by expanding the **Filtering** panel. Closed flights are not displayed by default, but can be displayed by selecting the Closed status as a filter. Flights contain delivery goals, options, and creatives. On this tab, you can edit, copy, view, and research flights. You can also view targeting for a flight. You can create flights only in a campaign. For more information, see “Add a Flight to a Campaign” on page 30.
Tiers	Lists all tiers, both active and inactive, that are set up in the system. Tiers organize the ad queue into a hierarchy, enabling you to control which flights are more important and therefore are evaluated more often. For more information, see “About Tiers” on page 43.

Tab	Description
Custom Actions	Lists all custom actions that are set up in the system. Custom actions are typically visitor-triggered event markers. For more information, see "About Custom Actions" on page 51.
Debug Ad Request	Makes an ad call to the engine and receives debugging information. For more information, see "Debugging an Ad Request" on page 53.
Labels	Adds custom fields and values to a flight. These custom fields are available through the API and BI reporting. For more information, see Chapter 9, "Managing Labels," on page 55.

Creative Tab

The **Creative** tab contains all creative-related functions. The following table lists and describes the tabs on the **Creative** tab:

Table 3.4 *Overview of the Creative Tab*

Tab	Description
Creative	Lists all creatives that are loaded in the system. Creatives are the actual ad content delivered to the page. The list can be filtered by Name and Advertiser. For more information, see "About Creatives" on page 59.
Multiple Creative	Creates multiple creatives. For more information, see "About Multiple Creatives" on page 63.
Smart Upload	Parses third-party files and sets up multiple creatives. For more information, see "About Smart Upload" on page 65.
Name Patterns	Creates names that are associated with creatives that are uploaded from third-party vendors. For more information, see "About Name Patterns" on page 67.

Tab	Description
Creative Formats	Lists all creative formats that are loaded in the system. Creative formats are based on two main elements: the format template (the code that is served to the visitor when a creative using this format is selected for delivery), and the accompanying format fields (which contain user-supplied data that is specific to each creative). Creative formats enable you to create your own standard creative templates. For example, if you have several types of text links that you use all the time, you can create a format for each link so that you do not have to re-create them each time. For more information, see "About Creative Formats" on page 69.
MIME Types	Lists custom MIME types that have been set up. This enables you to specify HTTP content types when a creative is delivered, which is useful for some browsers or applications. For more information, see "About MIME Types" on page 77.

Targeting Tab

The **Targeting** tab contains all targeting and demographic related items. When you select the **Targeting** tab, the **Targets** tab appears by default. The following table lists and describes the tabs on the **Targeting** tab:

Table 3.5 *Overview of the Targeting Tab*

Tab	Description
Targets	Lists all the targets that are loaded in the system. Targets are collections of demographic data and tags that can be connected using Boolean logic. Demographic data is implemented as tags in the system. Targets can be thought of as reusable rule sets that can represent audiences or parts of audiences. Targets are also an important building block for Products. The list can be filtered by entering a search query in the **Name like** field. For more information, see "About Targets" on page 81.

Tab	Description
Tags	Lists all tags and their values in the system. The list can be filtered by expanding the **Filtering** panel and entering information in the **Name like** and **Tag group name contains** fields. Tags are key=value pairs that are used for targeting. This list does not include predefined geotargeting tags. For more information, see "About Tags" on page 87.
Tag Groups	Lists all tags and their values in the system. Tags are key=value pairs that are used for targeting. This list does not include predefined geotargeting tags. For more information, see "About Tag Groups" on page 91.
Sites	Lists all site values that are loaded in the system. Sites can function as a tag. For more information, see "About Sites" on page 93.
Areas	Lists all area values that are loaded in the system. Areas can function as a tag. For more information, see "About Areas" on page 95.
Sizes	Lists all creative sizes that are loaded in the system. For more information, see "About Sizes" on page 97.
Supertags	Lists special tags that can be included in an ad call and which is expanded into one of any number of arbitrary collections of other tags and values. Supertags can be used to ease a customer's migration from other ad servers, and provides a level of tag management to ad operations. Essentially, supertags provide a way of mapping a value to an arbitrary pathinfo snippet. For more information, see "About Supertags" on page 99.
Placements	Lists placements that partners can bid on. Placements can also be configured to use several partners. For more information, see Chapter 23, "Managing Placements," on page 101.
Events	Lists events that allow an event name to be associated with one or more timestamps that correspond to a visitor action (such as visiting a certain page). For more information, see Chapter 24, "Managing Events," on page 105.

Assets Tab

The **Assets** tab contains all creative media (such as images or rich media) and creative code (such as HTML or Javascript). When you click the **Assets** tab, the **Media** tab appears by default. The following table lists and describes the tabs on the **Assets** tab:

Table 3.6 *Overview of the Assets Tab*

Tab	Description
Media	Lists all creative media that are uploaded in the system. The list can be filtered by advertiser. Selecting a media entry enables you to download the asset and view the creatives that use the asset. For more information, see "About Media" on page 109.
Advertiser Creative Templates	Lists all advertiser-specific creative templates that are uploaded in the system. Creative templates are typically third-party tags, and HTML and Javascript code. These assets are used in the **HTML** field of a creative. The list can be filtered by name and advertiser. Selecting a creative template enables you to upload code, edit or enter code, and view the creatives or Name that use the asset. For more information, see "About Advertiser Creative Templates" on page 111.
Snippets	Lists snippets that can be added to a creative. For more information, see "About Snippets" on page 117.

Sales Tab

The **Sales** tab contains the Sales Workflow portion of SAS Intelligent Advertising for Publishers. The following table lists and describes the tabs on the **Sales** tab:

Table 3.7 *Overview of the Sales Tab*

Tab	Description
RFPs	Lists all request for proposals (RFPs) that are loaded in the system. For more information, see "About RFPs" on page 121.
Proposals	Lists all proposals that are loaded in the system. For more information, see "About Proposals" on page 125.

Tab	Description
Products	Lists all products that are loaded in the system. Products are constructed from rate card data and targeting, and enable proposals, line items, and availability to be used. It is helpful to have a robust product catalog to enhance the process. A thorough set of key/values (taxonomy) results in effective targets and products. An efficient product list makes managing proposals and tracking availability easier. For more information, see “About Products” on page 135.
Uplifts	Lists uplifts that can be added to a line item. Uplifts consist of a tag and a list rate that can be applied to line items. Products or specific values of the tag can be added to uplifts. For more information, see “About Uplifts” on page 141.
Terms and Conditions	Lists all terms and conditions that are loaded in the system. For more information, see “Add Terms and Conditions” on page 145.
Check Avails	Queries simulation data and provides the number of impressions available to be booked or sold during the specified date range for a specified set of targeting or product criteria. For more information, see “About Checking Inventory Availability” on page 147.
Alert Settings	Displays workflow-related alert options. For more information, see “About Alerts” on page 151.
Settings	Sets options for the sales workflow. For more information, see “About Settings” on page 153.
Insertion Order Logos	Uploads insertion order logos. For more information, see “About Insertion Order Logos” on page 155.

Customers Tab

On the **Customers** tab, you set up customers such as advertisers or agencies in the system. User accounts are not set up on the **Customers** tab. The following table lists and describes the tabs on the **Customers** tab:

Table 3.8 *Overview of the Customers Tab*

Tab	Description
Advertisers	Lists all advertisers that are loaded in the system. For more information, see "Add an Advertiser" on page 159.
Agencies	Lists all agencies that are loaded in the system. For more information, see "About Agencies" on page 165.
Publishers	Lists all publishers that are loaded in the system. This tab is typically used for clients that have a network of different sites (publishers). For more information, see "About Publishers" on page 169.
Industries	Lists all industry classifications that are loaded in the system. For more information, see "About Industries" on page 173.
Regions	Lists all regions that are loaded in the system. Regions can be assigned to advertisers and agencies. For more information, see "About Regions" on page 175.

Partners Tab

You use the **Partners** tab to set up the third-party integration. For more information, see "About Partners" on page 179.

Reports Tab

You access the operational reports and Business Intelligence (BI) reports on the **Reports** tab. For more information, see "About the Reports Tab" on page 185.

Admin Tab

You perform administrative tasks on the **Admin** tab. The following table lists and describes the tabs on the **Admin** tab:

Table 3.9 *Overview of the Admin Tab*

Tab	Description
Settings	Specify targets in the **No Log Targets** field to prevent requests that match from being logged.

Tab	Description
Users	Enables administrators to create individual accounts for users so that actions performed by the user are identifiable and logged. Individual accounts also enable administrators to easily remove users from the system (for example, when they leave the company) without having to modify the password for all users.
SSOs	Enables administrators to create a Single sign-on (SSO) to provide a way of authenticating users in SAS Intelligent Advertising for Publishers through a third-party identity provider (IDP). For more information, see Chapter 2, “Setting up Single Sign-On (SSO) with the Identity Provider,” on page 5.
Roles	Enables administrators to assign roles to users. These roles enable administrators to limit or grant access to the user interface and to report for groups of user accounts. Every user in the system must be assigned at least one role, but each user can be assigned up to five roles in combination with teams. Each role has a set of configurable permissions.
Teams	Enables administrators to group user interface users and to associate these groups with specific Products. These user groupings might then be restricted to interact with only Products to which they are assigned and Products that are unassigned.
Currency	Enables administrators to modify conversion rates for predefined currencies that are specified in Flights and Line Items.
Logs	Enables administrators to view activity history logs for users.

Part 2

Ad Trafficking

Chapter 4

Managing Campaigns

About Campaigns

The hierarchy of a campaign in SAS Intelligent Advertising for Publishers reflects the elements that usually make up an order from one of your customers. The *campaign* is the container (the overall order) that holds *flights* (the line items that are the detailed delivery information) which, in turn, hold creatives (the text, images, video, and so on, that are served). You can copy and send tag information for campaigns and flights.

Add a Campaign

1. On the top navigation bar, click the **Traffic** tab. The **Campaigns** subtab appears by default and lists campaigns currently in your system. If you are returning to the **Campaigns** subtab after a search filter has been applied, only the filtered list of campaigns is displayed.

   TIP In the **Sales** tab, a salesperson works with proposals and line items. When these are converted into an order, you can automatically transfer proposals into campaigns and line items into flights.

2. Click **Add New Campaign**.
3. In the **Name** field, enter a name for the campaign.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.,?;:) in the **Name** field.

4. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the autosearch results. The advertiser that you select is the one that you use when creating the campaign.

To create a new advertiser for the campaign, click **Add a new advertiser**. For more information, see “Add an Advertiser” on page 159.

Note: Enter % if you want to display the full list of advertisers. You can also use the following options: ^*keyword* searches for the keyword is at the beginning, $*keyword* searches for the keyword is at the end, + at the beginning includes the keyword, and – at the beginning excludes the keyword.

5. To add an agency, enter the name in the **Agency** field. The agency must already have been defined on the **Customers** tab. .

   To create a new agency for the campaign, click **Add a new agency**. For more information, see “Add an Agency” on page 165.

   Note: To display the full list of agencies, enter %.

6. From the **Type** list, select one of the following campaign types. Although the names are fixed in the solution, you might use them to define campaigns to meet your business needs. You can report on these campaign types, so you should ensure consistency across the system. Selecting a campaign type does not make it more important in the ad queue; it is just a label for reporting. For a house-filler campaign, select **House**.

   - **Guaranteed**

     This type is usually applied to campaign flights that have impression goals and a required end date.

   - **Pre-emptible**

     This type is usually applied to campaign flights that are redirects to third-party networks or remnant inventory.

   - **Sponsorship**

     This type is usually applied to campaign flights that are consuming all the impressions for a product or target for a specific amount of time.

   - **House**

     This type is usually applied to campaign flights that create no direct revenue. For example, you might apply the House type to ads for another site you also own or to a house-filler campaign.

   - **Barter**

     This type is usually applied to ads that create no direct revenue. For example, you might run ads for a charitable organization.

   - **Make good**

     This type is usually applied to campaign flights that are attempting to complete a contracted volume of impressions that did not complete in the allotted time.

7. Select a **Salesperson** or **Trafficker** or both from the lists. You can select more than one trafficker.

   Defining a salesperson and a trafficker helps with reporting. They are the named users that have access to view these campaigns. If you are not entering a campaign as a trafficker user, then you should select a name here. Only the named trafficker or users with AdOps Manager or Administrator permissions can adjust or view a campaign.

   To add a new salesperson, click **Add a new Salesperson**.

8. (Optional) Select a **Trafficker 2** from the list.

9. Select a start date. When you click the **Start date** field, a calendar appears. Select the date on which the campaign should start.

10. Select an end date. When you click the **End date** field, a calendar appears. Select the date on which the campaign should end.

    An end date is optional but the advertiser usually provides one. You should define an end date to ensure proper scheduling.

11. Enter information in the **External ID**, **Description**, and **Notes** fields. These fields are used for reporting purposes and integrations in the XML API.

12. Expand the **Send Reports** panel and then do the following:

    a. To send a campaign report to selected users at the end of the campaign, select **Reports on end**.

    b. To send a campaign report to selected users at the end of the week, select **Reports at week end**.

    c. To send a campaign report to selected users at the end of the month, select **Reports at month end**.

    d. To mail campaign reports to a salesperson, select **Salesperson**.

    e. To mail campaign reports to a trafficker, select **Trafficker**.

    Note: The email addresses for the salesperson and trafficker are stored with their login details.

    f. To mail campaign reports to other users, enter the users' email addresses in the **Other** field.

    Note: Use a semicolon (;) to separate multiple addresses.

    g. To define the file type for the campaign report attachment that is emailed, select **CSV** or **EXCEL** from the **Attach as** list.

13. To apply a frequency cap (sometimes referred to as *controlled exposure*), enter the information in the **Frequency Caps** panel. Any capping rules that are applied here apply to all flights, creatives, and creative formats included in this campaign.

    Note: You can also apply capping rules at the flight, creative, or creative format level. It is recommended that you do not apply caps to house-filler campaigns.

    In the **Frequency Caps** panel, do the following:

    - Enter a value in the **Quantity** field to specify the number of impressions, clicks, or conversions to serve to a visitor.

    - Enter a value in the **Period** field to specify the period of time in which the number of impressions, clicks, conversions can be delivered before stopping delivery. This is based on the time of the first impression and then the application accounts for *x* of them for the next *n* days. Once the *n* days pass, another *n*-day period starts upon the next impression. For example, if you entered `3` in the **Quantity** field and `24 hours` in the **Period** field, up to 3 impressions in a 24-hour period are shown.

    - Enter a value in the **Over Lifetime** field to specify the number of impressions, clicks, or conversions that can serve to a visitor over the lifetime of the flight.

    - Enter a value in the **Per Session** field to specify the number of conversions that can serve to a visitor over the course of a visitor session. A visitor session is initiated whenever an ad request is made, and, by default, persists for 30 minutes after the visitor's last request.

14. Click **Add New Flight** to add a flight to the campaign. You can add the flight now or add it later. For more information, see "Add a Flight to a Campaign" on page 30.
15. Click **Save Campaign**. A message indicating that a campaign has been successfully created is displayed in the All Campaign window. By default, the new campaign appears at the top of the list.

To view a summary of the information for the new campaign, click on the campaign name. Progress information also appears when the campaign starts delivering. You can copy and send tag information for the campaign by clicking **Embed Code**. A pop-up window appears with both HSERVER and JSERVER tags to allow trafficking of IA campaigns or flights in other ad-serving systems.

To edit the campaign, click in the row for the campaign, make desired edits, and then click **Save Campaign**.

To delete a campaign, click in the row for the campaign, and then click **OK**.

To copy a campaign, click in the row for the campaign. You can then copy just the campaign, the campaign and its associated flights, or the campaign, associated flights and the shift remainders.

Add a Flight to a Campaign

A *flight* contains the detailed instructions, such as targeting options, and the creative that a campaign should deliver. A campaign must contain at least one flight, but it can contain many flights, depending on the advertiser's instructions. Flights can serve only within the date ranges applied at the campaign. You can copy and send tag information for campaigns and flights. To create a flight:

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Campaigns** subtab. Campaigns currently in your system are listed. If you are returning to the **Campaigns** tab after a search filter has been applied, only the filtered list of campaigns is displayed.
3. Click on the name of the campaign to which you want to add a flight.
4. Click **Add New Flight**. The Edit New Flight page appears. The campaign name and the advertiser are listed at the top of the page.
5. In the **Name** field, enter a name for the flight.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.,?;:) in the **Name** field.
6. From the **Status** list, select one of the following:
   - **Open**

     The flight can be served through the ad server.
   - **Pending**

     The flight automatically starts being served as soon as the start date or start time is reached, or when the status is changed to **Open**.
   - **Closed**

The flight will not be served.

- **Missing creative**

  No creatives are assigned to the flight, so the flight is not eligible to be served.

- **Paused**

  The flight is paused and resumes being served after the status is changed to **Open**.

- **Alerted**

  The **Reports on end** option is selected in the **Send Reports** panel of the Edit Flight window. When the flight has completed, its status is set to **Alerted** to indicate that the end-of-flight email alerts have been sent.

7. Select a size from the **Reservation Size** list. Reservation sizes allow flights that might be missing a creative to still be included in the forecasting simulation and removed from the available inventory.

8. Select a label from the **Flight Type** list. The flight type is applicable only in reporting and inventory forecasting.

9. To set the flight's general position in the ad queue, select from the **Tier** list. For more information, see "About Tiers" on page 43.

10. Select the time zone for the flight from the **Time Zone** list.

    Note: By default, the time zone is based on your user setting or on the system time zone if your user setting does not have a time zone set.

11. To select a start date for the flight, click the **Start date** field, and then select the date on which the flight should start. Next, select from the **Hour** list.

12. To select an end date for the flight, click the **End date** field, and then select the date on which the flight should end. Next, select from the **Hour** list.

13. To enable the flight to be served during a specific time window each day, select **Daily Window** and then select a **Start Hour** and an **End Hour**.

14. Select a **Currency** from the list. This is the currency the flight is billed in, which can be different from the system base currency. For example, this might be a flight built by your team in London and billed out in pounds instead of dollars. The associated **Conversion rate** is displayed.

15. To assign a product that contains targeting demographics tied with revenue data to the flight, enter information in the **Product** field.

    Note: A well-defined set of products can keep you from having to select other targeting modifiers in the rules section.

16. Select a **Product Line** from the list. This provides a way to define a genre of products for reporting purposes.

    Line items can be categorized into predefined product lines by using the **Predefined Product Line** option. If the **Predefined Product Line** option is enabled, a **Product Line** list appears in the Create Line Item or Edit Line Item windows. This option enables you to group line items into categories such as *Mobile* or *Video*. The product lines can then be filtered and reported on in BI reports. The Product Line categories are custom-defined. Contact Technical Support to enable the option and to specify the required Product Line values.

    Note: Enter % if you want to display the full list of product lines.

17. To set up reports to be emailed at specified intervals, expand the **Send Reports** panel and then do the following:

    a. To send a flight report to selected users at the end of the campaign, select **Reports on end**.

    b. To send a flight report to selected users every Monday at midnight, select **Reports at week end**.

    Note: The date and time are determined by the time zone used by the application.

    c. To send a flight report to selected users at midnight the first day of every month, select **Reports at month end**.

    d. To mail campaign reports to a salesperson, select **Salesperson**.

    e. To mail campaign reports to a trafficker, select **Trafficker**.

    Note: The email addresses for the salesperson and trafficker are stored with their login details.

    f. To mail campaign reports to other users, enter the users' email addresses in the **Other** field.

    Note: Use a semicolon (;) to separate multiple addresses.

    g. To define the file type for the campaign report attachment that is emailed, select **CSV** or **EXCEL** from the **Attached as** list.

18. To include a flight description and notes about the flight in reports, expand the **Description** panel and then enter information in the **Description**, **Notes**, **User Info**, and **External ID** fields.

19. Expand the **Goals & Revenue** panel and then do one of the following:

    - To apply a flat rate revenue to the flight, select **Flat Rate** and then do the following:

        - Enter a value in the **Flat Rate** field. The revenue is not calculated per impression but by the amount entered.

        - Enter a value in the **Goal** field for impressions, views, clicks, and actions.

          The flight attempts to serve enough to meet the goals specified in the fields in the time specified and as evenly as possible. When the goals are reached, the flight stops serving.

          A views goal can be set on flights and used when the viewability feature is enabled for the page where the flight is serving.

          Enter a value in the **Bonus** field for impressions, views, clicks, and actions.

          Revenue is not applied to bonus amounts. The amounts are treated as part of the overall goal by the ad server.

    - To apply a cost to every thousandth instance of an impression, click, or action, select **Cost Per** and then do the following:

        - Enter a value in the **Cost Per** field for the clicks per thousand for impressions, and cost per single views, clicks, and actions.

        - Enter a value in the **Cost Per** field for the **Daily Revenue Cap**.

          The flight serves normally each day until reaching the revenue amount that is specified in this field.

        - Enter a value in the **Goal** field for impressions, views, clicks, and actions.

The flight attempts to serve enough to meet the goals specified in the fields. When the goals are reached, the flight stops serving.

A views goal can be set on flights and used when the viewability feature is enabled for the page where the flight is serving.

- Enter a value in the **Bonus** field for impressions, views, clicks, and actions.

  Revenue is not applied to bonus amounts. The amounts are treated as part of the overall goal by the ad server.

The Effective Cost Per (**ECP**) is automatically calculated and displayed in the table.

20. To perform an ad hoc projection of all pre-emptible inventory using the product specified on the flight, expand the **Inventory** panel, and then click **Available**.

    Note: This calculation incorporates any default ads that appear in the forecast and match the flight's criteria. Results are split into Available, Reserved, Projected, and Served numbers. Contending flights can be used when ad inventory is not available.

    To include contending flights in the inventory projection, click **Contending Flights**. To set a goal for the inventory, click **Set Goal** and then enter a value.

21. To set options that affect the frequency limits, expand the **Frequency Caps** panel, and then do the following:

    - Enter a value in the **Quantity** field to specify the number of impressions, views, clicks, or conversions to serve to a visitor.
    - Enter a value in the **Period** field to specify the period of time in which the number of impressions, views, clicks, conversions can be delivered before stopping delivery. This is based on the time of the first impression and then the application accounts for X of them for the next N days. Once the N days pass, another N-day period starts upon the next impression. For example, if you entered `3` in the **Quantity** field and `24 hours` in the **Period** field, up to 3 impressions in a 24-hour period are shown.
    - Enter a value in the **Over Lifetime** field to specify the number of impressions, views, clicks, or conversions that can serve to a visitor over the lifetime of the flight.
    - Enter a value in the **Per Session** field to specify the number of impressions, views, clicks, or conversions that can serve to a visitor over the course of a visitor session. A visitor session is initiated whenever an ad request is made, and, by default, persists for 30 minutes after the visitor’s last request.

22. To set options that affect delivery of the flight, expand the **Delivery Configuration** panel, and then do the following:

    - To force the flight to serve as quickly as possible by making it the highest priority flight in the tier and bypassing any type of pacing, select **Top priority**. This option also prevents the flight from being skipped over if it has been seen recently by the visitor.

    CAUTION:
    Enabling this option can cause the flight to deliver its impression goal before the end date.

    - To append HTML code for a 1x1 image in the server response when a creative is served from the flight, select **Beacon count**.

      An impression is not counted until the 1x1 image is rendered by the browser. This ensures that the creative code is completely processed first by the browser.

This option is useful when serving third-party creatives because it can minimize impression discrepancies between third-party ad servers and SAS Intelligent Advertising for Publishers.

Select **Frequency cap based on beacon** to allow impression-based frequency capping to be based on beacon counts rather than actual flight serves.

Note: Do not select **Beacon count** for a flight that delivers ads to emails because the email reader does not allow the beacon count URL to be written into the page. Therefore, the delivery to the ad would not be counted.

- In the **Creative selection method** list, select one of the following methods:
  - **Weighted** (default)

    When there are multiple creatives in a flight, the weight (or ratio) is used to determine how often to serve each creative. By default, every creative has a weight of 1, so each creative is served equally.
  - **Sequential**

    The sequential method enables the “weights” to be used as a number sequence, so that it is possible to display multiple creatives in a specified order when the flight is served. The sequence restarts with each new user session.

    Note: The sequential method does not always start with the first ad in the sequence if the number sequential ad requests is less than the number of sequential ads. For example, if there are two sequential ad requests on the page and three sequential ads in the flight, the first pageview might deliver ads 1 and 2, but a refresh of the page might show ad 3 followed by ad 1.
  - **CTR Optimized**

    The CTR optimized delivers the creative with the best click-through rate (CTR) over creatives with lower CTRs.
  - **Score Optimized**

    Select an optimization method and enter a sample size for this method. The sample size must be a positive integer.
  - **Prioritized**

    When this option is selected, the creatives can be sequenced in a particular order similar to the Sequential method. With Prioritized, the creatives are always evaluated in the specified order, every time, starting with the first. If the first cannot serve, the second is evaluated, then the third, and so on. The flight is skipped only if all of the creatives have been checked and none can serve.
- In the **Companion type** list, select one of the following types:
  - **None** (default)

    The companion feature is disabled.
  - **Strict**

    The strict type requires two or more creatives to be present in the flight. This option enables the “companion” or “roadblock” feature, which provides a method of serving multiple creatives from the same flight to the same page at the same time. Flights with strict specified serve only if you select it as the first ad request on the page. Ad requests come in grouped by a VIEWID and

Cookie ID. For example, if you view a page that has five ads, all of the ads on that page have the same VIEWID. If the first ad evaluated matches one of the ads in the companion flight, it is served. However, if the second one in the group does not match, ads from other companion flights are served.

- **Loose**

  The loose type means that creatives are required to satisfy only some of the ad calls on the page. If the first ad is served from the companion but the second is not, the application still tries to serve the ad from the companion flight on sequential ad requests. Loose is the recommended option most of the time.

Flights containing frequency caps can result in a flight with a Companion Type policy (Strict or Loose) being canceled on the page after serving an ad, causing other flights to deliver to the subsequent ad calls on the page. Strict or Loose flights deliver together consistently when frequency capping is not invoked.

- (Optional) In the **Sticky duration** field, enter a value to denote the amount of time that you want the creative to persist.
- (Optional) In the **On schedule priority** field, enter a value to denote the prioritization level. Double-click the field name to enable it.
- (Optional) Using the **Delivery patience** slide control, you can make adjustments to account for changes in the schedule.

23. To place flights in a shared category so that they do not serve at the same time to the same page, expand the **Categories** panel, and then select an item in the **Categories** list. Advertisers cannot display ads at the same time when they belong to the same category. However, since a categorized advertiser does not compete with itself, an advertiser can serve more than once to a given page. Categories are created under the Edit Advertiser section. For more information, see "Edit an Advertiser" on page 162.
24. To define where or to whom a flight can serve and cannot serve, expand the **Targets** panel.
    - From the **Targets group logic** list, select **Match All**, **Match Any**, **Don't Match All**, or **Don't Match Any**.

      Match Any is the equivalent of using the Boolean OR operator. If any of the statements are true, the target matches. **Match All** is the equivalent of using the Boolean AND operator. All statements must be true for the target to match.

      Don't Match All is the equivalent of using both the Boolean OR and NOT operators. **Don't Match Any** is the equivalent of using both the Boolean AND and NOT operators. For example, if you want to include any of several target site values but want to exclude a specific target, select **Don't Match All** with those targets specified. The target can be excluded without creating a new negative target or statement.
    - In the **Targets** field, search for and select a target. Targets are set up by clicking the **Targeting** tab and then the **Targets** tab.
25. In the **Rules** panel, do the following:
    a. Select **Match All** or **Match Any**.

       Match Any is the equivalent of the Boolean OR operator. If any of the statements are true, the target matches. **Match All** is the equivalent of the Boolean AND operator. All statements must be true for the target to match".

    b. Select a value or a grouping rule. If you selected a grouping rule, select a value and continue with the next step.

c. Select an operator.

Note: Some operators are unavailable for the selected value.

d. Depending on the value that you selected, enter a value or select from search results in the field.

Note: You can enter *%* to specify a wildcard of any length or an underscore to specify a wildcard of a single character.

e. If you want to add another grouping rule or value, click .

26. Expand the **Sites & Areas** panel. By default, all sites and areas are selected to be served by the flight.

   - To specify sites that the flight can serve to, select **Selected Sites**, place your mouse pointer in the **Sites** field, search for sites, and then select them. To add a site that the flight can serve to, click **Add a new site**, enter information in the required **Name** field, and in the optional **Description** and **Revenue cut** fields, and then click **Add Site**.

   - To specify areas that the flight can serve to, select **Selected Areas**, place your mouse pointer in the **Areas** field, search for areas, and then select them. To add an area that the flight can serve to, click **Add a new area**, enter information in the required **Name** field and the optional **Sites** field, and then click **Add Area**.

27. Expand the **Tags** panel.

   - Click the **Custom**, **Geo**, or **Device** tab.

   - Select a tag name from the list and then the desired values. If you select multiple values, any of the values match.

   - Click **Add Tag**.

   - To add more tags, repeat these steps.

   Note: If you create multiple tags, every tag and at least one of the values from each has to match in order for the flight to serve.

28. Expand the **Third Party** panel, select from the **Name** list, and enter information in the **Third Party External ID** field.

   In the **Third Party External ID** field, you can enter a single reference or order number that can be applied and viewed in both flight and line item windows.

29. Expand the **Action Policies** panel.

   Action Policies are used as event markers. They are typically used to determine whether a visitor goes to a certain page after seeing or clicking on an ad.

   Custom action tracking enables you to create and track custom actions (for example, tracking whether a website visitor has viewed 50% of a video) and requires you to complete the Action Policies section, whereas standard action tracking does not. Standard action tracking simply requires a value entered in the **Cost Per Action** field in the **Goals and Revenue** section of the Edit Flight page.

   To add an action policy, do the following:

   - In the **Action** list, select a custom action that you want to be tracked.

   - In the **Track Post Impressions Period** field, enter a time value such as *6 hours* to specify how long a visitor has to trigger the action after having received this flight. The action will not be counted for a visitor who triggers the action beacon after the specified time period has elapsed.

- In the **Track Post Clicks Period** field, enter a time value such as *6 hours* to specify how long a visitor has to trigger the action after having clicked a creative from this flight. The action will not be counted for a visitor who triggers the action beacon after the specified time period has elapsed.
- In the **Revenue $** field, enter a value for the amount charged for each custom action.
- If you want a click action to supersede an impression, select **Allow Click to Trump Impression**.
- Click + **Add Policy**.

30. Expand the **Labels** panel.

    Labels provide a way of adding custom fields and values to a flight. These custom fields are available through the API and BI reporting.

    Click **Add New Label** and enter a name and a display name. Click **Add Label**.

    Once added, click **Add Label Value** and enter a value for the label once selected.

31. Expand the **Creative** panel. All creatives belonging to the advertiser set for the flight are listed.

    Creatives are the actual media and content objects that are delivered to the page. After a creative is added to a flight, that relationship is called a *flight creative*. The same creative can be added only once to the same flight.

32. To select existing creatives for the flight, select the creatives from the displayed list and then click **Add Selected Creative**. To add a new creative to the flight, click **Add New Creative**, and then enter desired information in the fields and make desired selections. For more information, see "Add a Creative" on page 59. The creatives that you have added to the flight are listed in the Flight Creative table and are assigned a Flight Creative ID (FCID) that is unique to the flight.

33. To preview an added creative, do the following:

    a. Select the check box next to the desired flight creative and click **Preview**. The Live Preview Flight Creative dialog box appears.

    b. Enter a URL to evaluate any ad calls on the page to see which ones this flight could serve to. If an ad call matches, the page is rendered with this flight creative showing where the matching ad call is.

    c. Click **Live Preview**. Advertisers can use the preview to provide screen shots of the ad "live" on the page to their clients. Click **Close** when the preview is finished.

    Note: By default, Live Preview uses the creative's size as the only targeting parameter from the ad calls on the page. If the ad calls on a page do not contain a size value, then the creative is displayed in each of the ad call positions on the page. Clients can request a configuration option called PreviewHonorsTargeting that instructs SAS Intelligent Advertising for Publishers to use all the targeting parameters in the ad call on the page.

34. To edit an added creative, click  in the row for the creative, make desired edits, and then click **Save Changes**.

    The **Tag Qualifier** field determines which qualified tags are temporarily switched into the evaluation context while evaluating that creative for serving.

    The **Sticky Set Name** field enables a set of creatives to persist rather than just one being served. This enables any of the creatives in the set to serve each time, rather than always favoring the first creative that served.

35. To view the passback URL for an added creative, click in the row for the creative. For more information, see "About Passback URLs" on page 38.

    Note: The displayed URL includes the targeting for only the ad server domain, the HSERVER tag, and the **`/saspb/fcid={fcid value}`** string. The URL does not include the targeting for the flight.

36. To remove an added creative, click in the row for the creative, and then click **OK**.

37. To save the flight, click either **Save Flight**, click the down arrow and select**Update Flight & Check Inventory**, **Update Flight & Add Another**, **Update Flight & Copy**, or **Update Flight & Make Many Copies**.

38. You can copy and send tag information for the flight by clicking **Embed Code**. A pop-up window appears with both HSERVER and JSERVER tags to enable the trafficking of IA campaigns or flights in other ad-serving systems.

About Passback URLs

Passback behavior is triggered by a network redirecting to SAS Intelligent Advertising for Publishers with a tag that includes SASPB and the FCID of the original ad served (for example, **`.../hserver/SASPB/FCID=1234`**). If the FCID is absent, invalid, or exists for an ad that is not in the visitor's recent queue, then a default ad is served, and no true passback is possible.

The SASPB/FCID passback tags that are sent by the network are ideally supplied by a format template that forms them in response to the original ad served (if the network is "cooperative"). Otherwise, the tags must be hardcoded in the trafficking of SAS Intelligent Advertising for Publishers passbacks in the network itself, usually by size.

Assuming that the passback FCID is for a valid ad recently served to the visitor, its associated path information, query string, and blacklisted advertisers are resurrected for the passback ad selection. The ad's advertiser is added to the blacklisted advertisers for this ad selection only. Ad selection starts at the top of the queue and blocks selecting any flight associated with the blacklisted advertisers. Passback requests can express additional or even overriding values for tags.

If the same FCID is passed back multiple times in a short time span, each passback is likely to resurrect the same set of tags for the passback ad because only the most recent pathinfo and other information is retained. This is expected to occur infrequently, and only when a page has multiple ads of the same size. In such cases, the ad tags are not likely to be very different. If this becomes an issue, prudent use of frequency capping at the creative level can mitigate its effect.

When serving an ad in response to a passback, the impression count for the original ad selection is decremented (that is, uncounted), using its timestamp, tags and other information, unless it was not in the visitor's queue. Frequency capping is ignored. No attempt is made to credit a capped creative-flight pair when this creative is uncounted. The ad served in response to a passback is counted normally, using its timestamp and the original ad's tags. The passback itself is not counted. That is, there is currently no separate counting of passbacks that easily enables the reporting of fill rate. During unique visitor (UV) analysis of sample data, no attempt is made to reconcile passbacks with their original flights. Therefore, UV results for a flight includes all visitors who were served that flight, regardless whether all serves of that flight were passed back.

Chapter 5

Managing Flights

About Flights

A *flight* contains the detailed instructions, such as targeting options, and the creative that a campaign should deliver. A campaign must contain at least one flight, but it can contain many flights, depending on the advertiser's instructions. Flights can serve only within the date ranges applied at the campaign. You can copy and send tag information for flights.

View a Flight

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Flights** subtab. All flights currently in your system are listed.
3. Select the flight name and click . The *flight name* page appears. You can edit the flight by clicking beside the flight name or view the history by clicking **View History**.

Copy a Flight

Note: You can create an original flight only in a campaign. For more information, see "Add a Flight to a Campaign" on page 30.

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Flights** subtab. All flights currently in your system are listed.
3. In the row for a flight, click . The Edit *flight name* page appears.
4. If you are copying a flight with a start date that has already occurred, specify a new date in the Save & Copy window.
5. Click **Make Copy**.

Edit a Flight

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Flights** subtab. Flights currently in your system are listed. If you are returning to the **Flights** tab after a search filter has been applied, only the filtered list of flights will be displayed.
3. Click on the name of a flight or click . The Edit *flight name* page appears.
4. Refer to "Add a Flight to a Campaign" on page 30 for information about editing a flight.

   Note: From the table on the Flights page, you can update the start date and end date by clicking on the dates in the row for the flight. You can also click the tier name in the row for the flight to change tiers or to update the weight for the tier.
5. Click **Save Flight** or click the down arrow and select **Update Flight & Check Inventory**, **Update Flight & Add Another**, **Update Flight & Copy**, or **Update Flight & Make Many Copies**.
6. You can copy and send tag information for the flight by clicking **Embed Code**. A pop-up window appears with both HSERVER and JSERVER tags to enable the trafficking of IA campaigns or flights in other ad-serving systems.

Edit Multiple Flights

You can edit one or more flight parameters for multiple flights simultaneously.

CAUTION:

Be extremely careful when selecting flights to edit as there can be serious negative consequences for flights that are selected but are not intended to be edited. Changes cannot be undone.

1. On the top navigation bar, click the **Traffic** tab.

2. Click the **Flights** subtab. All flights currently in your system are listed.
3. To edit all flights, select the check box in the top row. To select individual flights to edit, click the check box for one or more flights in the first column.
4. Click **Multi Edit**.
5. Double-click the label for each field that you would like to edit.

   Note: You can modify only the **Start date**, **End date**, **Tier**, **Type**, **Top priority**, and **Beacon count** fields in the Update Selected Flights window.
6. Refer to "Add a Flight to a Campaign" on page 30for information about editing a flight.

   Note: From the table on the Flights page, you can update the start date and end date by clicking on the dates in the row for the flight. You can also click the tier name in the row for the flight to change tiers or to update the weight for the tier.
7. Click **Update**.

Delete a Flight

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Flights** subtab. All flights currently in your system are listed.
3. In the row for a flight that you want to delete, click and then click **OK**.

View Targeting Details for a Flight

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Flights** subtab. All flights currently in your system are listed.
3. In the row for a flight, click .

View History of Changes to a Flight

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Flights** subtab. All flights currently in your system are listed.
3. Click . The Edit *flight name* page appears.
4. Click **View History**. Changes to the flight are annotated.

View the On-Schedule Percentage for a Flight

The on-schedule percentage (OSP) for a flight is calculated.

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Flights** subtab. All flights currently in your system are listed.
3. Click the on-schedule percentage in the **OSP** column for the flight to display a number of statistics about the flight, including goals, revenue, and CTR for impressions, views, clicks, and actions (for example, less than or equal to 79.0% is red, less than or equal to 89.0% is yellow, less than or equal to 100.0% is green, and greater than 100.0% is blue).

Chapter 6

Managing Tiers

About Tiers

Tiers enable you to designate a hierarchy of flights. The prioritization method of a tier enables you to select what criteria should be used when the engine calculates the priority of flights in a tier. The priority is used to order the flights in a tier so that the engine evaluates them in a certain order. Tiers can be prioritized by Schedule, Click Rate, Action Rate, CPM, eCPM, Weighted, Cost Per Second, Share Of Voice, or Broad Share of Voice. You might consider naming tiers such that your top tiers are easily recognizable. Note that when tiers are created they do not have flights in them. You design the tier structure around the type of flights that you want to put in them. Top tiers are usually schedule-based and more lucrative. You can also have tiers with no flights in them.

Schedule Prioritization Method

Schedule-based tiers are the most common. Scheduling attempts to meet the goals of a flight on the end date specified. If a flight is not on track to achieve this, it is behind schedule and the engine automatically adjusts the priority so that it has a chance to receive more impressions. Conversely, if the flight is serving (or has served) more than it needs to, it is considered ahead of schedule. If a flight does not have a goal, or does not have an end date, a schedule priority cannot be calculated. In this case, the flight is treated as being on schedule, and any flights that are even slightly behind schedule are evaluated first.

Click Rate Prioritization Method

The Click Rate prioritization method orders flights by their calculated click-through rate (CTR). Click-through rate is determined by comparing the number of impressions versus the number of clicks recorded. A higher ratio of clicks to impressions is considered better performing. Flights with a higher CTR are higher priority and are evaluated to serve before flights with a lower CTR. No clicks are recorded for newly created flights, so the CTR and where it is placed in the tier relative to other flights cannot be calculated. To compensate for this, a grace period and priority are set to enable the flight to serve so that the performance of the flight can be established. The grace period priority is used to determine the position of newly created flights in the ad queue relative to established, non-grace period flights. These are set on the Edit Tier window when the Click Rate prioritization method is selected.In Figure 6.1 on page 44, new flights are placed above 80% (0.8) of other flights in the tier for seven days. After this time period has elapsed, the CTR and tier position is calculated as normal. It is recommended to set the grace period length long enough so that a baseline CTR performance can be calculated. The grace period priority is typically set around 0.8 or 0.9 to ensure that the flight receives some traffic, but not too high so that it does not "steal" too many impressions from established flights.

Figure 6.1 *Grace Period Diagram*

Action Rate Prioritization Method

The Action Rate prioritization method orders flights by their calculated action rate. Action rate is determined by comparing the number of actions versus the number of impressions recorded. Action rate prioritization follows the same principles as click rate prioritization regarding grace period and priority calculation.

CPM Prioritization Method

The Cost Per Thousand (CPM) rate is the monetary rate charged per *x* number of impressions, clicks, or actions. These rates are specified in the Edit Flight window. For more information, see “Edit a Flight” on page 40. The Cost Per column sets the CPM rate for every 1,000 impressions and for every single click or action. When the CPM prioritization rate is set for a tier, the higher the total CPM rate (impressions, clicks, and actions), the higher priority the flight is considered. This calculation does not depend on the performance of the flight. CPM rate is a static calculation that is determined entirely from the value in the **Cost Per** field for the flight.

eCPM Prioritization Method

The effective Cost Per Thousand (eCPM) is a prioritization method that is used to determine the best performing flights based on the current number of impressions, clicks, actions, and the respective CPM rates for each. A flight that generates the same overall revenue as another using only half the impressions is going to be considered a better performing flight, and thus is given higher priority. CPM rates are set in the Edit Flight window. Because this prioritization method depends on the performance of flights, a grace period must be set for new flights.

Weighted Prioritization Method

The Weighted prioritization method enables you to weight flights by specifying a ratio to determine the order of evaluation. Once this method is selected, all flights in the tier are assigned an equal weight, which then can be altered. The weight can be edited for a flight in the Edit Flight window. For more information, see "Edit a Flight" on page 40. All flight weights can be viewed and edited by first navigating to the **Flights** tab in the **Traffic** tab, and then by placing the mouse pointer over a tier name that uses the Weighted prioritization method. Click **Edit All Weights**. Here, flight weights can be set, and a calculated percentage is shown next to the weight. Note that targeting can affect the performance of weights. For example, if a flight is weighted at 10% in the tier, but is the only flight targeted to a specific site in that tier, then it effectively serves 100% of the time there. A Weighted tier disregards the Recent Ad Queue (RAQ) for visitors so that the **Serve Soft Match at Tier End** option is effectively enabled even if not selected as a tier option. A flight in the tier can continue to serve repeatedly to a visitor (as long as any frequency caps have not been reached). Delivery pacing for a flight with an end date and impression goal is ignored when placed in a Weighted tier.

Cost per Second Prioritization Method

The Cost Per Second (CPS) prioritization method commonly applies to video ad serving where the revenue and rate associated with a creative can be calculated by the system to a CPS. Flights assigned to a tier that is using the CPS prioritization method then compete and are prioritized based on their CPS value, highest to lowest.

Share of Voice Prioritization Method

When the Share Of Voice (SOV) prioritization method is used, the percentage assigned to each flight indicates the percentage of overall matching traffic that the flight should serve. When a flight is in an SOV tier, the flight setup or edit screen displays a field in which you must assign the percentage. For example, if the highest tier in the system was using SOV and a flight was set to 20% (and the flight had a 728 x 90 creative and no targeting), then it would receive 20% of all 728 x 90 traffic. If the tier was at the bottom, then that flight would receive 20% of the 728 x 90 traffic that made it down to that tier. In a Weighted tier, the tier receives 100% of the traffic regardless and the percentages control how often the flights serve within that tier. With SOV, the percentages control how much overall matching traffic is served by that flight. Overall matching traffic indicates the traffic that matches the targeting for the flight. In an SOV tier the flight is selected based on its SOV percentage but if its targeting does not match the ad request's targeting, the ad server moves down to the next tier, unlike a Weighted tier in which the ad server remains in the tier to find another matching flight. An SOV tier disregards the RAQ for visitors so that the **Serve Soft Match at Tier End** option is effectively enabled even if not selected as a tier option. A flight in the tier can continue to serve repeatedly to a visitor (as long as any frequency caps have not been reached).

Broad Share of Voice Prioritization Method

The Broad Share Of Voice (BSOV) prioritization method is similar to the SOV prioritization method. However, when the BSOV prioritization method is used, flights within the tier are grouped and percentages are set according to their targeting signatures. For example, if Flight A is targeted to a Site called *Home*, Flight B is targeted to a Site called*Home*, and Flight C is targeted to a Site called *Sports*, Flights A and B would have the same targeting signature, and the percentages set for these flights would apply to that targeting group. Consequently, Flight A might be weighted 50% and flight B might be weighted 50%. Because Flight C is targeted differently, it might be weighted 100%. The same setup in a regular SOV tier would require multiple tiers, whereas BSOV allows one tier with many groupings, eliminating the need for additional tiers. To see the division of targeting signatures and the flights that belong to each, edit the BSOV tier and click **Flights**. The Voice column displays the specific targeting signature, and the Voice Total column displays the total SOV that the flights occupy.

Broad Weighted Prioritization Method

Broad Weighted is a combination of the Weighted and Broad Share Of Voice methods. It works like a weighted tier in that the flight weights stipulate what ratio a flight serves in relation to other flights, and in this case, flights with the same targeting signature. This is in contrast to Share Of Voice that specifies the percentage of all ad traffic that is evaluated for that tier. Note that a Broad Weighted tier disregards the Recent Ad Queue (RAQ) for visitors, so the **Serve Soft Match at Tier End** option is effectively enabled even if not selected as a tier option. A flight in the tier can continue to serve repeatedly to a visitor, as long as any frequency caps have not been reached. Delivery pacing for a flight with an end date and impression goal is ignored when placed in a Weighted tier.

Add a Tier

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Tiers** subtab. All tiers currently in your system are listed.
3. Click **+ Add New Tier**. The Add New Tier page appears.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0-9), special characters (~@#$%^&*-+(){}[]|/<>"='), spaces, underscores, and punctuation marks (!.,?;:).
5. Select **Active**, **Pre-emptible**, or **Biddable**.

   If you selected **Biddable**, you can also select **Private Marketplace** to make the tier biddable by only real-time bidding (RTB) campaigns in a private marketplace.

   Note: Selecting **Pre-emptible** allows the inventory allocated for this tier to be available when calculating the available inventory for a Sales Proposal. In other words, the flights in this tier are not to consume inventory when calculating the inventory. Tiers whose flights are catch-all flights such as House flights are typically marked **Pre-emptible**.

Note: If you select **Biddable**, you are allowed to bid if the winning flight is in that tier. If not selected, you do not get to bid.

6. In the **Target** field, enter the first few letters of the target's name and then select the target from the list.

   Note: Enter `%` if you want to display the full list of targets.

7. In the **Prioritization method** list, select one of the following options: **Schedule**, **Click Rate**, **Action Rate**, **CPM**, **eCPM**, **Weighted**, **Cost Per Second**, **Share of Voice**, **Broad Share of Voice**, or **Broad Weighted**.

   If you select **Click Rate**, **Action Rate**, or **eCPM**, then you must also enter a value for the number of days in the **Grace period** field and use the slider to select a value for the **Grace period priority**.

   The grace period applies to the initial placement of a flight in a tier. Flights that have been in the tier for a period of time longer than the grace period are served according to the ad serving algorithm selected for the tier. The schedule and weighted ad serving algorithms do not require a grace period. Flights can be added to the tier on-the-fly so that flights using the grace period can co-exist with flights using the tier's selected prioritization method.

   Selecting `1` for the **Grace period priority** places the flight ahead of all others that are not in their grace period, so this value yields the greatest opportunity. Selecting `0` places the flight after all other flights in the tier.

   After the grace period expires, the flight is served using the specified ad serving method.

8. Select one or more of the following options:

   - **Serve Soft Match at Tier End**

     Note: When this option is selected, SAS Intelligent Advertising for Publishers stays in the tier to serve previously served ads if there are no previously unserved ads that match the targeting. When this option is set, the ad server engine lingers in the tier until the ad request targeting does not match or flights in the tier are ahead-of-schedule and cannot serve again. This option works well for tiers with the top-value flights as they have more opportunities to serve before the engine drops to the next tier.

   - **Eliminate Duplicate Flights**
   - **Eliminate Duplicate Advertisers**

     Note: When you select **Eliminate Duplicate Flights** or **Eliminate Duplicate Advertisers**, SAS Intelligent Advertising for Publishers does not deliver the same flight or advertiser more than once in a series of ad calls with the same viewid values. Setting the viewid values differently in the ad calls on the page would override these settings because the different viewid would indicate a new page to SAS Intelligent Advertising for Publishers, and it could then serve the advertiser or flight again. The **Eliminate Duplicate Flights** and **Eliminate Duplicate Advertisers** settings are global for the tier. You cannot disable the settings for specific flights or advertisers in the tier. A companion ad policy enabled on a flight overrides the **Eliminate Duplicate Flights** setting or the **Eliminate Duplicate Advertisers** setting in a tier.

9. Click **Add Tier**.

View a Tier

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Tiers** subtab. All tiers currently in your system are listed.
3. In the row for a tier, click . The Tier *tier -name* page appears.

Edit a Tier

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Tiers** subtab. All tiers currently in your system are listed.
3. In the row for a tier, click the tier name or click . The Edit *tier-name* tier page appears.
4. Enter information in the **Name** field.
5. Select **Active** or **Pre-emptible** or both.

   Note: Selecting **Pre-emptible** allows the inventory allocated for this tier to be available when calculating the available inventory for a Sales Proposal. In other words, the flights in this tier are not to consume inventory when calculating the inventory. Tiers whose flights are catch-all flights such as House flights are typically marked **Pre-emptible**.
6. In the **Target** field, enter the first few letters of the target's name and then select the target from the list.

   Note: Enter `%` if you want to display the full list of targets.
7. In the **Prioritization method** list, select one of the following options: **Schedule**, **Click Rate**, **Action Rate**, **CPM**, **eCPM**, **Weighted**, **Cost Per Second**, **Share of Voice**, or **Broad Share of Voice**. For more information,

   If you select **Click Rate**, **Action Rate**, or **eCPM**, then you must also enter a value for the number of days in the **Grace period** field and use the slider to select a value for the **Grace period priority**.

   The grace period applies to the initial placement of a flight in a tier. Flights that have been in the tier for a period of time longer than the grace period are served according to the ad serving algorithm selected for the tier. The schedule and weighted ad serving algorithms do not require a grace period. Flights can be added to the tier on-the-fly so that flights using the grace period can co-exist with flights using the tier's selected prioritization method.

   Selecting `1` for the **Grace period priority** places the flight ahead of all others that are not in their grace period, so this value yields the greatest opportunity. Selecting `0` places the flight after all other flights in the tier.

   After the grace period expires, the flight is served using the specified ad serving method.
8. Select one or more of the following options:

- **Serve Soft Match at Tier End**
- **Eliminate Duplicate Flights**
- **Eliminate Duplicate Advertisers**
- **Multi variant**

9. If you want to update the tier weight for the flights associated with the tier, click **Flights**, enter a value in the box in the Tier Weights column, and then click **Change Flight Weight**.
10. Click **Update**.

Rank Tiers

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Tiers** subtab. All tiers currently in your system are listed.
3. Click **Rank Tiers**.
4. To change the rank of a tier, select it and then move it up or down in the list.
5. Click **Save Changes**.

Delete a Tier

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Tiers** subtab. All tiers currently in your system are listed.
3. In the row for a flight, click  and then click **OK**.

   Note: A tier containing flights cannot be deleted.

Chapter 7

Managing Custom Actions

About Custom Actions

Custom Actions are events other than the standard metrics of impressions, clicks, and actions that you want to track. Usually, actions are used to attribute an impression or click to an event. For example, if a visitor clicks on an ad, and then purchases the product that was being advertised, this sequence of events can be counted as an action. All advertisers start with a default set of custom actions.

Add a Custom Action

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Custom Actions** subtab.
3. Click **+ Add New Action**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), periods, underscores, and hyphens.

5. If you want to specify advertisers that you want the action applied to, deselect **For all advertisers** and then enter information in the **Advertisers** field.
6. Click **Add Action**.

Edit a Custom Action

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Custom Actions** subtab.
3. Click the custom action name or click in the row for the custom action. The Edit *custom action name* Custom Action page appears.
4. Enter information in the **Name** field.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores.
5. If you want to specify advertisers that you want the action applied to, deselect **For all advertisers** and then enter information in the **Advertisers** field.
6. Click **Update**.

Delete a Custom Action

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Custom Actions** subtab.
3. In the row for a custom action, click and then click **OK**.

Chapter 8

Debugging Ads

About Debugging Ads

Requesting an ad debug enables you to troubleshoot flight delivery issues that might be the result of targeting or flight policies such as frequency capping. The placement ID and name are displayed when evaluating network integration placements.

Debugging an Ad Request

Note: You can contact SAS Technical Support to configure the user interface for the **Debug Ad Request** to restrict access to Personally Identifiable Information [PII]. By default, the user interface allows access to PII.

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Debug Ad Request** subtab.
3. Enter the targeting parameters in the **Path Info** field.

   Note: You can customize the **Cookie** and **User Agent**, and **Referer** values if needed. If you do not enter a **`mid=`** cookie value, you need to select **Run** twice as the first execution of the request is to get a **`mid=`** cookie. For debugging purposes, you can enter any alphanumeric string as a valid mid cookie value (for example, `0001245678A`).

4. From the **Trace** list, select **Normal Ad Selection** or **All Ad Selection**. The Normal Ad Selection option returns session information as well as the reasons a flight or creative is selected or not selected. The All Ad Selection option returns evaluation information and all flights or creatives that match the targeting.
5. Click **Run**.

   The information in the **Cookie** field might be updated and a return code and results appear at the bottom of the page.

Chapter 9

Managing Labels

About Labels

Labels provide a way of adding custom fields and values to a flight. These custom fields are available through the API and BI reporting.

Add a Label

1. On the top navigation bar, click the **Traffic** tab.

   Labels provide a way of adding custom fields and values to a flight. These custom fields are available through the API and BI reporting.
2. Click the **Labels** subtab.
3. Enter a name and a display name. Click **Add Label**.

Delete a Label

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Labels** subtab.
3. In the row for a label, click  and then click **OK**.

Part 3

Ad Creative

Chapter 10

Managing Creatives

About Creatives

A *creative* is displayed to a customer as an advertisement. A creative must be assigned to a flight in order to be served by the ad server.

Add a Creative

1. On the top navigation bar, click the **Creative** tab.
2. Click **+ Add New Creative**.
3. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), underscores, and hyphens. You cannot use spaces in a creative name.

   TIP You do not have to enter the filename for the creative. Adding more description to the name might make it easier to find. For example, you might want to enter `My_Creative_1`.

4. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the list. The advertiser that you select is the one that you use when creating the campaign.

   Note: Enter `%` if you want to display the full list of advertisers.

5. Enter information in the **Description**, **Notes**, and **User Info** fields. If you are adding a video creative, enter a value for the number of seconds the ad will run in the **Duration** field.
6. Select one or more **Categories**. Category values can be added when editing an advertiser. If a creative with a category is served, no other flight or creative with the same category can serve to the same page view (or VIEWID) for that visitor.
7. In the **Media** panel, select a format based on the type of creative that you are delivering from the **Creative Format** list. For more information, see "About Creative Formats" on page 69.
8. Enter information in the fields that are specific to the selected creative format.
9. Click **Add Creative**. If the upload is successful, a green success bar appears. To view the creative, scroll down the page and then select in the row for the creative.

Preview a Creative

1. On the top navigation bar, click the **Creative** tab.
2. Click in the row for the creative. Click **Close**.

Note: Depending on how the roles in your profile are configured, you could see options to **View Team Creatives** or **View My Creatives**.

Edit a Creative

1. On the top navigation bar, click the **Creative** tab.
2. Click the creative name or click in the row for the creative. The Edit Creative — *creative name* page appears.
3. Modify the information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), underscores, and hyphens. You cannot use spaces in a creative name.

   TIP You do not have to enter the filename for the creative. Adding more description to the name might make it easier to find. For example, you might want to enter `My_Creative_1`.
4. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the autosearch results. The advertiser that you select is the one that you use when creating the campaign. Enter `%` if you want to display the full list of advertisers.

   Note: Be careful not to change the advertiser if it has already been assigned to a flight.
5. Enter information in the **Description**, **Notes**, and **User Info** fields. If you are adding a video creative, enter a value for the number of seconds the ad will run in the **Duration** field.

6. In the **Media** panel, select a format from the **Creative Format** list based on the type of creative that you are delivering. For more information, see "About Creative Formats" on page 69
7. Enter information in the fields that are specific to the selected creative format.
8. (Optional) In the **Flights** panel, add the creative to an existing flight. For more information, see "Add a Creative to an Existing Flight" on page 61.
9. (Optional) In the **Flight Creative** panel, remove the creative from a flight, or edit a flight. For more information, see "Manage Flight Creatives" on page 61.
10. Click **Update Creative** or click the down arrow and select **Update Creative & Copy** or **Update Creative & Make Many Copies**.

Add a Creative to an Existing Flight

1. On the top navigation bar, click the **Creative** tab.
2. Click the creative name or click in the row for the creative.
3. In the **Flights** panel, click on the next to **Available Flights** to update the list of flights.
4. Refine the list of flights by filtering flights using **Flight Name like**, **Campaign Name like**, or **Status** fields.
5. Select a flight from the list.
6. Click **Add Selected Flights**. The flight appears in the Flight Creative table. For more information, see "Manage Flight Creatives" on page 61.
7. If you want to apply date restrictions or special targeting options to the creative, click the in the row for the flight, enter information in the fields, and then click **Save Changes**.
8. Click **Update Creative**. A green success bar appears.
9. To indicate that the flight is now eligible for service, do the following:
    a. On the top navigation bar, click the **Traffic** tab.
    b. Click the **Flights** subtab.
    c. In the row for the flight to which you have added a creative, click on the arrow next to **Missing Creative** in the Status column.
    d. Select **Pending**. Within a few moments, the solution identifies the change and moves the campaign into service if it is within the date range that you have specified.

Manage Flight Creatives

1. Expand the **Creative** panel in the **Edit** *flight name* window, .

2. To add a new flight creative, click **+ Add New Creative** from the **Available Creative** panel. For more information, see “Add a Creative” on page 59.
3. To add an existing flight creative, select a flight creative from the **Available Creative** table and click **+ Add Selected Creative**.
4. To edit a flight creative, select a flight creative from the **Flight Creative** table and click . The **Edit Flight Creative** window appears.

   Note: To edit multiple flight creatives, click the check box for each flight creative that you want to edit, and then click **Edit**.

5. Make any desired edits, and then click **Save Changes**.

   The **Tag qualifier** field determines which qualified tags are temporarily switched into the evaluation context while evaluating that creative for serving.

6. To view the passback URL for a flight creative from the **Flight Creative** table, click in the row for the creative. For more information, see “About Passback URLs” on page 38.

   Note: The displayed URL includes the targeting for only the ad server domain, the HSERVER tag, and the **`/saspb/fcid={fcid value}`** string. The URL does not include the targeting for the flight.

7. To remove a flight creative from the **Flight Creative** table, click in the row for the creative, and then click **OK**.

Copy a Creative

1. On the top navigation bar, click the **Creative** tab.
2. In the row for a creative, click . The **Edit Creative** *creative name* page appears.
3. Enter unique information in the **Name** field, and then make any desired edits and selections.
4. Click **Update Creative**, or click the arrow and select **Update Creative & Copy** or **Update Creative & Make Many Copies**.

Delete a Creative

1. On the top navigation bar, click the **Creative** tab.
2. In the row for a creative, click and then click **OK**.

Chapter 11

Managing Multiple Creative

About Multiple Creatives

Using the multiple creatives tool, you can create and upload many creatives at once. These creatives can be assets that are already saved, new files that will be uploaded, or media files from a specific URL.

Create Multiple Creatives

1. On the top navigation bar, click the **Creative** tab.
2. Click the **Multiple Creative** subtab.
3. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the list. The advertiser that you select is the one that you use when creating the campaign.

   Note: Enter `%` if you want to display the full list of advertisers.
4. In the **Click URL** field, enter the URL that you want users to be redirected to when they click on the creatives.
5. From the **Creative Format** list, select the format of the creative you want to create.
6. From **Image File**, select **URL**, **Asset**, or **Upload** as the source of the media file.
7. From the **Size** list, select an ad size.
8. To upload a file, click **Choose File**, navigate to the creative file, and then click **Open**.

   The filename for the selected creative appears in the**File**list, but the file is not uploaded until you click **Add Media**.
9. In the **Alt Text** field, enter the alternative text that will be displayed if the image cannot be shown in the user's browser.

10. For each creative that you would like to add, repeat steps 5–9. Then click **Add Media**.
11. Click **Add Creative**. If the upload is successful, a green success bar appears.

Chapter 12

Managing Smart Upload

About Smart Upload

The **Smart Upload** tool provides an automatic and easy way of setting up third-party creatives. With this tool, third-party tags are uploaded to the application, the third party is identified, each creative is detected and parsed, and any formatting or URL tokens needed for the creatives to function correctly are automatically inserted.

Note: You can upload only creatives from DoubleClick (iframe/javascript), Atlas (iframe/javascript), MediaMind (iframe/javascript), Facilitate Digital (javascript), or HTML5 creatives in a ZIP file. Files encoded in UTF-8 with a byte-order mark (BOM) are also supported.

Add a Smart Upload

1. On the top navigation bar, click the **Creative** tab.
2. Click the **Smart Upload** subtab.
3. To keep the original filename for the creative, select **Preserve names**.
4. Enter a name in the **Preserved Creative Name** field. The name can be a regular expression or a string, and is used to generate the name of each creative from the uploaded file.

   Note: To use stored regular expressions or strings for the preserved creative name that you created, click **Choose Pattern**. Patterns are created and saved on the **Name Patterns** subtab on the **Creative** tab. For more information, see "About Name Patterns" on page 67.
5. Enter information in the **Constant Creative Name** field, and then select **Pre-pend** or **Append**.

   Note: You can enter only letters, numbers, hyphens, and underscores for the creative name.

You use the **Constant Creative Name** field to control a naming scheme for the creatives identified in the uploaded tag. For each creative, an underscore and a sequence number is added either before or after the constant creative name, depending on whether you select **Pre-pend** or **Append**. For example, if you entered `Demo_Creative` in the **Constant Creative Name** field, the constant creative name would become `0_Demo_Creative` if you selected **Pre-pend** or `Demo_Creative_0` if you selected **Append**.

6. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the autosearch results.

   Note: Enter `%` if you want to display the full list of advertisers.

7. Click **Choose File**, navigate to the third-party creative file, and then select **Open**.

   Note: Only files encoded in UTF-8 are supported.

8. Click **Upload**. Each found creative is displayed.

   Note: If the tag that you uploaded is not recognized, an error message is displayed indicating either that the third party is not recognized, or that the tag formatting is not recognized. In these cases, you must use the **Creative** tab to upload the creatives. For more information, see "Add a Creative" on page 59.

Chapter 13

Managing Name Patterns

About Name Patterns

Name patterns are common regular expressions or strings that are used to generate the name of each uploaded creative. They can be saved, edited, or reused. The Smart Upload tool can use name patterns when it uploads creatives from third-party vendors. For more information, see "About Smart Upload" on page 65.

Add a Name Pattern

To create a name pattern for a creative:

1. On the top navigation bar, click the **Creative** tab.
2. Click the **Name Patterns** subtab.
3. Click **+ Add New Creative Name Pattern**.
4. Enter information in the **Name** field.

   Note: You can enter only letters, numbers, dashes, dots, and underscores.
5. In the **Pattern** field, enter a regular expression or a string that will be used to generate the name of each creative from the file that is uploaded from third-party vendors.
6. Click **Add Creative Name Pattern**.

Edit a Name Pattern

To edit a name pattern for a creative:

1. On the top navigation bar, click the **Creative** tab.
2. Click the **Name Patterns** subtab.
3. Click the name pattern name or click  in the row for the creative. The Edit Name Pattern page appears.
4. Modify the information in the **Name** field.

   Note: You can enter only letters, numbers, dashes, dots, and underscores.
5. In the **Pattern** field, modify the regular expression or the literal string used to generate the name of each creative from file that is uploaded from third-party vendors.
6. Click **Update**.

Delete a Name Pattern

To edit a name pattern for a creative:

1. On the top navigation bar, click the **Creative** tab.
2. Click the **Name Patterns** subtab.
3. In the row for a name pattern, click  and then click **OK**.

Chapter 14

Managing Creative Formats

About Creative Formats

Creative formats provide a convenient way of formatting the look and function of creatives that are delivered through SAS Intelligent Advertising for Publishers. The creative formats are versatile because everything from a text link to a custom XML file for a video player can be set up in a format. Creative formats also help set up third-party creatives. Creative formats are based on two main elements: the format template (the code that is served to the visitor when a creative using this format is selected for delivery), and the accompanying format fields (which contain user-supplied data that is specific to each creative).

SAS Intelligent Advertising for Publishers provides the following hosted content formats, which can be used with almost all non-third party creatives:

Animated GIF
: This format is used with animated GIFs. There are no custom format fields required. When visitors click on the image, the link is opened in the same window.

Animated GIF Opening In New Window
: This format is used with animated GIFs. There are no custom format fields required. When visitors click on the image, the link is opened in a new window.

Flash
: This format should be used with all Flash creatives (SWF files). The minimum version required to view the file must be entered in the **Minimum version** field. For example, if Flash 9.0 was required to view the file, you must enter *9* or *9.0*.

Generic HTML
: This format is used when you specify that your own HTML file is to be used for the creative instead of an existing template.

  Note: This template can be used for both third-party and non-third-party creatives.

HTML5
: This format is used when you specify the format that you want to use when uploading HTML5 ads.

  Note: The default format is an HTML5_iframe system format.

SAS Intelligent Advertising for Publishers supports the following third-party formats for creatives:

Standard-Atlas_iframe
: You use the Atlas Iframe format with third-party Atlas Rich Media tags. Use this template if the Atlas tags contain iframes or use the iview call. This template requires the **Atlas unique identifier** and **Atlas unique code** fields. The unique identifier is the three-character code before /iview/, and the unique code is the string after /iview/.

Standard-Atlas_javascript
: You use the Atlas JavaScript format with third-party Atlas Rich Media tags. Use this template if the Atlas tags contain JavaScript or use the jview call. This template requires the **Atlas unique identifier** and **Atlas unique code** fields. The unique identifier is the three-character code before /iview/, and the unique code is the string after /iview/.

Standard-Doubleclick_iframe
: You use this format with DoubleClick tags that use an iframe or /adi/ call. Do not use this format if publisher or placement IDs are used in the tag. This template requires the **Dfa flight tags** and **Dfa size** custom format fields. The flight tag is the string after /adi/. The size is the string that contains width= and height=.

Standard-Doubleclick_iframe_2
: You use this format with DoubleClick tags that use an iframe or adi call, and tags that contain publisher and placement IDs. This template requires the **Dfa advertiser**, **Publisher id**, and **Placement** custom format fields. Dfa advertiser is the string after /adi/. The publisher ID follows publisher= and the placement ID follows placement=.

Standard-Doubleclick_javascript
: You use this format with DoubleClick tags that use JavaScript or adj call. Do not use this format if publisher or placement IDs are used in the tag. This template requires the **Dfa flight tags** and **Dfa size** custom format fields. The flight tag is the string after /adj/. The size is the string that contains width= and height=.

Standard-Doubleclick_javascript_2
: You use this format with DoubleClick tags that use JavaScript or adj call, and tags that contain publisher and placement IDs. This template requires the **Dfa advertiser**, **Publisher id**, and **Placement** custom format fields. The publisher ID follows publisher= and the placement ID follows placement=.

Standard-Eyeblaster_javascript
: You use this format with DG MediaMind (formerly known as Eyeblaster) tags. This format does not include support for the DG MediaMind iframe buster. The template requires only the **Script src url** custom field. Include the entire URL after script src=, but should exclude the parameters &ord= and &ncu= and the values that precede them. Enter the entire URL into this field from the tag, excluding &ncu=.

Sticky _Format
: You use this format to enable a set of creatives to persist, rather than just one being served. This enables any of the creatives in the set to serve each time, rather than always favoring the first creative that served.

All system creative formats have a viewability feature that counts how many times a creative has been viewed on a visitor's browser or device. To start logging viewability counts, contact Technical Support.

About Frequency Capping

Frequency capping, the ability to limit the exposure of creatives over a specified time span, can be applied to a creative format and at the campaign, flight, and creative levels. Capping can also be applied to impressions, clicks, and conversions. Frequency capping is strictly evaluated. That is, if the value is reached at any level (creative format, campaign, flight, or creative levels), SAS Intelligent Advertising for Publishers does not consider the object, and any hierarchical dependents, for delivery. For example, if the cap has been met at the campaign level, any flights in that campaign is not considered for delivery. Frequency capping for clicks or actions results in SAS Intelligent Advertising for Publishers not serving the creative associated with the click or action until the visitor is qualified to see it again under the frequency capping policy.

Add a Creative Format

1. On the top navigation bar, click the **Creative** tab.
2. Click the **Creative Formats** subtab.
3. Click **+ Add New Creative Format**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), underscores, and hyphens.
5. Enter information in the **Description** field.
6. In the **Target** field, enter the first few letters of the target's name and then select the target from the list. Adding the creative to a flight automatically applies the target to the flight creative.
7. If you do not want snippets added to the creative format, select **Skip snippets**. For more information about snippets, go to "About Snippets" on page 117.
8. To set options that affect the frequency limits for impressions, views, clicks, or conversions in the **Frequency Caps** panel:
   - Enter a value in the **Quantity** field in the Frequency Caps table to specify the number of impressions, views, clicks, or conversions to serve to a visitor. For more information, see "About Frequency Capping" on page 71.
   - Enter a value in the **Period** field in the Frequency Caps table to specify the period of time that must elapse after the quantity is reached for the impressions, views, clicks, or conversions before a visitor is eligible to receive the flight again. For example, if you enter `3` in the **Quantity** field and `1 week` in the

Period field, the flight is shown three times to a visitor and then is not shown again to the visitor until after the one-week waiting period.

Note: You must enter an integer value in the **Period** field. For example, enter **`1 hour`** instead of **`hour`**. A **`per day`** period indicates a calendar day that runs from Midnight to Midnight instead of a 24-hour time span. A month indicates 30 days instead of the actual number of days of a specific month.

- Enter a value in the **Over Lifetime** field in the Frequency Caps table to specify the number of impressions, views, clicks, or conversions to serve to a visitor over the lifetime of the flight.
- Enter a value in the **Per Session** field in the Frequency Caps tables to specify the number of impressions, views, clicks, or conversions to serve to a visitor over the course of a visitor session. A visitor session is initiated whenever an ad request is made, and then, by default, persists for 30 minutes after the visitor's last request.

9. In the **Format Fields** section, to add a field, enter information in the **Name** field and the optional **Description** field. Select **Media** or **Required**, if desired, and then click **+ Add Field**. These items can be reordered by dragging and dropping them.
10. In the **Format Template** section, select one of the following template sources:
    - **Upload**
    - **Text**
    - **Network**
11. If you selected the **Upload** template source, do the following:
    a. From the **MIME Type** list, select **application_javascript (application/javascript)**, **application_xml (application/xml)**, **JS (application/javascript)**, **text_html (text/html)**, **text_javascript (text/javascript)**, or **xml (application/xml)**.

       Note: You can create your own MIME types. For more information, see "Add a MIME Type" on page 77.

    b. Click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
    c. Continue with step 15.
12. If you selected the **Text** template source, do the following:
    a. From the **MIME Type** list, select **application_javascript (application/javascript)**, **application_xml (application/xml)**, **JS (application/javascript)**, **text_html (text/html)**, **text_javascript (text/javascript)**, or **xml (application/xml)**.
    b. Enter information in the **Text** box. The Text is the code that specifies how the creative is going to be displayed and how it functions.
    c. Continue with step 15.
13. If you selected the **Network** template source, do the following:
    a. From the **MIME Type** list, select **application_javascript (application/javascript)**, **application_xml (application/xml)**, **JS (application/javascript)**, **text_html (text/html)**, **text_javascript (text/javascript)**, or **xml (application/xml)**.
    b. From the **Request Method** list, select **GET** or **POST**.
    c. Enter the network URL in the **Request URL** field.

d. Enter information in the **Request Headers** field.

e. From the **Response Type** list, select **HTML**, **JSON**, or **XML**. The selection determines what is produced for the creative that is sent back to the visitor's browser (in response to the network request). When the network directly produces HTML or JavaScript content for the creative, the response template should be left empty. In this case, no processing is done on the network's response, and it is sent as is to the visitor's browser. When the network produces an XML or JSON response from metadata about the creative, the response template must be defined to construct the final creative content from the data provided by the network.

f. Enter information in the **Response Template** field. The response template is similar to a standard creative template because it contains text that defines the HTML or JavaScript content of the resulting ad. Tokens can be referenced in the same way. All of the tokens that are available to standard creative templates can be used here as well. In addition, a special NC token is used to reference selected content from the network's XML or JSON response.

g. To add a token pattern, click **+ Add Token Pattern**, select a tag from the list, and enter information in the **Token** field. Repeat this step for each token pattern that you want to add.

   Tokens that reference tags that specify multiple values (for example, **`/SITE=A,B,C`**) can reference a specific value. The syntax for these tokens is **`%%TOKEN:+1%%`** or **`%%TOKEN:-1%%`** (the plus and minus signs are required). Starting from **`0`**, the specified number determines which value in the comma-separated list is returned. Negative numbers reference the values that are relative to the end of the list. For example, **`+1`** returns the second value (or **`B`** in this example), and **`-1`** returns the last value (or **`C`**).

   Note: To specify a default value in the token that is used when no other value is provided, enter **`%%TOKEN:?"`*`default value`*`"%%`**. You can also set default values (for example, enter **`%%TOKEN:?"%%OTHER:?"`*`default for the default`*`"%%"%%`**).

14. Click **Add Creative Format**.

View a Creative Format

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Creative Formats** subtab.
3. Click  in the row for the creative.

Edit a Creative Format

1. On the top navigation bar, click the **Creative** tab.
2. Click the **Creative Formats** subtab.

3. Click the creative format name or click in the row for the creative. The Edit *creative format name* Creative Format page appears.
4. Edit the **Name**, **Description**, and **Target** fields as necessary.
5. If you do not want snippets added to the creative format, select **Skip snippets**. For more information about snippets, go to "About Snippets" on page 117.
6. To set options that affect the frequency limits, expand the **Frequency Caps** panel, and then do the following:
   - Enter a value in the **Quantity** field in the Frequency Caps table to specify the number of impressions, clicks, or conversions to serve to a visitor.
   - Enter a value in the **Period** field in the Frequency Caps table to specify the period of time that must elapse after the quantity is reached for the impressions, clicks, or conversions before a visitor is eligible to receive the flight again. For example, if you enter `3` in the **Quantity** field and `1 week` in the **Period** field, the flight is shown three times to a visitor and then is not shown again to the visitor until after the one-week waiting period.
   - Enter a value in the **Over Lifetime** field in the Frequency Caps table to specify the number of impressions, clicks, or conversions that can serve to a visitor over the lifetime of the flight.
   - Enter a value in the **Per Session** field in the Frequency Caps tables to specify the number of conversions that can serve to a visitor over the course of a visitor session. A visitor session is initiated whenever an ad request is made, and by default persists for 30 minutes after the visitor's last request.
7. To add a field, click **+ Add Field**, enter information in the **Name** field and the optional **Description** field, and then select **Media** or **Required**, if desired. Repeat this step for each field that you want to add.
8. In the **Format Template** field, select one of the following template sources:
   - **Upload**
   - **Text**
   - **Network**
9. If you selected the **Upload** template source, do the following:
   a. From the **MIME Type** list, select **application_xml (application/xml)**, **JS (application/javascript)**, or **xml (application/xml)**.
   b. Click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
   c. Continue with step 13.
10. If you selected the **Text** template source, do the following:
   a. From the **MIME Type** list, select **application_xml (application/xml)**, **JS (application/javascript)**, or **xml (application/xml)**.
   b. Enter information in the **Text** box.
   c. Continue with step 13.
11. If you selected the **Network** template source, do the following:
   a. From the **MIME Type** list, select **application_xml (application/xml)**, **JS (application/javascript)**, or **xml (application/xml)**.
   b. From the **Request Method** list, select **GET** or **POST**.

c. Enter the network URL in the **Request URL** field.

d. Enter information in the **Request Headers** field.

e. From the **Response Type** list, select **HTML**, **JSON**, or **XML**.

f. Enter information in the **Response Template** field.

g. To add a token pattern, click **+ Add Token Pattern**, select a tag from the list, and enter information in the **Token** field. Repeat this step for each token pattern that you want to add.

Tokens that reference tags that specify multiple values (for example, **`/SITE=A,B,C`**) can reference a specific value. The syntax for these tokens is **`%%TOKEN:+1%%`** or **`%%TOKEN:-1%%`** (the plus and minus signs are required). Starting from `0`, the specified number determines which value in the comma-separated list is returned. Negative numbers reference the values that are relative to the end of the list. For example, **`+1`** returns the second value (or **`B`** in this example), and **`-1`** returns the last value (or **`C`**).

Note: To specify a default value in the token that is used when no other value is provided, enter **`%%TOKEN:?"`*`default value`*`"%%`**. You can also set default values (for example, enter **`%%TOKEN:?"%%OTHER:?"`*`default for the default`*`"%%"%%`**).

12. Click **Update**.

Copy a Creative Format

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Creative Formats** subtab.
3. In the row for a creative, click . The Edit *creative format name* Creative Format page appears.
4. Enter unique information in the **Name** field, and make desired edits and selections. For more information, see “Edit a Creative Format” on page 73.
5. Click **Update**.

Delete a Creative Format

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **Creative Formats** subtab.
3. In the row for a flight, click and then click **OK**.

Chapter 15

Managing MIME Types

About MIME Types

MIME types enable you to specify HTTP content types when a creative is delivered.

Add a MIME Type

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **MIME Types** subtab.
3. Click + **Add New MIME Type**.
4. Enter information in the **Name** field and the **Content type** fields.
5. Click **Add MIME Type**.

Edit a MIME Type

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **MIME Types** subtab.
3. Click the MIME type name or click in the row for the MIME type. The Edit *MIME type name* MIME Type page appears.
4. Enter information in the **Name** field and the **Content type** fields.
5. Click **Update**.

Delete a MIME Type

1. On the top navigation bar, click the **Traffic** tab.
2. Click the **MIME Types** subtab.
3. In the row for a MIME type, click  and then click **OK**.

Part 4

Ad Targeting

Chapter 16

Managing Targets

About Targets

Targets provide a powerful and convenient way of identifying a specific demographic using sites, areas, custom tags, geotargeting, keywords, and visitor browser information. Targets can be applied to flights, tiers, products, and creatives. Targeting is one way to control who gets to see what ad, and where. Targeting enables you to deliver ads to specific demographics and page locations. You can group key=value pairs to create targets that represent audience and content segments.

You can specify a target rule (targeting operators "Is empty" and "Is not empty") for either the presence or lack of a tag in an ad request. The operator is selectable only for tags (any type, including integer and decimal), user agent, referrer, and keyword.

About Targetable Values

Targets consist of left values, operators, and right values. Left values are the elements being targeted, like sites, areas, tags, and so on. The operator specifies how it is being targeted (for example, equals, not equal to, like, and so on). The right value is the actual value of the element selected (for example, New York City would be a right value if City was the left value). Every tag defined in the system can be targeted. When creating a target, you can also use the following predefined Left Values:

Accepts Cookies

This value can be used to identify visitors that do or do not accept cookies. This is useful when a particular type of creative might require cookie support. It is also part of a set of tools for click fraud bot detection.

Area

All AREAs that are set up in the system can be targeted.

Day of Week

The specific day of the week can be targeted. This uses the day of week on the server, not the visitor.

Device Values

The following tags are based on device values found for the visitor:

- dev_type
- dev_os
- dev_browser_name
- dev_display_width
- dev_display_height
- dev_flash_capable
- dev_html_audio
- dev_html_video

Domain

The host name value of the visitor requesting the ad. For example, a target containing "Domain Like %.edu" would target all visitors with a host name ending with ".edu".

Event

A specific event with a count, a duration, and a relative time window. Examples and descriptions follow:

Note: To compare events, select an event, click , click **Select another event**, and then enter the first few letters of an event name in the **Events** field and then select the event from the autosearch results.

LVal="EVENT.MYEVENTNAME" Oper="Less Than" RVal="3,0,300"

match if the visitor has experienced fewer than three occurrences of the event MYEVENTNAME between 0 and 300 seconds ago (that is, in the last 300 seconds)

Rval="3,86400,172800"

between 86400 and 172800 seconds ago (that is between 24 and 48 hours ago)

Rval="3,2015-09-01,2015-09-30"

between 00:00:00 on 2015-09-01 and 23:59:59 on 2015-09-30

Rval="3,2015-09-01" (missing third parameter)

on or after 00:00:00 on 2015-09-01

Rval="3,,2015-09-30" (missing second parameter)

on or before 23:59:59 on 2015-09-30

Rval="3" (missing second and third parameter)

ever (that is, visitor has < 3 events ever)

Geotargeting Values

The following tags are based on geotargeting values found for the visitor:

- City
- Conn Speed
- Country
- ISP name
- Metro Code
- Proxy description
- Proxy type
- Region
- SIC code
- ZIP code text

Geotargeting is based on the evaluation of the request IP. The application compares the IP to a data file obtained from a vendor. The vendor is responsible for updating the file. The application receives an updated file from vendors every week. There is an error rate of around 3%. Device values are also based on data received from vendors. The application has a specialized device detection module that a vendor keeps up-to-date.

IP Address
: The IP address of the visitor can be targeted using this value.

Keyword
: Multiple keywords can be specified in an ad call, and these keywords can be targeted.

Recurrence
: Specify a start and end time, or duration for this target. Also, set a weekly or monthly frequency for the target, as well as the specific pattern for that recurrence.

Referrer URL
: This targets the referring URL value returned by the visitor's browser. For example, if the ad call resided on http://www.aimatch.com, then this URL would be reported as the referrer URL by the browser. Note that not all browsers report this value correctly, and ad calls in IFRAMEs might not report correctly either.

Site
: All sites that are set up in the system can be targeted

Target
: Other targets that are set up in the system can be referenced in other targets.

Time
: A target can include a specific time or range of times. Note that this uses the time of the server, not the visitor. The values specified do not roll over to the next day. In other words, specifying **Time Between 23:00 and 01:00** does not target 11 p.m. to 1 a.m. the next day. Likewise, targeting **Time Before 02:00** does not include anytime prior to midnight.

User Agent
: This is the user agent value returned by the visitor's browser. This can be used to target to specific browsers, operating systems, and or a combination of browsers and operating systems.

Add a Target

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Targets** subtab.
3. Click **+ Add New Target**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.?;:). You cannot enter double quotation marks or commas.
5. Enter information in the **Description** field.
6. Select **Match All**, **Match Any**, **Don't Match All**, or **Don't Match Any**.

   Match Any is the equivalent of using the Boolean OR operator. If any of the statements are true, the target matches. Match Any requires that any of the targeting elements can be matched (meaning, the ad request contains a left value and right value that is also in the target), and the flight are eligible to serve to that request.**Match All** is the equivalent of using the Boolean AND operator. All statements must be true for the target to match. Match All requires that all of the targeting elements in the target must match what is specified in the ad call.

   Don't Match All is the equivalent of using both the Boolean OR and NOT operators. **Don't Match Any** is the equivalent of using both the Boolean AND and NOT operators. Don't Match Any and Don't Match All are targeting operators that exclude groups of targeting statements. These operators can be used when creating new targets or creating rules for a flight or line item. For example, if you want to any of several target site values but want to exclude a specific target, select **Don't Match All** with those targets specified. The target can be excluded without creating a new negative target or statement.

   The Don't Match Any operator is the negation of the Match Any operator. It evaluates as "false" when one of the targeting statements is true on an incoming ad request. The Don't Match All operator is the negation of the Match All operator. It evaluates as "false" when all of the targeting statements are true on an incoming ad request.
7. Select a grouping rule or a left value. If you select a grouping rule, select a left value and continue with the next step.
8. Select an operator.

   Note: Some operators are unavailable for the selected left value.
9. Depending on the left value that you selected, enter a value or select from search results in the field.

   Note: You can enter `%` to specify a wildcard of any length or an underscore to specify a wildcard of a single character.
10. If you want to add another grouping rule or left value, click the green plus sign icon.
11. Click **Add Target**.

View a Target

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Targets** subtab.
3. Click in the row for the target. The target name page appears.
4. To view the history of edits to the target, select **View History**.
5. To remove an item associated with the target, click in the row for the association and then click **OK**.

Edit a Target

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Targets** subtab.
3. Click the target name or click in the row for the target. The Update Target page appears.
4. Make desired edits in the **Name** and **Description** fields.

   For information about editing an event, see "Edit an Event in a Target" on page 85.
5. Modify the targeting values as desired. For some left values, you can click **click to edit** to edit the fields. Once the desired values have been entered, click **Save & Close**.
6. Click **Update**.

Edit an Event in a Target

1. Click next to an event. The Event Occurrences and Time Periods window appears.
2. Enter a value in the **Occurrences** field.
3. Select **Relative Dates**, **Absolute Dates**, **Any Dates**, or **Session**.

   If you selected **Relative Dates**, enter values in **This event start** and **This event end** fields. The values represent seconds.

   If you selected **Absolute Dates**, enter time and date values (or click the calendar icon to select the values) in the **Start Date** and **End Date** fields.

Copy a Target

Note: You can create an original flight only in a campaign. For more information, see “Add a Flight to a Campaign” on page 30.

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Targets** subtab.
3. In the row for a target, click . The Update Target page appears.
4. To make edits to the target, follow the instructions in “Edit a Target” on page 85.

Delete a Target

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Targets** subtab.
3. In the row for a target, click  and then click **OK**.

Chapter 17
Managing Tags

About Tags

You can create a custom set of tags to precisely target advertisement positions or to pass additional data into SAS Intelligent Advertising for Publishers that is made available for targeting. Tags are key-value pairs that are used for targeting and to map audience and content segments. Taxonomies are the basic building blocks of your audience, product, and data scheme. Key-value pairs represent the data that is passed to SAS Intelligent Advertising for Publishers. This data influences real-time ad selection, can help refine your concept of audience, and improve product definitions.

Tags can be strings, integers, decimals, or dates.

Integer tags contain only whole numbers as tag values. These tags enable you to easily implement comparison operators such as greater than and less than to targeting. All possible tag values (integers) are automatically populated when an integer tag is created.

Add a Tag

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Tags** subtab.
3. Click **+ Add New Tag**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9) and underscores. Tags must not start with a numeric value.

5. Select a **Type** (String, Integer, Decimal, or Date). Integer and Decimal tags do not use explicitly defined values. Instead, any numeric value with a decimal can be specified in an ad request and can be defined in targeting. Integer and Decimal tags

should not be used with dynamic inventory adjustments. The Date tag enables you to target any date. You do not need to manually create the date values.

6. (Optional) Select a **Tag Group**. For information about tag groups, see Chapter 18, “Managing Tag Groups,” on page 91.
7. To add the tag to the **Dynamic target** list in the Edit Product window, select **Dynamic target**.

   Dynamic targeting is available for instances where targeting is complex or granular (for example, when targeting to large groups of ZIP codes). Only the tag itself, and no values, are selected for the product.
8. To enable logging for the tag in the Tags Report, select **Tagsum logging**. For more information, see “Tags” on page 189.
9. Click **Add Tag**.
10. To add values to the tag, enter them in the **Values** field, and then click **Add Values**.

    Note: You can enter only letters, numbers, hyphens, dots, and underscores. You can use commas to delimit multiple values. To create a display name for a value, enter the display name for the tag value after a pipe character (|) (for example, *tag_value|display_name*). You can use any other special character in the value name. You can enter up to 100 values. You should use the XML API if you need to add a very large set of values. For more information, see Appendix 1, “XML API Examples,” on page 195.
11. To specify the sites that use the tag, click , select **Specific Site(s)**, select one or more sites from the list, and then click **Save Sites**.

Edit a Tag

1. On the top navigation bar, click the **Targeting** subtab.
2. Click the **Tags** tab.
3. Click the tag name or click in the row for the tag. The Edit Tag page appears.
4. Enter information in the **Name** field.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores. The information must start with a letter.
5. To set the tag for a dynamic target, select **Dynamic target**.
6. To enable tagsum logging, select **Tagsum logging**.
7. Click **Update**.
8. To add values to the tag, enter them in the **Values** field, and then click **Add Values**.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores. You can enter up to 100 values. It is recommended to use XML API if you need to add a very large set of values.
9. To specify the sites that use the tag, click , select **Specific Site(s)**, select one or more sites from the list, and then click **Save Sites**.

Delete a Tag

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Tags** subtab.
3. In the row for a tag, click  and then click **OK**.

Chapter 18

Managing Tag Groups

About Tag Groups

Tag groups offer a way to group a set of individual tags. For example, the cars tag group can include the tags for "make," "model," and "color."

Note: Each tag can be part of only one tag group.

For information about tags, go to "About Tags" on page 87.

Add a Tag Group

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Tag Groups** subtab.
3. Click + **Add New Tag Group**.
4. Enter information in the **Name** field.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores. The tag group name must start with a letter.
5. (Optional) Enter information in the **Tags** field.
6. (Optional) Enter information in the **Description** field.
7. Click **Add Tag Group**.

Edit a Tag Group

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Tag Groups** subtab.
3. Click the tag group name or click  in the row for the tag. The Edit Tag Group page appears.
4. (Optional) Modify the information in the **Name** field to rename the tag group.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores. The tag group name must start with a letter.
5. (Optional) Modify information in the **Tags** and **Description** fields.
6. Click **Update**.

Delete a Tag Group

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Tag Groups** subtab.
3. In the row for a tag group, click  and then click **OK**.

Chapter 19

Managing Sites

About Sites

Sites are part of the building blocks of targets. Targets offer a powerful and convenient way of targeting a specific demographic consisting of sites, areas, custom tags, geotargeting, keywords, and visitor browser information. All sites set up in the system can be targeted.

Add a Site

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Sites** subtab.
3. Click **+ Add New Site**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), periods, underscores, and hyphens.
5. Enter information in the **Description** field.
6. To record and report on a revenue share for the campaigns that you have entered, enter a value in the **Revenue cut** field. A revenue cut is usually used when you represent sites that you do not own.
7. Click **Add Site**.

Add Multiple Sites

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Sites** subtab.
3. Click **+ Add Multiple Sites**.
4. Enter information in the **Revenue cut** and **Names** fields.
5. Click **Add Sites**.

Edit a Site

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Sites** subtab.
3. Click the site name or click in the row for the site. The Edit *site name* Site page appears.
4. Enter information in the **Name** field.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores.
5. Enter information in the **Description** field.
6. Enter a value in the **Revenue cut** field..
7. Click **Update**.

Delete a Site

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Sites** subtab.
3. In the row for a site, click and then click **OK**.

Chapter 20

Managing Areas

About Areas

Areas are part of the building blocks of targets. Targets offer a powerful and convenient way of targeting a specific demographic consisting of sites, areas, custom tags, geotargeting, keywords, and visitor browser information. All areas set up in the system can be targeted.

Add an Area

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Areas** subtab.
3. Click **+ Add New Area**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), periods, underscores, and hyphens.
5. Enter information in the **Sites** field.
6. Click **Add Area**.

Add Multiple Areas

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Areas** subtab.
3. Click + **Add Multiple Areas**.
4. Enter information in the **Names** fields.
5. Click **Add Areas**.

Edit an Area

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Areas** subtab.
3. Click the area name or click in the row for the area. The Edit *area name* Area page appears.
4. Enter information in the **Name** field.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores
5. Enter information in the **Sites** field.
6. Click **Update**.

Delete an Area

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Areas** subtab.
3. In the row for an area, click and then click **OK**.

Chapter 21

Managing Sizes

About Sizes

Sizes are part of the building blocks of targets. Targets offer a powerful and convenient way of targeting a specific demographic consisting of sites, areas, sizes, custom tags, geotargeting, keywords, and visitor browser information. There are many default sizes included in the system, but you can define more sizes using the Targeting and Sizes tabs.

Add a Size

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Sizes** subtab.
3. Click **+ Add New Size**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), periods, underscores, and hyphens.
5. Enter values in the **Width** and **Height** fields.
6. Click **Add Size**.

Edit a Size

1. On the top navigation bar, click the **Targeting** tab.

2. Click the **Sizes** subtab.
3. Click the size name or click  in the row for the size. The Edit *size name* Size page appears.
4. Make desired edits in the **Name**, **Width**, and **Height** fields.

   Note: You can enter only letters, numbers, hyphens, dots, and underscores in the **Name** field.
5. Click **Update**.

Delete a Size

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Sizes** subtab.
3. In the row for a size, click  and then click **OK**.

Chapter 22

Managing Supertags

About Supertags

Supertags are special tags that can be included in an ad call and expanded into one of any number of arbitrary collections of other tags and values. Supertags can be used to ease a customer's migration from other ad servers, and provides a level of tag management to ad operations. Supertags provide a way of mapping a value to an arbitrary path information snippet. You can filter existing supertags by value and path information.

Any number of supertag values can be created to predefine multiple sets of tags and values. When an ad call includes "SUPERTAG=supertagvalue", the tags in the corresponding pathinfo are added to the ad request internally. The "SUPERTAG" tag name itself is not configurable. Multiple values can be supplied for SUPERTAG if they are separated by commas, which is similar to other tags. The resulting substitution yields the union of all the values' definitions, and later definitions take precedence. An ad call that includes SUPERTAG can freely include other tags as well, and the complete set of tags implied by that mix determines what ad is served. When a SUPERTAG value maps to a path that includes a tag that is also explicitly included in the ad call, the explicit value from the ad call takes precedence. Like other tag values, supertag values are case insensitive. Unlike other tag values, there are no restrictions on use of punctuation, and special characters need only to be URL-encoded appropriately. These relaxed restrictions maximize compatibility when supertags are used to help with migration from other ad servers.

When a supertag value has an embedded dot (period, "."), the string in front of the dot defines a namespace for the value. Different namespaces can define the same value.

Add a Supertag

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Supertags** subtab.
3. Enter the name for the supertag in the **Value** field.
4. Enter tag and value pairs in the **Path Info** field. These values are used in conjunction with the path info from an ad request during flight evaluation.
5. (Optional) In **Target**, enter a target that, when matched, automatically adds the supertag path info to a visitor's ad request path info for flight evaluation.

   Note: To display the full list of targets, enter *%*.
6. Click **Save Supertag**.

Filter Supertags

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Supertags** subtab.
3. Expand the **Filtering** section.
4. Enter information in the **Value Like** field.
5. Enter information in the **Path Info Like** field.
6. Click **Filter**.

Delete a Supertag

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Supertags** subtab.
3. In the row for a supertag, click  and then click **OK**.

Chapter 23

Managing Placements

About Placements

Placements are an extension of the Partners component. Previously, one target applied by a partner could affect the entire integration with that partner. Placements extend this functionality by enabling not only multiple placement entities to be created (which enable many different targeting combinations to be used), but a placement can be configured to use several partners. Different floor prices can be set for each partner by each biddable tier. Once an ad request is received and a flight creative is selected, placements are considered for serving. If there are no bids meeting the configured floor price, the previously selected creative is used instead. If a flight creative cannot be found to match the ad request in the tier being evaluated, then placements do not serve even if they are eligible for the request.

Before placements can be created and configured, partners that are bidding to serve the placements need to be configured by SAS Technical Support. Contact Technical Support for more information about the requirements to set up a partner. Once a partner is enabled, site and size mappings should be configured under the **Partners** tab. Any targets specified for the partner are also applied at the placement level. For more information about partners, see Partners on page 179.

Add a Placement

Specified site values are applied as targeting to the placement, and concatenated with any targets specified as well. It is important to ensure that there are no conflicting statements. For example, if an included target has a rule "Site Equals A" and site "B" that is specified in the **Sites** field, the placement would not serve. It is best practice to exclude any site targeting in targets used for placements. For each unique combination of targeting, a separate placement is needed. For example, if different floor prices are needed for Site=A versus Site=B, then two placements would be necessary.

Any sites or sizes specified need to have a corresponding mapping value setup for the partner in question (if the partner requires it) under the **Partners** tab.

Each partner that bids to the placement should have a placement key that is configured. The placement keys, if required by the partner, are distinct from the site and size mappings under the **Partners** tab. If the partner needed for the placement is not in the list, contact Technical Support.

To add a placement:

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Placements** subtab.
3. Click **+ Add New Placement**.
4. Enter information in the **Name** field.
5. Enter a **Target**.
6. Enter a **Size**. Placement size must be mapped on the partner page or the ads will not serve.
7. Select a **Device** to specify whether to request a video ad, web ad, or application ad.
8. Enter a **Default floor price**. Before floor prices can be configured, a tier needs to be set as biddable in order to be eligible to serve a placement. Once a tier is biddable, placements will be considered for serving when that tier is evaluated for an ad request. Floor prices can then be configured for each tier by partner on the **Floor Prices** tab on the Placements page. The floor price specifies the lowest acceptable bid price from the configured partners. For more information, see "About Tiers" on page 43.
9. (Optional) In the **Parent Placement** field, enter the first few letters of a name for a placement and then select a placement from the autosearch results. If you specify a parent placement, it is sent as a bid request in parallel with the current placement.

   You typically specify parent placements when a placement needs to include different channels (display, video, or app), targets, or floor prices.
10. Click **Add Placement**.

Edit a Placement

To edit a placement:

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Placements** subtab.
3. Click a placement name or click  in the row for the placement. The Edit Placement page appears.
4. On the **Properties** tab, enter information in the **Name** field.
5. Enter a **Target**.
6. Enter a **Size**. Placement size must be mapped on the partner page or ads will not serve.
7. Select a **Device** to specify whether to request a video ad, web ad, or application ad.

8. Enter a **Default floor price**. Before floor prices can be configured, a tier needs to be set as biddable in order to be eligible to serve a placement. Once a tier is biddable, placements are considered for serving when that tier is evaluated for an ad request. Floor prices can then be configured for each tier by partner on the **Floor Prices** tab on the Placements page. The floor price specifies the lowest acceptable bid price from the configured partners.

9. (Optional) In the **Parent Placement** field, enter the first few letters of a name for a placement and then select a placement from the autosearch results. If you specify a parent placement, it is sent as a bid request in parallel with the current placement.

   You typically specify parent placements when a placement needs to include different channels (display, video, or app), targets, or floor prices.

10. Select or delete any associated **Sites**.

11. On the **Network Participation** tab, partners with configured sites are listed.

    Any sites configured for this partner must also have **Site Mappings** configured on the partner edit page or they will not have active placements. Click **Edit partner network settings** to go to that partner site and edit the necessary items.

12. On the **Floor Prices** tab, the name, target and size information is listed.

    Click the ellipsis (...) button beside **Placement Bidding** to set different floor price for each partner included in the placement. If a bid from a partner is lower than what is specified in the placement, the bid is rejected. The default floor price on the **Properties** tab is used initially to populate the floor prices for each partner and tier. If the default is later changed, it does not affect the floor prices specified on the **Floor Prices** tab even if they have not been changed from the default value. For more information, see “About Tiers” on page 43.

13. Click **Update**.

Chapter 24

Managing Events

About Events

Events allow an event name to be associated with one or more timestamps that correspond to a visitor action (such as visiting a certain page). Event occurrences are logged using state vector requests, and can be used to target visitors that have a certain number of event occurrences.

Add an Event

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Events** subtab.
3. Click **+ Add New Event**.
4. Enter information in the **Name** field.
5. Enter a **Max Count**. This is the maximum number of event occurrences that should be tracked per visitor. For example, a flight could be targeted to visitors that have registered five or more occurrences of the event “homeowner”. The Max Count in this case would need to be set to at least five or higher. If a visitor registers more event occurrences than the Max Count setting, their count for the event stays at the maximum number. The default maximum setting is ten.
6. Enter a **Max Age**. This is the maximum age of event occurrences that should be kept. Each event occurrence has a timestamp, and if the timestamp is older than what is specified for this setting, that occurrence is discarded. For example, if a visitor has registered the event “homeowner” five or more times in the past seven days, the event would have a Max Age setting of seven days. The default maximum setting is 365 days.

Note: Spell out the time units: seconds, minutes, hours, days, months, or years. If you do not include a time unit, the default is seconds.

7. (Optional) In the **Target** field, enter a target consisting of rules that, when matched, logs an occurrence of the event for the visitor.

   Note: To display the full list of targets, enter %.

8. Click **Add Event**.

Edit an Event

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Events** subtab.
3. Click an event name or click  in the row for the event. The Edit Event page appears.
4. Enter information in the **Name** field.
5. Enter a **Max Count**. This is the maximum number of event occurrences that should be kept.
6. Enter a **Max Age**. This is the maximum age of event occurrences that should be kept.
7. Click **Update**.

Delete an Event

1. On the top navigation bar, click the **Targeting** tab.
2. Click the **Events** subtab.
3. Click an event name or click  in the row for the event.
4. Click **OK** to confirm.

Part 5

Ad Assets

Chapter 25

Managing Media

About Media

The **Media** tab enables you to upload an asset to the solution.

Add New Media

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Media** tab.
3. Click + **Add New Media**.
4. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the list.

   Note: Enter `%` if you want to display the full list of advertisers.
5. Click **Choose File** to select an asset.
6. Navigate to the file that you want to upload, and then click **Open**.
7. Click **Add Media**.

Edit Media

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Media** tab.

3. In the row for a medium, select a filename or click . The Edit Media page appears.
4. If you want to change the advertiser, enter the first few letters of the advertiser's name in the **Advertiser** field and then select the advertiser from the list.

   Note: Enter `%` if you want to display the full list of advertisers.
5. If you want to use another file, click **Choose File**, navigate to the file that you want to upload, and then click **Open**. To view the existing asset, click the asset link.
6. To edit a creative for the media, click the creative filename in the Name column, and then make desired changes. For more information, see "Edit a Creative" on page 60.
7. Click **Update**.

Delete Media

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Media** tab.
3. In the row for a media file, click  and then click **OK**.

Chapter 26

Managing Advertiser Creative Templates

About Advertiser Creative Templates

The Advertiser Creative Templates tab enables you to upload a file or add text to create a template.

Add an Advertiser Creative Template

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Advertiser Creative Template** subtab.
3. Click + **Template**.
4. In the **Advertiser** field, enter the first few letters of an advertiser's name and then select the advertiser from the list.

   Note: Enter `%` if you want to display the full list of advertisers.
5. In the **Template type** field, select **Upload** or **Text**.
6. If you selected the **Upload** template type, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
7. If you selected the **Text** template type, do the following:
   a. Enter information in the **File Name** field.

      Note: You can enter only letters, numbers, hyphens, and underscores

   b. From the **MIME Type** list, select **application_xml (application/xml)**, **JS (application/javascript)**, or **xml (application/xml)**.

To create an MIME type, follow the instructions in “Add a MIME Type” on page 77.

c. Enter information in the **Template Text** box.

8. Click **Add Advertiser Creative Template**.

Edit an Advertiser Creative Template

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Advertiser Creative Template** subtab.
3. In the row for the advertiser creative template, select the filename or click . The Edit Advertiser Creative Template page appears.
4. To change the advertiser, remove the existing advertiser’s name, enter the first few letters of a new advertiser’s name in the **Advertiser** field and then select the advertiser from the list.
5. In the **Template type** field, select **Upload** or **Text**.
6. If you selected the **Upload** template type, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
7. If you selected the **Text** template type, do the following:
   a. Enter information in the **File Name** field.

      Note: You can enter only letters, numbers, hyphens, and underscores

   b. From the **MIME Type** list, select **application_xml (application/xml)**, **JS (application/javascript)**, or **xml (application/xml)**.

      To create an MIME type, follow the instructions in “Add a MIME Type” on page 77.

   c. Enter information in the **Template Text** box.
8. Click **Update**.

Delete an Advertiser Creative Template

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Advertiser Creative Template** subtab.
3. In the row for the advertiser creative template, click and then click **OK**.

About Predefined Tokens

Tokens are placeholders for data or content that are used in creatives and creative formats. For example, because creatives vary in size, the code needs to reflect the size of each creative. Tokens for width and height represent the values of width and height for a creative. These tokens enable multiple creatives to use a single creative format that always includes the correct width and height.

Token names are in uppercase and begin and end with double percent signs (“%%”). This formatting enables the server to identify tokens and to determine what values to replace them with. For example, if a creative has a width of 728 and a height of 90, a creative format might have the following code: **<img src=”%%MEDIA%%” width=”%%WIDTH%%” height=”%%HEIGHT%%”>** . You can have “ %%” appear in your output if the original template contains “%%%%”. A series of four % characters ("%%%%") results in a single pair of % ("%%") in the resulting token expansion (for example, "Email address %%%%addr%%%%" becomes "Email address %%addr%%" after substitution.

SAS Intelligent Advertising for Publishers provides predefined tokens, which can be placed in both file-based and URL-based creatives. The following table lists and describes the data sources that are used to replace the tokens.

Note: Appending **:URLENCODED** (for example, **%%DATETIME:URLENCODED%%**] will URL-encode the token value. Appending **:URLDECODED** will decode a URL-encoded string.

Token	Description
%%ALTIMAGE%%	The string that was entered in the **Alt Image** field for a creative.
%%ALTTEXT%%	The string that was entered in the **Alt Text** field for a creative.
%%BASEURL%%	URL pointing to the client's ad server (for example,http://crtl.aimatch.com/{*client-shortname*}.
%%BEACONURL%%	The URL added to the creative code returned to the page that instructs SAS Intelligent Advertising for Publishers to increment the impression count. Note: this token requires an "img src=". This token returns only the beacon counting URL itself, and would be used in contexts that call it appropriately, such as in an image tag. *Note:* This token requires an img src= HTML tag to be prepended to the URL.

Token	Description
%%CLICKURL%%	A click URL that points to the SAS Intelligent Advertising for Publishers ad server for click-counting and returns the entered click URL that you entered for the creative to the browser for relocation.
%%CLICKDESTINATIONURL %%	A click URL that returns the destination URL specified in a creative's Click URL field.
%%CRTUINFO%%	The information that was entered in the **User info** field in the Edit Creative window.
%%CUSTOMER%%	The client's short name (the subdomain name for the client's UX).
%%DATE%%	Date value when passed in after the `/date=` tag in an ad call.
%%DATETIME%%	Current date and timestamp value in the format `YYYY-MM-DD HH:MM:SS`.
%%DURATIONS%%	The ad's display duration value in seconds (for example, `60`].
%%DURATIONHMS%%	The ad's display duration value in HH:MM:SS format (for example, `01:30:00`, indicating 1 hour and 30 minutes).
%%FCID%%	The unique FCID (Flight Creative ID) value for a creative that links the creative to a flight.
%%FLIGHTID%%	The unique integer value for a flight in the UX.
%%FLTUINFO%%	The information that was entered in the **User info** field in the **Description** panel of the Edit Flight window.
%%GENERIC%%	The string passed in on an ad call between the string `/generic=` and the next `/`.
%%GEO_CITY%%	The visitor's city.
%%GEO_CONN_SPEED%%	The visitor's connection speed.
%%GEO_COUNTRY%%	The visitor's country.
%%GEO_IP%%	The visitor's IP address.
%%GEO_METRO_CODE%%	The visitor's metro area.
%%GEO_REGION%%	The visitor's region (state or province).

Token	Description
%%GEO_ZIP_CODE_TEXT%%	The visitor's ZIP code.
%%HEIGHT%%	The height assigned to the creative when a size is assigned to it.
%%MEDIA%%	The string that was entered in the **Media** field for a creative.
%%MID%%	The visitor's SAS Intelligent Advertising for Publishers cookie value.
%%PATH%%	The pathinfo string passed in on an ad call between http:// and ? in the URL
%%PRECLICKURL%%	The click URL that points to the SAS Intelligent Advertising for Publishers ad server for click-counting followed by the string **`relocate=`**. This string enables a click URL to be appended so that SAS Intelligent Advertising for Publishers can issue a relocate URL. It is useful for third-party ad servers to append a click destination while enabling SAS Intelligent Advertising for Publishers to count the click.
%%QUERY%%	The query string passed in on an ad call after the ? in the URL.
%%RANDOM%%	The random number passed in on an ad call after the string **`/random=`**.
%%REFERER%%	The URL of the page containing the ad call URL.
%%TIME%%	Time value when passed in after the **`/time=`** tag in an ad call.

If you create a field for a creative template, a corresponding custom token is also created. Custom tokens are prepended with **`X_`** unless you selected the **Media** check box. If you selected the **Media** check box, the token name matches the name of the field that you created. For more information, see "Add a Creative Format" on page 71.

You can create a dynamic token to replace a string in the creative code content. For example, an ad call can pass in a key=value pair such as **`/toys=boardgame`**:in the http://crtl.aimatch.com/customername/hserver/site=toycatalog/size=728x90/toys=boardgame URL. The creative content can then use the token %%TOYS%% to get the associated value "boardgame": **`<a href="%%CLICKURL%%?%%TOYS%%"><img src="%%MEDIA%%" width=728 height=90></a>`** where %%TOYS%% is replaced by boardgame in the query string of the click URL.

Tokens that reference tags that specify multiple values (for example, **`/SITE=A,B,C`**) can reference a specific value. The syntax for these tokens is **`%%TOKEN:+1%%`** or **`%%TOKEN:-1%%`** (the plus and minus signs are required). Starting from **`0`**, the specified number determines which value in the comma-separated list is returned. Negative numbers reference the values that are relative to the end of the list. For example, **`+1`** returns the second value (or **`B`** in this example), and **`-1`** returns the last value (or **`C`**).

Note: To specify a default value in the token that is used when no other value is provided, enter **`%%TOKEN:?"`*`default value`*`"%%`**. You can also set default values (for example, enter **`%%TOKEN:?"%%OTHER:?"`*`default for the default`*`"%%"%%`**).

Chapter 27

Managing Snippets

About Snippets

Code snippets are user-defined sections of code that are appended to ad responses. Typically, these snippets would be third-party beacons but can consist of any text or code. By default, a code snippet delivers with all ad responses (except for TSERVER requests). Snippets can be targeted so that they serve only to responses that match the applied targeting. Code snippets are accessed on the **Snippets** tab.

Add a Snippet

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Snippets** tab.
3. Click + **Snippet**.
4. To activate the snippet, select **Active**.
5. In the **Template type** field, select **Upload** or **Text**.
6. If you selected the **Upload** template type, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
7. If you selected the **Text** template type, follow these steps:
   a. Enter information in the **File Name** field.

      Note: You can enter only letters, numbers, hyphens, and underscores

   b. From the **MIME Type** list, select **application_xml (application/xml)**, **JS (application/javascript)**, or **xml (application/xml)**.

To create an MIME type, follow the instructions in “Add a MIME Type” on page 77.

c. Enter information in the **Template Text** box.

8. Click **Add Snippet**.

Edit a Snippet

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Snippets** subtab.
3. In the row for the snippet, select the filename or click . The Edit Snippet page appears.
4. In the **Template type** field, select **Upload** or **Text**.
5. If you selected the **Upload** template type, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
6. If you selected the **Text** template type, do the following:
   a. Enter information in the **File Name** field.

      Note: You can enter only letters, numbers, hyphens, and underscores

   b. From the **MIME Type** list, select **application_xml (application/xml)**, **JS (application/javascript)**, or **xml (application/xml)**.

      To create an MIME type, follow the instructions in “Add a MIME Type” on page 77.

   c. Enter information in the **Template Text** box.
7. Click **Update**.

Delete a Snippet

1. On the top navigation bar, click the **Assets** tab.
2. Click the **Snippets** subtab.
3. In the row for the snippet, click  and then click **OK**.

Part 6

Ad Sales

Chapter 28

Managing RFPs

About RFPs

Request for proposals (RFPs) are entered in the system with data from the Advertiser. After an RFP is entered into the system, multiple proposals can be generated for the RFP.

Add an RFP

1. On the top navigation bar, click the **Sales** tab.
2. Click the **RFPs** subtab.
3. Click **+ Add New RFP**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.,?;:).

5. In the **RFP Details** field, select **Upload** or **Text**.
   - If you selected **Upload**, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
   - If you selected **Text**, enter information in the box.

6. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the list. If you want to send the RFP to a new advertiser, click **Add a new advertiser**.

   Note: Enter `%` if you want to display the full list of advertisers.

7. If you want the selected advertiser to be billed, click **Bill to Advertiser**.
8. In the **Agency** field, enter the first few letters of an agency's name and then select the agency from the list. If you want to send the RFP to a new agency, click **Add a new agency**.

   Note: Enter `%` if you want to display the full list of agencies.

9. If you want the selected advertiser to be billed, click **Bill to Agency**.
10. In the **Billing Contact** field, enter the first few letters of a billing contact's name and then select the billing contact from the list. If you want to create a new billing contact, click **Add a billing contact**.

    Note: Enter `%` if you want to display the full list of billing contacts.

11. Enter a value in the **Budget $** field.
12. Select a **Salesperson** from the list.
13. Enter information in the **Notes** field.
14. Click **Save and Create Proposal** or **Add RFP** if you want to create more RFPs and build the proposal later.

View an RFP

1. On the top navigation bar, click the **Sales** tab.
2. Click the **RFPs** subtab.
3. Click  in the row for the RFP.
4. To add a proposal, click **Add New Proposal** and then follow the instructions in "Add an RFP" on page 121.

View RFPs for Your Team

1. On the top navigation bar, click the **Sales** tab.
2. Click the **RFPs** subtab.
3. Click **View Team RFPs**. Only the RFPs for your team are listed.

View Your RFPs

1. On the top navigation bar, click the **Sales** tab.

2. Click the **RFPs** subtab.
3. Click **View My RFPs**. Only the RFPs that belong to you are listed.

Edit an RFP

1. On the top navigation bar, click the **Sales** tab.
2. Click the **RFPs** subtab.
3. Click the RFP name or click  in the row for the RFP.
4. Modify the information in the **Name** field.
5. In the **RFP Details** field, select **Upload** or **Text**.
   - If you selected **Upload**, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
   - If you selected **Text**, modify the information in the box.
6. In the **Advertiser** field, enter the first few letters of an advertiser's name and then select the advertiser from the list. If you want to send the RFP to a new advertiser, click **Add a new advertiser**.

   Note: Enter `%` if you want to display the full list of advertisers.
7. If you want the selected advertiser to be billed, click **Bill to Advertiser**.
8. In the **Agency** field, enter the first few letters of an agency's name and then select the advertiser from the list. If you want to send the RFP to a new agency, click **Add a new agency**.

   Note: Enter `%` if you want to display the full list of agencies.

   If you want the selected advertiser to be billed, click **Bill to Agency**.
9. In the **Billing Contact** field, enter the first few letters of a billing contact's name and then select the billing contact from the autosearch results. If you want to create a new billing contact, click **Add a billing contact**.

   Note: Enter `%` if you want to display the full list of billing contacts.
10. Enter a value in the **Budget $** field.
11. Select a **Salesperson** from the list.
12. Enter information in the **Notes** field.
13. Click **Update**.

Delete an RFP

1. On the top navigation bar, click the **Sales** tab.
2. Click the **RFPs** subtab.
3. In the row for an RFP, click  and then click **OK**.

Chapter 29

Managing Proposals

About Proposals

Proposals are responses to RFPs, although proposals can be created without being associated with an RFP. If proposals are accepted, they are used to generate campaigns. Every time a proposal is edited, the previous version is saved, and can be accessed by clicking the number in the Version column on the **Proposals** tab.

Before a proposal can be converted into a campaign and flights, both the proposal and line items go through a set of checks to ensure they are set up correctly and are not violating any rules. This is referred to as compliance, and a proposal cannot be converted unless both the proposal and line items are found to be compliant.

Proposal compliance is based on the following:

- An advertiser credit check. If the advertiser has specified a credit limit, the proposal is not considered compliant unless the total cost of the proposal and total costs of the current flights for the advertiser are below this limit.
- Total proposal cost. The total proposal cost must not exceed the Proposal high value threshold, which is set under the **Settings** tab in the Sales section.
- An advertiser must be selected in a proposal or its RFP.

Line item compliance is based on the following:

- A price check and inventory check. The specified floor rate of the product is checked against the total revenue based on the specified flat rate or CPM of the line item. If the revenue is lower than the floor rate, then the line item is not considered compliant.
- A check is performed if a line item uses a restricted product.
- Any changes to an already-converted line item results in a check.
- Line item start date is not before today.
- Line item start date or end date is not outside the proposal's date range.

Add a Proposal

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Proposals** subtab.
3. Click **+ Add New Proposal**.
4. To associate the proposal with an RFP, enter the first few letters of an RFP in the RFP field and then select the RFP from the list.

   Note: Enter `%` if you want to display the full list of RFPs.
5. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"='), spaces, underscores, and punctuation marks (!.,?;:).
6. Select an **Insertion Order Logo** from the list.
7. In the **Insertion Order Template** list, select **Default** or **Custom**.
8. Select an **Insertion Order Locale** from the list.

   Note: Not all templates support the locale setting. The selected locale might have no effect on the template.
9. In the **Terms and conditions** field, enter the first few letters for terms and conditions and then select terms and conditions from the list.

   Note: Enter `%` if you want to display the full list of terms and conditions.
10. Click the **Expiration date** field and select a date in the calendar that appears.
11. Click the **Start date** field and select a date in the calendar that appears.
12. Click the **End date** field and select a date in the calendar that appears.
13. Select a **Currency** from the list.
14. In the **Advertiser** field, enter the first few letters of the advertiser's name and then select the advertiser from the list. If you want to associate the proposal with a new advertiser, click **Add a new advertiser** or **Add a new contact**. Repeat this step for every advertiser that you would like to add.

    Note: Enter `%` if you want to display the full list of advertisers.
15. If you want the selected advertiser to be billed, select **Bill to Advertiser**.
16. Enter information in the **Advertiser Contact** field. The billing contact is chosen automatically by using the first billing contact found for the advertiser depending on

which is selected for Bill To. If there is none, then the advertiser contact specified on the proposal is used. If not specified, then the first primary contact found for the advertiser is used. This field allows a contact to be specified for billing details in the proposal export instead of being chosen automatically.

17. Enter information in the **Advertiser Billing Contact** field.
18. In the **Agency** field, enter the first few letters of an agency's name and then select the agency from the list. If you want to associate the proposal with a new agency, click **Add a new agency**.

    Note: Enter `%` if you want to display the full list of agencies.
19. If you want the selected advertiser to be billed, select **Bill to Agency**.
20. Enter information in the **Agency Contact** field.
21. Enter information in the **Agency Billing Contact** field. The billing contact is chosen automatically by using the first billing contact found for the agency depending on which is selected for Bill To. If there is none, then the Agency Contact specified on the proposal is used. If not specified, then the first primary contact found for the agency is used. This field allows a contact to be specified for billing details in the proposal export instead of being chosen automatically.
22. Select a **Salesperson** from the list.
23. Select a **Trafficker** from the list. You can also select a second trafficker.
24. Enter a percentage value in the **Discount** field.
25. Enter information in the **External ID** field.
26. Enter information in the **PO Number** field.
27. Select a value from the **Probability to Campaign** list.

    This option sets a percentage of the likelihood that the proposal will be accepted and generated into a campaign. If the value is set at 90% or higher, the proposal and line items are included as reserved impressions in forecasts. The default value is 90% and can be configured by contacting Technical Support.
28. Enter information in the **Notes** field.
29. Enter information in the **Internal Notes** field.
30. Select an **Asset** to associate with the proposal.
31. Click **Create Proposal**, or select **Create Proposal & Edit** or **Create Proposal & Copy** from the list.

View a Proposal

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Proposals** subtab.
3. Click  in the row for the proposal.
4. From here you can click **View History** to view the history of the proposal, **Convert to Campaign** to convert the proposal into a campaign (see "Convert a Proposal" on page 131 for more information), **Edit Proposal** to edit the proposal, **Export**

Proposal to export the raw, unformatted proposal data into an XLSX file, or **Export PDF** to export the proposal into a PDF file.

5. If you want to add a line item, click **+ Add New Line Item** and then follow the instructions in “Add a Line Item” on page 128.

Add a Line Item

1. Enter information in the **Name** field.
2. To select a start date for the line item, click the **Start date** field, and select the date on which the line item should start. Next, select from the **Hour** list.
3. To select an end date for the line item, click the **End date** field, and select the date on which the line item should end. Next, select from the **Hour** list.
4. Select a **Currency** from the list.
5. To assign a product that contains targeting demographics tied with revenue data to the line item, enter information in the **Product** field. To filter the list of products, click and then do the following:

   a. Select **Match Any** or **Match All**.

   b. In the **Sites** field, enter the first few letters of a site’s name and then select the site from the list.

      Note: Enter `%` if you want to display the full list of sites.

   c. In the **Areas** field, enter the first few letters of a site’s name and then select the site from the list.

      Note: Enter `%` if you want to display the full list of areas.

   d. Click **Apply Filter**.

6. Select a **Product Line** from the list.
7. From the **Size** list, select an ad size.
8. Enter a value in the **Additional fees** field.
9. In the **Goals & Revenue** panel, do one of the following:
   - To apply a flat rate revenue to the flight, select **Flat Rate** and then do the following:
     - Enter a value in the **Flat Rate** field.
     - Enter a value in the **Goal** field for impressions, clicks, and actions.

       The flight attempts to serve enough to meet the goals specified in the fields. When the goals are reached, the flight stops serving.
     - Enter a value in the **Bonus** field for impressions, clicks, and actions.

       Revenue is not applied to bonus amounts. The amounts are treated as part of the overall goal by the ad server.
   - To apply a cost to every thousandth impression, click, or action, select **Cost per** and then do the following:
     - Enter a value in the **Cost per** field for impressions, clicks, and actions.

- Enter a value in the **Cost per** field for the daily revenue cap.

  The flight serves normally each day until reaching the revenue amount that is specified in this field.

- Enter a value in the **Goal** field for impressions, clicks, and actions.

  The flight attempts to serve enough to meet the goals specified in the fields. When the goals are reached, the flight stops serving.

- Enter a value in the **Bonus** field for impressions, clicks, and actions.

  Revenue is not applied to bonus amounts. The amounts are treated as part of the overall goal by the ad server.

The Estimated Cost Per **ECP** is calculated automatically and displayed in the table.

10. In the **Inventory** panel, click **Available** to perform an ad hoc inventory projection using the product specified on the flight. To include contending flights in the inventory projection, click **Contending Flights**. To set a goal for the inventory, click **Set Goal** and then enter a value.

11. The **Uplifts** panel displays the list rate associated with the line item. The uplift is applied whenever the tag is added to a line item. For more information, see "About Uplifts" on page 141.

12. In the **Frequency Caps** panel, do the following:

    a. Enter a value in **Quantity** field to specify the number of impressions, clicks, or conversions to serve to a visitor.

    b. Enter a value in **Period** field to specify the period of time that must elapse after the quantity is reached for the impressions, clicks, conversions before a visitor is eligible to receive the flight again. For example, if you entered `3` in the **Quantity** field and `1 week` in the **Period** field, the flight is shown three times to a visitor and then is not be shown again to the visitor until after the one-week waiting period.

    c. Enter a value in **Over Lifetime** field to specify the number of impressions, clicks, or conversions that can serve to a visitor over the lifetime of the flight.

    d. Enter a value in **Per Session** field to specify the number of conversions that can serve to a visitor over the course of a visitor session. A visitor session is initiated whenever an ad request is made, and, by default, persists for 30 minutes after the visitor's last request.

13. To make changes to targets, rules, tags, sites and areas, and categories, click **Show** in the **Advanced** panel, and then do the following:

    a. In the **Targets** panel, do the following

       i. From the **Targets group logic** list, select **Match All** or **Match Any**.

          Match All and **Match Any** are options used to determine how multiple targets are handled. **Match All** and **Match Any** correspond to the Boolean logic operators AND and OR. You select **Match All** to specify that all targets must match in order for the flight to serve. You select **Match Any** to specify that only one of the targets needs to match in order for the flight to serve.

       ii. In the **Target** field, search for and select a target. Targets are set up by clicking the **Targeting** tab and then the **Targets** tab.

    b. In the **Rules** panel, do the following:

       i. Select **Match All** or **Match Any**.

Match Any is the equivalent of the Boolean OR operator. If any of the statements are true, the target matches. **Match All** is the equivalent of the Boolean AND operator. All statements must be true for the target to match.

ii. Select a left value or a grouping rule. If you selected a grouping rule, select a left value and continue with the next step.

iii. Select an operator.

Note: Some operators are unavailable for the selected left value.

iv. Depending on the left value that you selected, enter a value or select from search results in the field.

Note: You can enter `%` to specify a wildcard of any length or an underscore to specify a wildcard of a single character.

v. If you want to add another grouping rule or left value, click .

c. In the **Tags** panel, do the following:

i. Click the **Custom**, **Geo**, or **Device** tab.

ii. Select a tag name from the list and then select the desired values. If you select multiple values, any of the values match.

iii. Click **Add Tag**.

iv. To add more tags, repeat steps 1–3.

Note: If you create multiple tags, every tag and at least one of the values from each has to match in order for the flight to serve.

d. In the **Sites and Areas** panel, do the following:

i. To specify sites that the flight can serve to, select **Selected Sites**, place your mouse pointer in the **Sites** field, search for sites, and then select them. To add a site that the flight can serve to, click **Add a new site**, enter information in the required **Name** field, and in the optional **Description** and **Revenue cut** fields, and then click **Add Site**.

ii. To specify areas that the flight can serve to, select **Selected Areas**, place your mouse pointer in the **Areas** field, search for areas, and then select them. To add an area that the flight can serve to, click **Add a new area**, enter information in the required**Name** field and the optional **Sites** field, and then click **Add Area**.

e. In the **Categories** panel, enter the first few letters of a category's name and then select the site from the list.

Note: Enter `%` if you want to display the full list of categories.

14. In the **Other** panel, enter information in the **Notes** and **External ID** fields.

15. Click **Create Line Item** or click the down arrow next to **Create Line Item** and click **Create Line Item & Split By Month**, **Create Line Item & Add Another**, **Create Line Item & Add Go To Next**, **Create Line Item & Copy**, or **Create Line Item & Make Many Copies**.

Convert a Proposal

Before a proposal can be converted into a campaign and flights, both the proposal and line items go through a set of checks to ensure they are set up correctly and are not violating any rules. This is referred to as compliance, and a proposal cannot be converted unless both the proposal and line items are found to be compliant. Any outstanding triggered alerts that require acknowledgment or approval prevent a proposal from being converted.

If the proposal and line items are compliant, a **Convert to Campaign** button appears on the View Proposal page. Click this button to create a campaign and flights with the information from the proposal and line items. Once converted, a link to the converted campaign is shown on the View Proposal page.

If the proposal or line item fields are edited, it might be necessary to reconvert the proposal. You can select the line items to reconvert. If the edits require reconversion, the **Convert to Campaign** button reappears on the View Proposal page.

View Proposals for Your Team

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Proposals** subtab.
3. Click **View Team Proposals**. Only the proposals for your team are listed.

View Your Proposals

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Proposals** subtab.
3. Click **View My Proposals**. Only the proposals that belong to you are listed.

Edit a Proposal

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Proposals** subtab.
3. Click the proposal name or click  in the row for the proposal.
4. To make changes:
   - To associate the proposal with an RFP, enter the first few letters of an RFP in the RFP field and then select the RFP from the list.

Note: Enter `%` if you want to display the full list of RFPs.

- Modify the information in the **Name** field.
- Select an **Insertion Order Logo** from the list.
- From the **Insertion Order Template** list, select **Default** or **Custom**.
- Select an **Insertion Order Locale** from the list.

  Note: Not all templates support the locale setting. The selected locale might have no effect on the template.
- In the **Terms and conditions** field, enter the first few letters for terms and conditions and then select terms and conditions from the list.

  Note: Enter `%` if you want to display the full list of terms and conditions.
- Click the **Expiration date** field, and then select a date in the calendar that appears.
- Place your mouse pointer in the **Start date** field, and then select a date in the calendar that appears.
- Place your mouse pointer in the **End date** field, and then select a date in the calendar that appears.
- Select a **Currency** from the list.
- In the **Advertiser** field, enter the first few letters of an advertiser's name and then select the advertiser from the list. If you want to send the RFP to a new advertiser, click **Add a new advertiser**.

  Note: Enter `%` if you want to display the full list of advertisers.

  If you want the selected advertiser to be billed, click **Bill to Advertiser**.

  You can also add a new contact by clicking **Add a new contact**.
- Enter information in the **Advertiser Contact** field.
- In the **Agency** field, enter the first few letters of an agency's name and then select the advertiser from the list. If you want to send the RFP to a new agency, click **Add a new agency**.

  Note: Enter `%` if you want to display the full list of agencies.

  If you want the selected advertiser to be billed, click **Bill to Agency**.
- Enter information in the **Agency Contact** field.
- Select a **Salesperson** from the list.
- Select a **Trafficker** from the list. You can also select a second trafficker.
- Enter a percentage value in the **Discount** field.
- Enter information in the **External ID** field.
- Enter information in the **PO Number** field.
- Select a value from the **Probability to Campaign** list.
- Enter information in the **Notes** field.
- Enter information in the **Internal Notes** field.

5. If you want to add a line item, click **+ Add New Line Item** and then follow the instructions in "Add a Line Item" on page 128. Otherwise, continue with the next step.

6. Click **Update Proposal**, or select the arrow and click **Update Proposal & Edit** or **Update Proposal & Copy** .

Copy a Proposal

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Proposals** subtab.
3. In the row for a flight, click .
4. Click **OK**. The Edit Proposal page appears.
5. Make any desired changes and then click **Update Proposal**.

Delete a Proposal

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Proposals** subtab.
3. In the row for a proposal, click  and then click **OK**.

Chapter 30

Managing Products

About Products

Products are a combination of rate-card data tied to targets. That rate-card data consists of the list rate, goal rate, and floor rate. Tags and targets are built on key=value pairs that represent the data that is passed to SAS Intelligent Advertising for Publishers. A thorough set of keys and values (taxonomy) results in effective targets and products. This data influences real-time ad selection, can help refine your concept of audience, and improve product definitions. Products are later specified in proposals, and are assigned to flights after they are booked. Products can be reported on through the BI reporting tool, and they provide an easy way of viewing revenue against rate-card and traffic data for a target or groups of targets.

Add a Product

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Products** subtab.
3. Click + **Add New Product**.
4. Enter information in the **Name** field.

Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.,?;:).

5. Enter information in the **Description** field.

6. Select a **Tier** from the list. This specifies what tier the product can be assigned to. A lower revenue or lower priority product might need to be limited to lower tiers to not interfere with flights in higher tiers.

7. If you want to restrict the product, select **Yes** next to **Restricted**.

   If you selected **Yes**, the **Alerts** panel appears. In the **Alerts on violation** field, enter the first few letters of the contact to whom you would like to receive alerts and then select the contact from the list. Repeat this step for each contact that you would like to add.

   Note: Enter `%` if you want to display the full list of contacts.

   To specify the type of alerts, select **Notifications** or **Require Approval**.

8. If you want to package the product with other products, select **Yes** next to **Package**.

   Note: A package that is added to a flight does not by itself apply any targeting. Each product from the package must first be associated with a creative, and then the targeting from that product is applied to the creative.

   If you selected **Yes**, the **Products** panel appears. In the **Products** field, enter the first few letters of a product that you would like to package with the new product and then select the product from the list. Repeat this step for each product that you would like to add.

   Note: Enter `%` if you want to display the full list of products.

9. In the **Sizes** field, enter a value and then select a value from the list. Enter `%` if you want to display the full list of values.

   Note: If you are packaging the product with existing products, the **Sizes** field is not available.

10. In the **Target group logic** list, select **Match All** or **Match Any**.

    Note: If you are packaging the product with existing products, the **Target group logic** list is not available.

11. Enter information in the **Targets** field. Multiple targets can be assigned to a single product. Once a product is assigned to a flight, the targeting associated with that product determines where the flight serves.

    Note: If you are packaging the product with existing products, the **Targets** field is not available.

12. If you want to profile the product in a summary report, select **Profile in product sum report**. When this option is disabled, this product is not included in data for the product report, which can speed up data processing time.

13. If you want the product to be the top priority, select **Top Priority**. When enabled, this option carries over to any flights that are converted from line items that use this product.

14. In the **Companion Type** list, select **None**, **Strict**, or **Loose**. When the setting is either Strict or Loose, this option carries over to any flights that are converted from line items that use this product.

15. In the **Terms and conditions** field, enter the first few letters for terms and conditions and then select terms and conditions from the list.

Note: Enter `%` if you want to display the full list of terms and conditions.

16. In the **Additional Targeting** field, select **Dynamic**, **Ad-hoc**, or **None**.
17. If you select **Dynamic**, then you must select a target from the list and then click **Add Dynamic Target**. Repeat this step for as many dynamic targets that you would like to add.

    You use dynamic targeting for instances where targeting is complex or granular (for example, the product is intended to target to any ZIP code). Only the tag itself, and no values, are selected for the Product.
18. If you select **Ad-hoc**, follow these steps:
    a. To specify sites for the product, select **Selected Sites**, place your mouse pointer in the **Sites** field, search for sites, and then select them. To add a site for the product, click **Add a new site**, enter information in the required **Name** field, and in the optional **Description** and **Revenue cut** fields, and then click **Add Site**.
    b. To specify areas for the area, select **Selected Areas**, place your mouse pointer in the **Areas** field, search for areas, and then select them. To add an area for the product, click **Add a new area**, enter information in the required **Name** field and the optional **Sites** field, and then click **Add Area**.
    c. Click the **Custom**, **Geo**, or **Device** tab. Select the tag name and then the values desired. If multiple values are selected, any of the values can match.
    d. Select a tag name from the list and then the desired values. If you select multiple values, any of the values match.
    e. Click **Add Tag**.
    f. To add more tags, repeat these steps.
19. Enter a value in the **List Rate $** field. This is the rate card price of the product from a media kit.
20. From the **List Rate** list, select **CPM**, **CPC**, **CPA**, or **Flat Rate**.
21. In the **Additional Pricing** field, select **Additional rates**, **Discounted rates**, or **No additional rates**.

    If you selected **Additional rates**, enter values in the **Floor Rate $** and **Goal Rate $** fields. The Goal Rate is the price at which the product is expected to be sold at, assuming discounting from the list rate. The Floor Rate specifies the lowest price acceptable for the product and triggers notifications or required pricing approvals if a proposal uses an eCPM below the floor rate. If you selected **Discounted rates**, enter a value in the **Discount %** field.
22. In the **Teams** panel, select the teams to whom you would like to assign the product.
23. Click **Add Product**.

View Targeting Details

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Products** subtab.
3. In the row for a product, click . A window appears where you can view the targeting details for the selected product.

4. Click **Close**.

View Products for Your Team

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Products** subtab.
3. Click **View Team Products**. Only the products for your team are listed.

View Your Products

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Products** subtab.
3. Click **View My Products**. Only the products that belong to you are listed.

Copy a Product

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Products** subtab.
3. In the row for a product, click . The Edit *product name* Product page appears.
4. Make any desired changes and then click **Update Product** or click the arrow next to **Update Product** and select **Update Product & Copy** or **Update Product & Make Many Copies**.

Edit a Product

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Products** subtab.
3. Click the product name or click in the row for the product.
4. To make changes:
   - Modify the information in the **Name** field.
   - Enter information in the **Description** field.
   - Select a **Tier** from the list.
   - If you want to restrict the product, select **Yes** next to **Restricted**.

If you selected **Yes**, the **Alerts** panel appears. In the Alert fields, enter the first few letters of the users and teams to whom you would like to receive alerts, and then select the contact from the list. Repeat this step for each contact that you would like to add.

Note: Enter `%` if you want to display the full list of contacts.

To specify the type of alerts, select **Notifications** or **Require Approval**.

- If you want to package the product with other products, select **Yes** next to **Package**.

  If you selected **Yes**, the **Products** panel appears. In the **Products** field, enter the first few letters of a product that you would like to package with the new product and then select the product from the list. Repeat this step for each product that you would like to add.

  Note: Enter `%` if you want to display the full list of products.

- In the **Sizes** field, enter a value and then select a value from the autosearch results. Enter `%` if you want to display the full list of values.

  Note: If you are packaging the product with existing products, the **Sizes** field is not available.

- In the **Target group logic** list, select **Match All** or **Match Any**.

  Note: If you are packaging the product with existing products, the **Target group logic** list is not available.

- Enter information in the **Targets** field.

  Note: If you are packaging the product with existing products, the **Targets** field is not available.

- If you want to profile the product in a summary report, select **Profile in product sum report**.

- If you want the product to be the top priority, select **Top Priority**.

- In the **Terms and conditions** field, enter the first few letters for terms and conditions and then select terms and conditions from the list.

  Note: Enter `%` if you want to display the full list of terms and conditions.

- In the **Additional Targeting** field, select **Dynamic**, **Ad-hoc**, or **None**.

  If you selected **Dynamic**, select a target from the list and then click **Add Dynamic Target**. Repeat this step for as many dynamic targets that you would like to add.

  If you select **Ad-hoc**, follow these steps:

  1. To specify sites for the product, select **Selected Sites**, place your mouse pointer in the **Sites** field, search for sites, and then select them. To add a site for the product, click **Add a new site**, enter information in the required **Name** field, and in the optional **Description** and **Revenue cut** fields, and then click **Add Site**.

  2. To specify areas for the area, select **Selected Areas**, place your mouse pointer in the **Areas** field, search for areas, and then select them. To add an area for the product, click **Add a new area**, enter information in the required **Name** field and the optional **Sites** field, and then click **Add Area**.

  3. Click the **Custom**, **Geo**, or **Device** tab. Select the tag name and then the values desired. If multiple values are selected, any of the values can match.

4. Select a tag name from the list and then the desired values. If you select multiple values, any of the values match.
        5. Click **Add Tag**.
        6. To add more tags, repeat steps 1–5.
    - Enter a value in the **List Rate $** field.
    - From the **List Rate** list, select **CPM**, **CPC**, **CPA**, or **Flat Rate**.
    - In the **Additional Pricing** field, select **Additional rates**, **Discounted rates**, or **No additional rates**.

      If you selected **Additional rates**, enter values in the **Floor Rate $** and **Goal Rate $** fields. If you selected **Discounted rates**, enter a value in the **Discount %** field
    - In the **Teams** panel, select the teams to whom you would like to assign the product.
    - Enter information in the **Internal Notes** field.
5. If you want to add a line item, click **+ Add New Line Item** and then follow the instructions in “Add a Line Item” on page 128. Otherwise, continue with the next step.
6. Click **Update Product** or click the arrow next to **Update Product** and select **Update Product & Copy** or **Update Product & Make Many Copies**.

Delete a Product

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Products** subtab.
3. In the row for a product, click  and then click **OK**.

Chapter 31

Managing Uplifts

About Uplifts

Uplifts consist of a tag and a list rate that can be applied to line items. Uplifts can be thought of as options that can be added to a line item that increase the value of the line item. Products or specific tag values can also be added to an uplift.

When a general uplift is created, it is applied whenever that tag is added to a line item. For example, if an uplift is created for the Age tag with a list rate of $1.00, that rate is effectively added to the floor rate of the line item when the Age tag is specified in the Tags targeting section. The tag must be applied in the **Tags** or **Sites & Areas** section in the Line Item page. The Line Item page appears after you click **Show** in the **Advanced targeting** section in the Edit Line Item window.

Add an Uplift

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Uplifts** subtab.
3. Click + **Add New Uplift**.
4. Enter information in the **Tag** field.
5. Enter a value in the **List Rate $** field.
6. Click **Add Uplift** or click the arrow next to **Add Uplift** and select **Add Product Uplift** or **Add Tag Value Uplift**.
7. If you selected **Add Product Uplift**, do the following:
    a. Enter information in the **Product** field.

   b. Enter a value in the **List Rate $** field.
   c. Click **Add Product Uplift**.
8. If you selected **Add Tag Value Uplift**, do the following:
   a. Enter information in the **Tag value** field.
   b. Enter a value in the **List Rate $** field.
   c. Click **Add Tag Value Uplift**.

View an Uplift

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Uplifts** subtab.
3. Click in the row for the uplift.

Edit an Uplift

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Uplifts** subtab.
3. Click the uplift name or click in the row for the uplift.
4. Make desired changes in the **Tag** and **List Rate $** fields.
5. Click **Save Uplift** or click the arrow next to **Save Uplift** and select **Add Product Uplift** or **Add Tag Value Uplift**.
6. If you selected **Add Product Uplift**, do the following:
   a. Enter information in the **Product** field.
   b. Enter a value in the **List Rate $** field.
   c. Click **Add Product Uplift**.
7. If you selected **Add Tag Value Uplift**, do the following:
   a. Enter information in the **Tag value** field.
   b. Enter a value in the **List Rate $** field.
   c. Click **Add Tag Value Uplift**.

Delete an Uplift

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Uplifts** subtab.

3. In the row for an uplift, click  and then click **OK**.

Chapter 32

Managing Terms and Conditions

About Terms and Conditions

Terms and conditions represent the details of a contract accompanying an insertion order. Terms and conditions can be uploaded as a PDF or text and then appended to the end of a proposal during export.

Add Terms and Conditions

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Terms and Conditions** subtab.
3. Click **+ Add New Terms and Conditions**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"='), spaces, underscores, and punctuation marks (!.,?;:).
5. In the **Advertisers** field, enter the first few letters of an advertiser's name and then select an advertiser from the autosearch results. Repeat this step for every advertiser that would like to add.

   Note: Enter `%` if you want to display the full list of advertisers.
6. In the **Agencies** field, enter the first few letters of an agency's name and then select the agency from the autosearch results. Repeat this step for every advertiser that you would like to add.

   Note: Enter `%` if you want to display the full list of agencies.
7. In the **Template type** field, select **Upload** or **Text**.

- If you selected the Upload template type, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.
- If you selected the Text template type, enter information in the **Terms and conditions text** box.

8. Click **Add Terms and Conditions**.

Edit Terms and Conditions

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Term and Conditions** subtab.
3. Click the terms and conditions name or click in the row for the terms and conditions. If you click a terms and conditions name, a sample of the verbiage is displayed, then click **Edit Terms & Conditions**.
4. To make changes:
   - Modify the information in the **Name** field.
   - In the **Advertisers** field, enter the first few letters of an advertiser's name and then select an advertiser from the autosearch results. Repeat this step for every advertiser that would like to add.

     Note: Enter `%` if you want to display the full list of advertisers.
   - In the **Agencies** field, enter the first few letters of an agency's name and then select the agency from the autosearch results. Repeat this step for every advertiser that you would like to add.

     Note: Enter `%` if you want to display the full list of agencies.
   - In the **Template type** field, select **Upload** or **Text**.

     If you selected the Upload template type, click **Choose File**, navigate to the file that you want to upload, and then click **Open**.

     If you selected the Text template type, enter information in the **Terms and conditions text** box.
5. Click **Update**.

Delete Terms and Conditions

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Terms and Conditions** subtab.
3. In the row for the terms and conditions, click and then click **OK**.

Chapter 33

Checking the Availability of Inventory

About Checking Inventory Availability

SAS Intelligent Advertising for Publishers enables you to search for all pre-emptible inventory in the forecast that matches the flight's targeting and flight dates. Pre-emptible refers to any flight sitting in a tier with the **Pre-emptible** option selected and represents the available inventory in a simulation. Forecasting and fulfillment knowledge based on capacity, booked and reserved, and available inventory data are linked so that you can accurately determine inventory availability. For more information, see Chapter 6, “Managing Tiers,” on page 43. Note that this calculation takes into account any default ads that appear in the forecast that match the flight's criteria. The list of contending flights is limited to 100 flights.

When checking availability on a flight or line item, the results returned are split into Available, Reserved, and Served numbers:

- Available – displays the number of impressions available to book not including the current flight or line item's reservation. If availability is checked after the start date, then the number returned shows the availability for the remaining time period.
- Reserved – displays the number of impressions that the current flight or line item is projected to serve. If checked after the start date, the number returned shows what is projected for the remaining time period. Reserved is calculated as the difference between that estimated availability and the contending flights total, so it might not match the amount the flight is actually forecast to serve when a frequency cap is involved.
- Served – displays the number of impressions the flight has served through the current date.

Check Inventory Availability

To check the availability of the inventory:

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Check Avails** subtab.
3. Expand the **Filters** panel.
4. Select a **Date Range** from the list.
5. In the **Break down by** field, select **Total**, **Month**, or **Day**.
6. Enter a **Product**. You can also filter by product site and area.

   Note: Enter `%` if you want to display the full list of products.
7. Select a **Size** from the list.
8. If you want to check the availability of flights for other products, select **Contending Flights**.

   The **Contending Flights** tool helps you address the lack of availability for a product by providing a list of flights to which you might make changes to that could free up impression inventory.
9. To filter availability by targets, do the following:

   Note: You might need to click **Show** to display the **Targets**, **Tags**, and **Sites & Areas** panels.

   a. In the **Targets** panel, select **Match All** or **Match Any**.

   b. In the **Targets** field, search for and select a target.
10. To filter availability by tags, do the following:

    a. In the **Tags** panel, click the **Custom**, **Geo**, or **Device** tab. Select the tag name and then the values desired. If multiple values are selected, any of the values can match.

    b. To add more tags, click **Add Tag** and repeat this step.
11. To filter availability by specific sites, select **Selected Sites**, click in the **Sites** field, search for sites, and then select them. To make a site available for future logging and inventory forecasting, click **Add a new site**, enter information in the required **Name** field, and in the optional **Description** and **Revenue cut** fields, and then click **Add Site**.
12. To filter availability by specific areas, select **Selected Areas**, click in the **Areas** field, search for areas, and then select them. To make an area available for future logging and inventory forecasting, click **Add a new area**, enter information in the required **Name** field and the optional **Sites** field, and then click **Add Area**.
13. To filter availability by frequency cap values:

    a. In the **Frequency Caps** table, enter a value in the **Quantity** field to specify the number of impressions, clicks, or conversions to use as filters.

    b. In the **Frequency Caps** table, enter a value in the **Period** field to specify the period of time that must elapse after the quantity is reached for the impressions, clicks, conversions to use as a filter.

c. In the **Frequency Caps** table, enter a value in the **Over Lifetime** field to specify the number of impressions, clicks, or conversions over the lifetime of a flight to use as a filter.

d. In the **Frequency Caps** table, enter a value in the **Per Session** field to specify the number of conversions over the course of a visitor session to use as a filter. A visitor session is initiated whenever an ad request is made, and, by default, persists for 30 minutes after the visitor's last request.

14. To use schedule filters, expand the **Schedule** panel and then do the following:

    a. Select **Daily**, **Weekly**, or **Monthly**.

       If you selected **Daily**, select a value from the **Run at Hour** list.

       If you selected **Weekly** or **Monthly**, select a value from the **Run on Day** and **Run at Hour** lists.

    b. Enter a time value in the **Expire in** field.

15. To create an Excel file that contains the availability information, click **Export xslx**.
16. To create a comma-separated file that contains the availability information, click **Export csv**.
17. To create a schedule report, click **Schedule Report**.
18. To manage schedules, click **Manage Schedules**.
19. Click **Check Availability**.

Chapter 34

Setting Alerts

About Alerts

Alerts can be configured that notify or require action by specified users when a certain rule or condition is violated. For example, you can set up an alert that requires specified users to be notified when a price is set below floor rate, when too much inventory is allocated when submitting a line item, or when a credit limit on the proposal is violated. When you create a proposal or line item that violates a configured rule, the system requires that a notification of the violation be sent to the specified user. The proposal or line item can still be submitted. You can also specify that an approval is required before the edit is accepted.

Triggered alerts can be viewed in the **Alerts** tab of the My Work section. See “My Work Tab” on page 15 for more information.

Types of alerts include the following:

- Price – triggered if a line item is booked below the assigned product's floor price. See “About Products” on page 135 for more information on floor pricing.
- Credit – triggered if the credit limit specified for the advertiser on a proposal or line item is violated. See “Set Credit Limit” on page 161 for more information about setting credit limits.
- Inventory – triggered when the impression goal for a line item is higher than what is calculated as available. See “About Checking Inventory Availability” on page 147 for more information about availability.
- High Value Proposal – triggered if a proposal's value exceeds the value specified in the **Proposal high value threshold** field. See “About Settings” on page 153 for more information this field.
- Change Order – This alert is triggered when a converted line item is modified in a way that would affect the associated flight. When triggered, a pop-up dialog box appears showing which fields on the line item were changed, what the value of the fields were when the line item was last converted, what the value of the fields are on the associated flight, and what the values of the fields on the associated flight will be if the line item is reconverted.

Set Alerts

To set an alert:

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Alert Settings** subtab.
3. Select a team setting from the list and then click **Choose**.
4. In the **Alerts on violation** fields for the **Price**, **Credit**, **Inventory**, **High Value Proposal**, and **Change Order** sections, enter the first few letters of a contact and then select contacts from the drop-down list.
5. In the **Level** field for the **Price**, **Credit**, **Inventory**, **High Value Proposal**, and **Change Order** sections, select either **None**, **Warnings**, **Notifications**, or **Require Approval**.
6. Click **Save**.

Chapter 35

Setting View Options

About Settings

In the Settings tab, you can specify certain options for the solution.

Set Options

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Settings** subtab.
3. To specify a tax rate that will be included in the calculated totals for proposals exported to PDF files, enter a value in the **Tax rate (%)** field.
4. To make a proposal fail validation if both advertiser and agency are missing, select **Proposal Save Requires Advertiser or Agency**.
5. Enter information in the **Business ID Label 1** and **Business ID Label 2** fields if you want the advertiser and agency windows to display the fields and use the values for the labels.
6. If you want to require a business ID for the billed to advertiser or agency before an insertion order can be generated for a proposal, select **Business ID Required For IO Generation**.
7. If you want to require a legal entity name for the billed to advertiser or agency before an insertion order can be generated for a proposal, select **Legal Entity Name Required For IO Generation**.
8. If you want to require a salesperson before an insertion order can be generated for a proposal, select **Salesperson Full Name Required For IO Generation**.
9. If you want to generate a PO number, select **Po number auto generated**.
10. If you want to require a PO number before an insertion order can be generated for a proposal, select **PO Number Required For IO Generation**.

11. If you want to require that all line items have end dates before an insertion order can be generated for a proposal, select **Line Item End Dates Required For IO Generation**.
12. If you want a line item name auto-generated with the product name, select **Line Item Name Auto Generated with Product Name**.
13. If you want to allow PDF exports only on proposals that can be converted to a campaign, select **Restrict PDF Export on Uncovertable Proposal**.
14. Enter information in the **Bill Direct Label** field to create a label that will be displayed in the advertiser window.
15. Enter information in the **Bill Direct Other Choice Label** field to create the label for the second radio button of the **bill direct** field in the advertiser windows.
16. If you want to add a proposal logo, click **Choose File**, navigate to the desired JPEG or PNG file and then click **Open**.
17. Enter a value in the **Logo height** field so that the logo height is scaled to this value in proposal export.

    Note: You cannot enter a value greater that 100.
18. In the **Proposal Template** list, select **Default** or **Custom**.
19. Select a value in the **Proposal Threshold Expiration** list to specify the number of days before a proposal expires after crossing the auto conversion threshold.
20. Select a value in the **Proposal expiration notify in days** list. This setting configures the number of days prior to a proposal's expiration date that an expiration alert email is sent.
21. Enter a value in the **Proposal high value threshold** field.
22. Select the fields that you want to propagate into the destination object during the conversion process from the **Flight Conversion Properties** list. When a field is disabled, it is not updated during conversion for the corresponding flight.
23. Click **Save**.

Chapter 36

Managing Insertion Order Logos

About Insertion Order Logos

SAS Intelligent Advertising for Publishers enables you to upload and change artwork files that will be placed as a logo in an advertiser's insertion order.

Add an Insertion Order Logo

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Insertion Order Logos** subtab.
3. Click + **Add New Insertion Order Logo**.
4. Click **Choose File**, navigate to a JPEG or PNG file and then click **Open**.
5. Click **Add Insertion Order Logo**.

Edit an Insertion Order Logo

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Insertion Order Logos** subtab.
3. Click the insertion order logo name or click in the row for the insertion order logo.
4. Click **Choose File**, navigate to a JPEG or PNG file and then click **Open**.

5. Click **Update**.

Delete an Insertion Order Logo

1. On the top navigation bar, click the **Sales** tab.
2. Click the **Insertion Order Logos** subtab.
3. In the row for an insertion order logo, click and then click **OK**.

Part 7

Customers

Chapter 37

Managing Advertisers

About Advertisers

You can add as many advertisers as you like. When you add an advertiser, you need to assign it to one or more categories.

You create advertiser categories in order to prevent competing advertisers from displaying on the same page view. For example, if two advertisers are categorized as *Automaker*, only the first advertiser is displayed on the page view. In other words, a categorized advertiser does not compete with itself. Note that an advertiser can serve more than once to a given page view. In this context, *Page view* refers to a sequential set of ad calls with the same **`/viewid=`** value.

In addition to attaching a category to an advertiser in the system, you must apply the category to the advertiser's flights so that certain flights are excluded and others are not excluded. More than one category can be applied to the advertiser and its flights. You cannot pre-define categories. You add categories to an advertiser when you are creating or editing the advertiser. You can apply categories on a per-flight basis. See "About Flights" on page 39 for information about flights.

Add an Advertiser

1. Select **Customers** ⇨ **Advertisers** ⇨ **Add New Advertiser**.
2. Enter information in the **Name** field.

Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), spaces, underscores, and hyphens.

3. Enter information in the **Legal Entity Name** field.

   The legal entity name is used for proposal PDF file exports, and is used in cases where the name needs to contain different characters or a longer and more formal version of the advertiser name that is used in other areas of SAS Intelligent Advertising for Publishers.

4. In the **Credit Limit $** field, specify a monetary limit that if exceeded creates an alert or prevents the booking of a proposal.

   Note: Only Finance and Administrator users can modify the **Credit Limit $** field.

5. If you want to require prepayment from the advertiser, select **Advertiser must pre-pay**.

6. In the **Standard discount (%)** field, enter a value that will be applied to exported proposal PDF files.

   Note: Only Finance and Administrator users can modify the **Standard discount (%)** field.

7. Select an item from the **Industry** list.

   The BI reports group advertisers by industry. For more information about industries, see Chapter 40, "Managing Industries," on page 173.

8. Select an item from the **Region** list.

   The BI reports group advertisers by region. For more information about regions, see Chapter 41, "Managing Regions," on page 175.

9. Enter a value in the **Account number** field.

10. In the **Terms and Conditions** field, enter the first few letters of a name for terms and conditions. Then select terms and conditions from the autosearch results. For more information, see Chapter 32, "Managing Terms and Conditions," on page 145.

    Note: Enter % if you want to display the full list of term and conditions.

    The selected terms and conditions are used by default when the advertiser is selected in a proposal.

11. Enter values in the **Identification number** fields.

    Note: For example, enter a tax identification number, business number, or company number.

12. To set the invoicing method for the advertiser, select **Direct** or **Agency**.

13. Select a **Customer Status** from the list.

14. In the **Custom Actions** panel, enter values in the **Revenue $** fields for the default actions. For more information, see Chapter 7, "Managing Custom Actions," on page 51.

    If you want to delete an existing default action, click the "X" next to the action name.

15. If you want to create new actions, click **+ Add Action** and then do the following:

    a. In the **Action** field, enter the first few letters of an action and then select the action from the autosearch results.

       Note: Enter % if you want to display the full list of actions.

b. Enter a value in the **Revenue $** field.

c. Repeat for as many actions that you would like to add.

16. In the **Categories** panel, enter information in the **New category** field and then click **Add**. For more information, see "About Advertisers" on page 159.
17. To add a contact for the advertiser, click **Add a Contact**, enter information and make selections in the Add New Contact window. Then click **Add Contact**.
18. Click **Update**.

Set Credit Limit

1. To quickly add a credit limit to an advertiser, click the **Customers** tab on the top navigation bar.
2. Click the **Advertisers** subtab.
3. Click **Set Credit Limit** in the row of the advertiser.
4. Enter an amount in the **Credit limit** field.
5. Select **Advertiser must pre-pay** to require pre-payment from the advertiser.
6. Click **Set Credit Limit** . The credit limit appears in the table.

View an Advertiser

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Advertisers** subtab.
3. Click the advertiser name.

View Advertisers for Your Team

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Advertisers** subtab.
3. Click **View Team Advertisers**. Only the advertisers for your team are listed.

View Your Advertisers

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Advertisers** subtab.

3. Click **View My Advertisers**. Only the advertisers that belong to you are listed.

Edit an Advertiser

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Advertisers** subtab.
3. Click the advertiser name or click  in the row for the advertiser.
4. Modify the information in the **Name** field.

   Note: You can enter only letters, numbers, hyphens, spaces, and underscores
5. Enter information in the **Legal Entity Name** field.

   The legal entity name is used for proposal PDF file exports. This name is used in cases where the name needs to contain different characters or a longer, more formal version of the advertiser name used in other areas of the UX.
6. In the **Credit Limit $** field, specify a monetary limit that if exceeded creates an alert or prevents the booking of a proposal.

   You can also set the credit limit by clicking **Credit Limit** in the row for the advertiser and then entering the value in the Set Credit Limit window. You can also set the prepayment requirement for the advertiser in the Set Credit Limit window.

   Note: Only Finance and Administrator users can modify the **Credit Limit $** field.
7. If you want to require prepayment from the advertiser, select **Advertiser must pre-pay**.
8. In the **Standard discount (%)** field, enter a value that will be applied to exported proposal PDF files.

   Note: Only Finance and Administrator users can modify the **Standard discount (%)** field.
9. Select an item from the **Industry** list.
10. Select an item from the **Region** list.
11. Enter a value in the **Account number** field.
12. In the **Terms and Conditions** field, enter the first few letters of a name for terms and conditions. Then select terms and conditions from the autosearch results.

    Note: Enter *%* if you want to display the full list of term and conditions.
13. Enter values in the **Identification number** fields.

    Note: For example, enter a tax identification number, business number, or company number.
14. To set the invoicing method for the advertiser, select **Direct** or **Agency**.
15. Select a **Customer Status** from the list.
16. In the **Custom Actions** panel, enter values in the **Revenue $** fields for the default actions.

    If you want to delete an existing default action, click the X next to the action name.
17. If you want to create new actions, click **+ Add Action** and then do the following:

a. In the **Action** field, enter the first few letters of an action and then select the action from the autosearch results.

   Note: Enter % if you want to display the full list of actions.

b. Enter a value in the **Revenue $** field.

c. Repeat for as many actions that you would like to add.

18. In the **Category** panel, enter information in the **New category** field and then click **Add**. For more information, see “About Advertisers” on page 159.

19. To add a contact for the advertiser, click **Add a Contact**, enter information and make selections in the Add New Contact window, and then click **Add Contact**.

20. Click **Update**.

Delete an Advertiser

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Advertisers** subtab.
3. In the row for an advertiser, click  and then click **OK**.

Chapter 38

Managing Agencies

About Agencies

You can add as many agencies as you like, and you can create categories for those agencies.

Add an Agency

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Agencies** subtab.
3. Click **Add New Agency**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a-z, A-Z,0-9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.,?;:).

5. Enter information in the **Legal Entity Name** field.

   The legal entity name is used for proposal PDF file exports, and is used in cases where the name needs to contain different characters or a longer, more formal version of the agency name used in other areas of SAS Intelligent Advertising for Publishers.

6. In the **Credit Limit $** field, specify a monetary limit that if exceeded creates an alert or prevents the booking of a proposal.

   Note: Only Finance and Administrator users can modify the **Credit Limit $** field.

7. If you want to require prepayment from the agency, select **Agency must pre-pay**.

8. In the **Standard discount (%)** field, enter a value that will be applied to exported proposal PDF files.

   Note: Only Finance and Administrator users can modify the **Standard discount (%)** field.

9. In the **Commission (%)** field, enter a value that will be applied to exported proposal PDF files.

   Note: Only Finance and Administrator users can modify the **Commission (%)** field.

10. Enter an alphanumeric value in the **External ID** field.

11. Select an item from the **Region** list.

12. Enter a value in the **Account number** field.

13. In the **Terms and Conditions** field, enter the first few letters of a name for terms and conditions. Then select terms and conditions from the autosearch results.

    Note: Enter *%* if you want to display the full list of term and conditions.

    The selected terms and conditions are used by default when the agency is selected in a proposal.

14. Enter values in the **Identification number** fields.

    Note: For example, enter a tax identification number, business number, or company number.

15. Select a customer status from the list.

16. In the **Categories** panel, enter information in the **New category** field and then click **Add**. For more information, .

17. To add a contact for the agency, click **Add a Contact**, enter information and make selections in the Add New Contact window, and then click **Add Contact**.

18. Click **Update**.

View an Agency

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Agencies** subtab.
3. Click the agency name.

View Agencies for Your Team

1. On the top navigation bar, click the **Customers** tab.

2. Click the **Agencies** subtab.
3. Click **View Team Agencies**. Only the agencies for your team are listed.

View Your Agencies

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Agencies** subtab.
3. Click **View My Agencies**. Only the agencies that belong to you are listed.

Edit an Agency

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Agencies** subtab.
3. Click the agency name or click  in the row for the agency.
4. Modify the information in the **Name** field.
5. Enter information in the **Legal Entity Name** field.

   The legal entity name is used for proposal PDF file exports. It is used when the name needs to contain different characters or a longer, more formal version of the agency name used in other areas of SAS Intelligent Advertising for Publishers.
6. In the **Credit Limit $** field, specify a monetary limit that if exceeded creates an alert or prevents the booking of a proposal.

   Note: Only Finance and Administrator users can modify the **Credit Limit $** field.
7. If you want to require prepayment from the agency, select **Agency must pre-pay**.
8. In the **Standard discount (%)** field, enter a value that will be applied to exported proposal PDF files.

   Note: Only Finance and Administrator users can modify the **Standard discount (%)** field.
9. In the **Commission (%)** field, enter a value that will be applied to exported proposal PDF files.

   Note: Only Finance and Administrator users can modify the **Commission (%)** field.
10. Enter an alphanumeric value in the **External ID** field.
11. Select an item from the **Region** list.
12. Enter a value in the **Account number** field.
13. In the **Terms and Conditions** field, enter the first few letters of a name for terms and conditions and then select terms and conditions from the autosearch results.

    Note: Enter % if you want to display the full list of term and conditions.

    The selected terms and conditions are used by default when the agency is selected in a proposal.

14. Enter values in the **Identification number** fields.

    Note: For example, enter a tax identification number, business number, or company number.

15. Select a customer status from the list.
16. In the **Category** panel, enter information in the **New category** field and then click **Add**. For more information, see “About Advertisers” on page 159.
17. To add a contact for the agency, click **Add Contact**, enter information and make selections in the Add New Contact window, and then click **Add Contact**.
18. Click **Update**.

Delete an Agency

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Agencies** subtab.
3. In the row for an agency click  and then click **OK**.

Chapter 39

Managing Publishers

About Publishers

A publisher owns the website where advertisements are placed. You can create a publisher in SAS Intelligent Advertising for Publishers if you want to limit reporting access. Creating publishers is useful in a network environment where you want to limit reporting access for publishers in some use cases.

Add a Publisher

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Publishers** subtab.
3. Click + **Add New Publisher**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), spaces, underscores, and hyphens.
5. In the **Sites** field, enter the first few letters of a site and then select the site from the autosearch results.

   Note: Enter % if you want to display the full list of sites.
6. Click **Add Publisher**.

View a Publisher

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Publishers** subtab.
3. Click the publisher name.

View Publishers for Your Team

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Publishers** subtab.
3. Click **View Team Publishers**. Only the publishers for your team are listed.

View Your Publishers

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Publishers** subtab.
3. Click **View My Publishers**. Only the publishers that belong to you are listed.

Edit a Publisher

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Publishers** subtab.
3. Click the publisher name or click  in the row for the publisher.
4. Modify the information in the **Name** field.
5. In the **Sites** field, enter the first few letters of a site and then select the site from the autosearch results.

   Note: Enter % if you want to display the full list of sites.

   To delete existing sites in the **Sites** field, click the X next to the site name
6. Click **Update**.

Delete a Publisher

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Publishers** subtab.
3. In the row for a publisher, click  and then click **OK**.

Chapter 40

Managing Industries

About Industries

You can add as many industries as you like to represent your core interests and associations.

Add an Industry

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Industries** subtab.
3. Click **+ Add New Industry**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.,?;:).

5. Click **Add Industry**.

Edit an Industry

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Industries** subtab.
3. Click the industry name or click in the row for the industry.

4. Modify the information in the **Name** field.
5. Click **Update**.

Delete an Industry

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Industries** subtab.
3. In the row for an industry, click and then click **OK**.

Chapter 41

Managing Regions

About Regions

Regions are attributes that you can assign to advertisers and agencies for reporting.

Add a Region

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Regions** subtab.
3. Click **+ Add New Region**.
4. Enter information in the **Name** field.

   Note: You can enter only alphanumeric characters (a–z, A–Z, 0–9), special characters (~@#$%^&*-+(){}[]|/<>"=']), spaces, underscores, and punctuation marks (!.,?;:).
5. Click **Add Region**.

Edit a Region

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Regions** subtab.
3. Click the region name or click in the row for the region.

4. Modify the information in the **Name** field.
5. Click **Update**.

Delete a Region

1. On the top navigation bar, click the **Customers** tab.
2. Click the **Regions** subtab.
3. In the row for a region, click  and then click **OK**.

Part 8

Ad Partners

Chapter 42

Managing Partners

About Partners

Partners are third parties that can communicate directly with ad engines within SAS Intelligent Advertising for Publishers in order to provide real-time bidding during ad serving. SAS Intelligent Advertising for Publishers supports PubMatic, Index, Rubicon, and AppNexus as partners. Partners are distinct from customers. They are not related to advertisers, agencies, and so on.

You can use the Partners section to set the site and size mappings that will define how to map items in SAS Intelligent Advertising for Publishers to items from the partner.

Tier floor prices define the minimum acceptable bid response from the partner. Targets define which requests are eligible for bids from partners. Define tier floor prices and targets only when not using placements.

Note: You can set up bid requests for PubMatic and Index using the Partners section without using placement.

Modify Settings for PubMatic

1. On the top navigation bar, click the **Partners** tab.
2. Click the **PubMatic** subtab.
3. Click the **Size Mappings** tab.
4. Enter a network size in each size field as needed.
5. Click **Save Sizes**.

6. Click the **Tier Floor Prices** tab.
7. Enter a bid floor price per size per tier in the fields as needed. Tiers are ordered by their ranks.
8. Click **Save Tiers**.
9. Click the **Target** tab.
10. If you want to change the existing target, click the X next to the target name, enter the first few letters of another target, select the target from the autosearch results, and then click **Save Target**.
11. Click the **Site Mappings** tab.
12. Enter a site in each site field as needed, and then click **Save Sites**.

Modify Settings for Index

1. On the top navigation bar, click the **Partners** tab.
2. Click the **Index** subtab.
3. Click the **Size Mappings** tab.
4. Enter a network size in each size field as needed.
5. Click **Save Sizes**.
6. Click the **Tier Floor Prices** tab.
7. Enter a bid floor price per size per tier in the fields as needed. Tiers are ordered by their ranks.
8. Click **Save Tiers**.
9. Click the **Target** tab.
10. If you want to change the existing target, click the X next to the target name, enter the first few letters of another target, select the target from the autosearch results, and then click **Save Target**.
11. Click the **Site Mappings** tab.
12. Enter a site in each site field as needed, and then click **Save Sites**.

Modify Settings for Rubicon

1. On the top navigation bar, click the **Partners** tab.
2. Click the **Rubicon** subtab.
3. Click the **Size Mappings** tab.
4. Enter a network size in each size field as needed.
5. Click **Save Sizes**.
6. Click the **Target** tab.

7. If you want to change the existing target, click the X next to the target name, enter the first few letters of another target, select the target from the autosearch results, and then click **Save Target**.
8. Click the **Site Mappings** tab.
9. Enter a site in each site field as needed, and then click **Save Sites**.

   Note: Rubicon also requires the use of placements. For more information, see "About Placements" on page 101.

Modify Settings for AppNexus

1. On the top navigation bar, click the **Partners** tab.
2. Click the **AppNexus** subtab.
3. Click the **Size Mappings** tab.
4. Enter a network size in each size field as needed.
5. Click **Save Sizes**.
6. Click the **Target** tab.
7. If you want to change the existing target, click the X next to the target name, enter the first few letters of another target, select the target from the autosearch results, and then click **Save Target**.
8. Click the **Site Mappings** tab.
9. Enter a site in each site field as needed, and then click **Save Sites**.

Part 9

Reports

Chapter 43

Reports

About the Reports Tab

On the **Reports** tab, you can run Business Intelligence (BI) reports and operational reports.

About BI Reports

The Business Intelligence (BI) reporting system is a powerful and extensive report builder with access to a large configurable data set that can help analyze historical and future (projected) data. The reporting tool is called the Analyzer. Unlike operational reports that contain near real-time data, BI data is a day behind.

Open a BI Report

1. On the top navigation bar, click the **Reports** tab.
2. Click **BI Reports**.
3. Click .

4. Click a report name.

After you select a report, the main report window appears. The column on the right contains a list of available fields that can be used as columns or filters in the report. The fields are broken down by section. Fields contain campaign and flight information and tags that have been set up for BI logging. Fields with a number in parentheses indicate additional properties that can be displayed after the field is added to the report. The **Measures** section contains count data fields such as **Impressions Served**.

Note: Always use the **Save As** button to rename reports that include "_empty" in the filename.

Data Cubes for BI Reports

Each report uses a specific data cube that specifies what data is accessible to the Analyzer. The data cubes are accessible through the following "empty" reports that all other reports are based on:

Bigsum_empty
: contains historical logging data down to the creative level. Tags and geotargeting tags can also be included in the cube. If a tag needs to be added to this cube, contact SAS Technical Support.

flights_empty
: contains non-delivery flight informational data. There is no date data or tag data in the cube. The cube can be used to report on flight attributes like impression goals, start and end dates, and so on.

Inventory_empty
: contains simulated (projected) data for future dates. Like the Bigsum_empty cube, it also contains data down to the creative level and can include tags. By default, dates up to three months in the future can be selected.

Mediumsum_empty
: contains historical data down to the creative level, along with site and area tags. This is a much smaller cube and generally has a much faster processing time than the Bigsum_empty cube. This cube should be used for most historical reporting.

Pace_empty
: although it is similar to the Mediumsum_empty cube, this cube also contains time of day data. This means that delivery by time of day down to the hour can be seen for individual flights.

Workflow_empty
: unlike the other data cubes, this one does not contain any delivery data. It is used to access the sales workflow data to construct reports showing proposals and line items.

Filtering BI Reports

Note: To minimize the run time of the report, you should avoid selecting **contains** or **Match a specific string** as filters.

Filters restrict the data set to display the desired information. The fields on the right can be used as filters. Click + next to **No Filter in use**. You can drag fields such as the **Date** field into this expanded area. After you drag the **Date** field, a dialog box appears in which you can filter the report by date attributes.

Dragging a standard field to the filter area enables you to select explicit values. You can select multiple values in a row by clicking a value, holding down the Shift key, and clicking the last value. You can also select multiple values by clicking a value, holding down the Ctrl key, and then clicking each desired value. After you select the values, click > to move them to the selected values box to the right. If you need to select all the values, click >>. To move the values back to the deselected box, click < and <<. By default, values moved to the right box are treated as "included." This means that the report displays only the data that contains these values. You can also select **Excluded** in the menu next to **Currently** to display only the data without these values in the report.

Use the option buttons at the top of the dialog box to determine where individual values are selected (**Select from a list**). You can also use the option buttons to select values that are based on a string (**Match a specific string**). If you select **Match a specific string**, then entering `2012` filters on all dates that contain `2012`. You can use most fields as filters, but it is recommended that you use the **Date** field instead of the **Month** field to filter dates. After you select the values, click **OK** to activate the filter.

Building a BI Report

After the desired filters are set up, you drag fields to the main report area as columns. Measures can generally be placed only as columns on the right. Fields can also be placed above the Measure columns. If you need to undo a change to the report, click the blue left arrow in the upper right part of the BI screen. This is also useful when you are adding a column or filter results in an error. Changes can be redone by clicking the blue right arrow. Properties contain additional data related to a field. Fields that have a number inside parentheses in the field list indicate that there are available properties. For example, you can right-click the **Advertiser** field header to open the **Show Properties** menu. Properties that can be enabled appear as additional columns with a light yellow background. Unlike fields, properties cannot be used as filters and cannot be rearranged in the column order. You can right-click each column to open a menu with additional options. Fields and measures will have different options available.

Generating Graphs and Charts in a BI Report

BI also generates graphs and charts from report data. Click  next to **View As**. Measures are always placed on the vertical axis. Regular fields are placed on the horizontal axis. Filters and fields can still be added and modified while you are in the graph mode. To switch back to the regular view, select the **Table** icon next to .

Exporting, Mailing, and Scheduling BI Reports

To export reports, click **More** in the upper right part of the BI window. The **Export Report** submenu enables conversion to PDF, Excel, and CSV files.

You can email reports after you have saved them. Be sure to click  if you open a "_empty" report first. Click  at the top of the screen, select the report, file type, and then enter the email address.

To schedule BI reports, set up the fields that you would use to email the report, enter **Daily**, **Weekly**, or **Monthly** in the scheduling options, and then enter the time at which you want the report to start running.

Note: Scheduling multiple reports to run simultaneously can result in some or all reports failing to complete running or failing to be emailed. Previously scheduled reports can be accessed by clicking **Manage Schedules**.

User-Defined Fields

User-defined fields are custom calculations that can be created and defined in the Analyzer. Right-click a measure (for example, **Impressions Served**) and then select the **User Defined Number** submenu. In this submenu, you can select the % of, Rank, and Running Sum, which are predefined calculations. For example, you can create a column that shows the numerical rank of each row's Impressions Served by selecting **Rank by Impressions Served**. After you click **Next**, change the Name of the column, number format, and decimal places, and then click **Done**. The new column is created and placed in the Analyzer view.

In the **User Defined Number** submenu, select **Calculated Number** to create custom calculations. Available measures are listed in the left section, and the actual calculation is displayed in the right section. Each measure is enclosed in brackets when it is placed in the calculation. A measure can be placed in the calculation by selecting it and pressing the > button, or by manually entering the calculation.

About the Operational Reports

Operational Reports include any report that is not part of the Business Intelligence (BI) suite. You access the operational reports on the **Reports** tab. The operational reports provide a fast and easy way of looking at historical data, and can provide nearly real-time data (delayed by a maximum of six minutes) unlike BI, which can only report up to the preceding day. Users with full Administrator access can view all reports and all data, but users with more restricted roles see only a subset of that data. For example, a user assigned the Agency role sees only data for flights that are assigned to their Agency.

All operational reports can be scheduled to run at a specified time or interval using the **Schedule** section on each report's page. All reports can also be exported to CSV and XLSX format.

Large (>25 MB) operational reports are stored remotely and a link to the report is sent via email. The link is active for 14 days. Smaller reports are sent as email attachments.

On the Reports page, you can run the following types of operational reports:

Advertiser's
: provides a general overview of what advertisers, campaigns, flights, and creatives served within the specified criteria (report options). Impressions, clicks, actions, click rate, and action rate are the displayed metrics.

Area Size
: displays impressions, clicks, and actions for area values and each creative size that served under each area.

Billing
: this large report lists advertisers, campaigns, and flights that were active during the specified date range. Metrics shown are start and end dates, impressions, clicks, actions, impression goals (including bonus), click goals (including bonus), action goals (including bonus), flat rate revenue, and calculated revenues. Note that the displayed flat rate is not calculated. Instead, the full flat rate amount set in the flight is displayed.

Custom Action
: the only report that displays custom action activity. The name of the custom action, attribution (impression or click), and the number of actions are displayed for advertisers, sites, areas, flights, and creatives.

Device/Geo
: displays historical impression, click, and action data for mobile device and geo tags. The report can be filtered by Advertiser, Agency, Campaign Name, Flight Name, Site, Area, or Size. Device and geo tags are selected in the **Options** field. To see all available options, enter `%`. The output also includes an interactive chart that works in a similar way to the Dashboards. Note that device and geo tags must be logged in Bigsum to be used in this report.

Flight Status
: displays flights filtered by status. This report can be used to help determine how flights are performing. Advertisers, campaigns, and flights are displayed along with the tier, products and targets, and start and end dates that belong to those flights. Realized amounts and goals are displayed for impressions, clicks, and actions. Note that the site filter that can be applied to the report filters only against flights that are explicitly targeted to the specified site when you are using the **Sites & Areas** section on the Flight window. If a flight is targeted to a site through other means, such as an actual target, those flights are not included in the report results if a site is specified in the criteria. Along with the above, On Schedule Percentage is displayed. The On Schedule Percentage value indicates how close, percentage-wise, a flight is to serving its ideal daily amount. On Schedule Percentage is the only value in the report that should be used to determine performance of a flight.

Performance
: provides a general overview of what advertisers, campaigns, flights, and creatives served within the specified criteria (report options). Impressions, views, clicks, actions, click rate, and action rate are the displayed metrics. Unlike the Advertiser's and Publisher's reports, the Performance report can be filtered to specific tiers, campaigns, flights, and creatives.

Product
: the only report that displays future, or projected, data. The results of the report show products and the flights that could serve to the demographic in the product. Projected impressions are displayed along with available inventory amounts.

Publisher’s
: provides a general overview of what advertisers, campaigns, flights, and creatives were served within the specified criteria (report options). It is exactly the same as the Advertiser’s report with the exception of revenue.

Site Area
: displays flight impressions, clicks, and actions for site values as well as each area value that served with the site.

Size Served
: displays impressions, clicks, and actions for each creative size that served during the specified date range.

Tags
: displays impressions, clicks, and actions for the tag values under the specified tag. The value **`OTHER`** includes all impressions that did not include the specified tag in the ad call, or included a value for the tag that does not exist. If no data for a tag exists, make sure that the **Tagsum logging** option is enabled by editing the tag in SAS Intelligent Advertising for Publishers.

Tag Assessment
: displays a list of tag values associated with the selected flight type to help determine the most productive tag and tag combinations. The report computes the value of every tag that is used in targeting in active and pending flights. The tags are ordered by value and display the % of highest tag value for all other tags. The highest value tag is 100%.

Third Party Data
: third-party data can be provided to SAS Intelligent Advertising for Publishers, which can be displayed using this report.

Time of Day
: provides a 24-hour breakdown of impressions, clicks, and actions for the specified date range. Note that the site filter that can be applied to the report filters against only flights that are explicitly targeted to the specified site using the **Sites & Areas** section on the Flight window. If a flight is targeted to a site through other means, like an actual target, those flights are not included in the report results if a site is specified in the criteria.

Unique Visitor
: lists flights along with unique impression, click, and action amounts. These amounts reflect the number of unique visitors that were served the flight, or submitted a click or action for the flight. You use the Option criteria to apply one of the following levels of uniqueness: unique for a 24-hour period, unique for a month, or unique in the entirety of the record. You can specify criteria for the report in the **Advertiser**, **Campaigns**, **Flights**, **Areas**, **Emails**, and **Subject** fields. The information in the **Subject** field is used for emailed or scheduled reports. Select the **Include views** check box to add a column that displays counts for viewed ads.

Workflow Revenue
: displays Proposal and Line Item data for the specified criteria. The Date Range selection filters based on the Proposal start date, and Proposals without a start date are included regardless of the specified date range. Proposal Advertiser, Agency, Bill To, Salesperson, External ID, and start and end dates are included. Product, Product Line, and Billable Amount data is displayed for each Line Item. The CSV and XLSX exports include a breakdown of Billable Amount by month.

Run a Report

1. On the top navigation bar, click the **Reports** tab.
2. Click a *Report Name* on the dashboard or select a report from the **Reports** list.
3. In the report window, enter the desired filters and then click **Run Report**.

Export a Report

1. On the top navigation bar, click the **Reports** tab.
2. Click a *Report Name* on the dashboard.

3. In the report window, enter the desired filters and then click **Export xslx** to export the report as an Excel file or click **Export csv** to export the file as a comma-separated values (CSV) file.

Schedule a Report

1. On the top navigation bar, click the **Reports** tab.
2. Click a *Report Name* on the dashboard.
3. In the report window, enter the desired filters.
4. Expand the **Schedule** panel and then do the following:
   a. Select **Daily**, **Weekly**, or **Monthly**.

      If you select **Daily**, select a value from the **Run at Hour** list.

      If you select **Weekly** or **Monthly**, select a value from the **Run on Day** and **Run at Hour** lists.

   b. Enter a time value in the **Expire in** field.
5. Click **Schedule Report**.

   To modify existing schedules for reports, click **Manage Schedules** and then make desired changes.

Part 10

Appendixes

Appendix 1
XML API Examples

About the REST API for SAS Intelligent Advertising for Publishers

REST (REpresentational State Transfer) is a distributed programming architectural style in which the World Wide Web is treated generally as a large distributed application. In this style, the application is distributed as structured data rather than by explicitly invoking remote procedure calls on remote objects. Each data element is a resource addressed by a URI (Uniform Resource Identifier), which might be provided in multiple representations.

The REST application programming interface (API) that is used by SAS Intelligent Advertising for Publishers is a programming-language-independent API. It is deployed in a network environment using HTTP or HTTPS transport using HTTP methods (GET, POST, DELETE, PUT, and PATCH) to send and receive data and application state (using XML representations) between the client and the API's service URLs. A number of API or integration points have been developed to support this approach:

creative templates
: The creative template store for SAS Intelligent Advertising for Publishers provides a method to build and rapidly deploy custom creative templates. SAS Intelligent Advertising for Publishers customers can build unique formats or customized versions of extensible industry standards, like Video Ad Serving Template (VAST), without the need for development support from SAS Intelligent Advertising for Publishers. A comprehensive token replacement library within the creative format management system easily allows templates to incorporate data passed in on the ad request.

business process API
: A number of APIs are provided to enable customers to integrate SAS Intelligent Advertising for Publishers with their existing business systems (such as billing or data warehousing). These provide customers with the ability to pass campaign information between systems without re-keying.

creative APIs
: The recommended approach to set up video creative in the system uses the creative APIs and enables the transcoding service to build the creative element within SAS Intelligent Advertising for Publishers and inform the trafficker once the process is complete. SAS Intelligent Advertising for Publishers works with partners who have successfully implemented this process for other customers.

server-to-server connection
: Although not strictly an API, SAS Intelligent Advertising for Publishers features a unique creative type—the network creative that, when configured, permits the application to make a server-to-server request for creative instruction from another supplier's system and removes the need for subsequent HTTP requests to be made from the user's browser or device. SAS Intelligent Advertising for Publishers development currently works with customers to ensure compatibility with any relevant third-parties. The customer would be able to use this feature if required.

Authenticate an API Request

You authenticate an API request using one of the following methods:

- obtain your API Key from your account page. Next, add **`?api_key=<API_KEY>`** to the end of the API URL that you are requesting.

  ```
  https://beta.aimatch.com/advertisers.xml?api_key=API_KEY
  ```

- use your login credentials (login and password) in a standard Basic HTTP authentication header

  ```
  Authorization: Basic dGVzdDpwYXNz
  ```

- use your API Key in a standard Basic HTTP authentication header. In this method, the password used in the authentication must be blank.

  ```
  Authorization: Basic dHY4NXJCY1VvRGVGYkQwWTE2ZTE6
  ```

Requesting Access to the XML API

The URLs for the API comply with standard REST conventions and is the same URL structure that is used to access the HTML interface. The only difference is that you must inform the server that you want to access the XML data. You request access to the XML data from the server in one of the following ways:

- add *.xml* to the URL
- add an Accept HTTP Header of *application/xml* to your request

```
Accept: application/xml
```

If you are creating or updating an entity, you must submit XML content in the body of your request. In these cases, make sure that you always set the Content-Type header to **`application/xml`**.

```
Content-Type application/xml
```

Standard Responses

Errors on Create and Update API Calls

Returns with a status code of 422 and the following typical XML structure:

```
<errors>
<error>Name has already been taken</error>
</errors>
```

Conflicts

If you try to update an existing record and the lock version is out of date with the current lock version, then a 409 status code is returned with no content.

About the Rules for Nesting Items

Creating Nested Items with Other Items in a Single Request

To create an item with nested items, such as areas associated with sites, the nested items are specified by the nested item name, singular or plural, depending on the relationship, followed by a hyphen and **`attributes`**. The attribute type with value **`array`** informs the server to process child elements as an array so that more than one item can be related to another, such as many sites for one area.

This example creates an item with nested items for an area with many sites:

```
<?xml version="1.0" encoding="UTF-8"?>
<area>
  <name>new_area</name>
```

```
  <sites-attributes type="array">
    <site>
      <description nil="true">Example of a site nested in an area</description>
      <name>new_area_first_site</name>
      <revenue-cut type="decimal">3.25</revenue-cut>
    </site>
    <site>
      <description nil="true"></description>
      <name>new_area_second_site</name>
      <revenue-cut type="decimal">7.25</revenue-cut>
    </site>
    <site>
      <description nil="true"></description>
      <name>new_area_third_site</name>
      <revenue-cut type="decimal">5.25</revenue-cut>
    </site>
  </sites-attributes>
</area>
```

Here is an example of creating an item with nested items for a flight that has many targets and has many flight-creatives. In this example, flight-creatives have only one target:

```
<?xml version="1.0" encoding="UTF-8"?>
<flight>
  <campaign-id>1</campaign-id>
  <name>new_flight</name>
  <targets-attributes type="array">
    <target>
    ...
    </target>
  </targets-attributes>
  <flight-creatives-attributes type="array">
    <flight-creative>
      <creative-id>1</creative-id>
      ...
      <target-attributes>
      ...
      </target-attributes>
      ...
    </flight-creative>
  </flight-creatives-attributes>
  ...
</flight>
```

Updating Nested Items with an Item in a Single Request

When you update an item with nested items, the nested item's ID must be present or an attempt to create a new item is made.

Here is an example of updating an area and its sites, and creating a new site that is also associated with the area:

```
PUT: https://beta.sas.com/areas/1.xml


<?xml version="1.0" encoding="UTF-8"?>
<area>
```

```
    <name>updated_area</name>
    <sites-attributes type="array">
      <site>
        <id type="integer">1</id>
        <description nil="true">Example of a site nested in an area updated</description>
        <name>updated_area_first_site</name>
        <revenue-cut type="decimal">5.25</revenue-cut>
      </site>
      <site>
        <id type="integer">2</id>
        <name>updated_area_second_site</name>
        <revenue-cut type="decimal">10.25</revenue-cut>
      </site>
      <site>
        <id type="integer">3</id>
        <description nil="true"></description>
        <name>updated_area_third_site</name>
        <revenue-cut type="decimal">6.25</revenue-cut>
      </site>
      <site>
        <description nil="true">new fourth site for area 1</description>
        <name>updated_area_fourth_site</name>
        <revenue-cut type="decimal">2.0</revenue-cut>
      </site>
    </sites-attributes>
  </area>
```

About Search Filtering

The XML REST API supports the search filtering that is used by SAS Intelligent Advertising for Publishers. To use a search filter, add the search parameter to the url.

Here is an example of a list of campaigns filtered by an advertiser name and ordered by campaign name descending:

```
GET: https://beta.aimatch.com/campaigns.xml?search%5Badvertiser_name_equals%5D=My+Advert
```

Here is an example of a list of flights filtered using campaign advertiser name:

```
GET: https://beta.aimatch.com/campaigns.xml?search%5Badvertiser_name_equals%5D=My+Advert
```

Note: A flight belongs to a campaign and a campaign belongs to an advertiser.

Here is an example of a list of campaigns filtered by end date before or on a certain date:

```
GET: https://beta.aimatch.com/campaigns.xml?search%5Bend_date_lte%5D=2012-01-01
```

Here is an example of getting the list of tags containing tag values with value:

```
GET: https://beta.aimatch.com/tags.xml?search%5Bvalues_value_like%5D=myval
```

Using the XML REST API to Manage the Ad Selection Entities

You use the XML REST API to list, display, create, and update the following ad selection trafficking entities:

- advertiser creative templates
- advertisers
- areas
- campaigns
- creatives
- creative formats
- flights
- media
- sites
- tags
- targets

Managing Advertisers

Get a List of Advertisers

Here is an example of getting a list of advertisers:

```
GET: https://beta.aimatch.com/advertisers.xml
```

Here is the XML response:

```
<advertisers type="array">
  <advertiser>
    <id type="integer">1</id>
    <name>TestAdvertiser1</name>
  </advertiser>
  <advertiser>
    <id type="integer">2</id>
    <name>TestAdvertiser2</name>
  </advertiser>
  <advertiser>
    <id type="integer">3</id>
    <name>TestAdvertiser3</name>
  </advertiser>
</advertisers>
```

This XML response provides 30 advertisers at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL.

Here is an example of a URL that includes a page query parameter:

```
https://beta.aimatch.com/advertisers.xml?page=3
```

Get a Specific Advertiser

Here is an example of getting a specific advertiser:

```
GET: https://beta.aimatch.com/advertisers/3.xml
```

Here is the XML response:


```
<advertiser>
  <legal-entity-name nil="true"/>
  <status nil="true"/>
  <id type="integer">3</id>
  <name>TestAdvertiser3</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal" nil="true"></credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status nil="true"/>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-02-02T18:09:11-05:00</updated-at>
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean" nil="true"></pre-pay>
  <created-at type="datetime">2011-02-02T18:09:11-05:00</created-at>
  <standard-discount type="decimal" nil="true"></standard-discount>
  <region-id type="integer" nil="true"/>
  <contacts type="array"/>
  <category-list type="array"/>
</advertiser>
```


Get a Specific Advertiser Using Name Instead of ID

Here is an example of getting a specific advertiser using a name instead of an ID:

```
GET: https://beta.aimatch.com/advertisers/
by_name.xml?advertiser%5Bname%5D=TestAdvertiser3
```

Here is the XML response:


```
<advertiser>
  <legal-entity-name nil="true"/>
  <status nil="true"/>
  <id type="integer">3</id>
  <name>TestAdvertiser3</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal" nil="true"></credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status nil="true"/>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-02-02T18:09:11-05:00</updated-at>
```

```
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean" nil="true"></pre-pay>
  <created-at type="datetime">2011-02-02T18:09:11-05:00</created-at>
  <standard-discount type="decimal" nil="true"></standard-discount>
  <region-id type="integer" nil="true"/>
  <contacts type="array"/>
  <category-list type="array"/>
</advertiser>
```


Here is an example of getting a specific advertiser using Name instead of ID with categories:

```
GET: https://beta.aimatch.com/advertisers/1.xml
```

Here is the XML response:


```
<advertiser>
  <legal-entity-name nil="true"/>
  <status nil="true"/>
  <id type="integer">1</id>
  <name>TestAdvertiser1</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal">5000000.0</credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status nil="true"/>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-02-16T20:19:53-05:00</updated-at>
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean">false</pre-pay>
  <created-at type="datetime">2011-02-16T20:19:53-05:00</created-at>
  <standard-discount type="decimal">1.5</standard-discount>
  <region-id type="integer" nil="true"/>
  <contacts type="array"/>
  <category-list type="array">
    <category>danadv1cat1</category>
    <category>danadv1cat2</category>
  </category-list>
</advertiser>
```


Here is an example of getting a specific advertiser using Name instead of ID with contacts:

```
GET: https://beta.aimatch.com/advertisers/2.xml
```

Here is the XML response:


```
<advertiser>
  <legal-entity-name nil="true"/>
  <status nil="true"/>
  <id type="integer">2</id>
  <name>TestAdvertiser2</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal" nil="true"></credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status nil="true"/>
```

```
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-02-18T17:06:41-05:00</updated-at>
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean">false</pre-pay>
  <created-at type="datetime">2011-02-18T17:06:41-05:00</created-at>
  <standard-discount type="decimal">1.5</standard-discount>
  <region-id type="integer" nil="true"/>
  <contacts type="array">
    <contact>
      <address1>123 x</address1>
      <address2>suite 123</address2>
      <city>Raleigh</city>
      <contactable-id type="integer">2</contactable-id>
      <contactable-type>Advertiser</contactable-type>
      <country>USA</country>
      <created-at type="datetime">2011-02-18T17:06:41-05:00</created-at>
      <deleted-at type="datetime" nil="true"></deleted-at>
      <email>xmldanadv_12@blah.com</email>
      <full-name>dan adv_12</full-name>
      <id type="integer">11</id>
      <lock-version type="integer">0</lock-version>
      <phone>919-311-1234</phone>
      <state>NC</state>
      <title>CEO</title>
      <updated-at type="datetime">2011-02-18T17:06:41-05:00</updated-at>
      <website>http://danadv.blah.net</website>
      <zip>23456</zip>
    </contact>
  </contacts>
  <category-list type="array">
    <category>xmlcat1</category>
    <category>xmlcat2</category>
  </category-list>
</advertiser>
```

Create an Advertiser

Here is an example of creating an advertiser:

```
POST: https://beta.aimatch.com/advertisers.xml
```

```
<advertiser>
  <name>My-New-Advertiser</name>
  <category-list type="array">
    <category>Clothing</category>
    <category>Children</category>
  </category-list>
</advertiser>
```

Response (HTTP status: 201 Created):

```
<advertiser>
  <legal-entity-name nil="true"/>
  <status nil="true"/>
  <id type="integer">4</id>
  <name>My-New-Advertiser</name>
```

```
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal" nil="true"></credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status nil="true"/>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-03-02T18:06:30-05:00</updated-at>
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean" nil="true"></pre-pay>
  <created-at type="datetime">2011-03-02T18:06:30-05:00</created-at>
  <standard-discount type="decimal" nil="true"></standard-discount>
  <region-id type="integer" nil="true"/>
  <contacts type="array"/>
  <category-list type="array">
    <category>Clothing</category>
    <category>Children</category>
  </category-list>
</advertiser>
```

Update an Advertiser

Here is an example of updating an advertiser:

```
PUT: https://beta.aimatch.com/advertisers/14.xml

<advertiser>
  <credit-limit type="decimal" nil="true">50000</credit-limit>
  <lock-version type="integer">0</lock-version>
  <name>TestAdvertiser3</name>
  <category-list type="array">
    <category>Foo</category>
    <category>Bar</category>
  </category-list>
</advertiser>
```

The response will not contain anything, but the status code will be 200.

Update an Advertiser Using the Name Instead of the ID

Note: The advertiser’s name cannot be updated when you use this URL.

Here is an example of updating an advertiser using the name instead of the ID:

```
PUT: https://beta.aimatch.com/advertisers/by_name.xml

<advertiser>
  <name>TestAdvertiser3</name>
  <credit-limit type="decimal">60000</credit-limit>
  <lock-version type="integer">1</lock-version>
  <standard-discount type="decimal">2.25</standard-discount>
</advertiser>
```

Specifying the Category List

You can specify a category list in one of the following ways:

- by using a list of elements

```
<advertiser>
  <name>TestAdvertiser3</name>
  <category-list type="array">
    <category>Foo</category>
    <category>Bar</category>
  </category-list>
</advertiser>
```

- by using a comma separated string list

```
<advertiser>
  <name>TestAdvertiser3</name>
  <category-list>Foo, Bar</category-list>
</advertiser>
```

Managing Areas

Get a List of Areas

Here is an example of getting a list of areas:

```
GET: https://beta.aimatch.com/areas.xml
```

Here is the XML response:

```
<areas type="array">
  <area>
    <id type="integer">4</id>
    <name>TestArea0</name>
  </area>
  <area>
    <id type="integer">5</id>
    <name>TestArea1</name>
  </area>
  <area>
    <id type="integer">14</id>
    <name>TestArea10</name>
  </area>
</areas>
```

This XML response provides 30 areas at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/areas.xml?page=3
```

Get a Specific Area

Here is an example of getting a specific area:

```
GET: https://beta.aimatch.com/areas/14.xml
```

Here is the XML response:

```
<area>
```

```
  <created-at type="datetime">2010-08-31T13:49:39-04:00</created-at>
  <id type="integer">14</id>
  <lock-version type="integer">0</lock-version>
  <name>TestArea10</name>
  <updated-at type="datetime">2010-08-31T13:49:39-04:00</updated-at>
  <sites type="array"/>
</area>
```

Here is an example of getting a specific area with sites:

```
GET: https://beta.aimatch.com/areas/10.xml
```

Here is the XML response:

```
<area>
  <created-at type="datetime">2011-01-20T13:28:59+13:00</created-at>
  <id type="integer">10</id>
  <lock-version type="integer">8</lock-version>
  <name>area10</name>
  <updated-at type="datetime">2011-02-16T04:56:39+13:00</updated-at>
  <sites type="array">
    <site>
      <created-at type="datetime">2011-02-15T05:30:32+13:00</created-at>
      <description nil="true"></description>
      <id type="integer">112</id>
      <lock-version type="integer">10</lock-version>
      <name>site112</name>
      <revenue-cut type="decimal">3.25</revenue-cut>
      <updated-at type="datetime">2011-02-16T07:22:21+13:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-02-15T08:22:03+13:00</created-at>
      <description nil="true"></description>
      <id type="integer">113</id>
      <lock-version type="integer">5</lock-version>
      <name>site113</name>
      <revenue-cut type="decimal">7.25</revenue-cut>
      <updated-at type="datetime">2011-02-16T07:22:21+13:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-02-15T08:27:44+13:00</created-at>
      <description nil="true"></description>
      <id type="integer">114</id>
      <lock-version type="integer">1</lock-version>
      <name>site114</name>
      <revenue-cut type="decimal">5.25</revenue-cut>
      <updated-at type="datetime">2011-02-16T04:56:39+13:00</updated-at>
    </site>
  </sites>
</area>
```

Create an Area That Is Associated with New and Existing Sites

Here is an example of creating a list of areas with new and existing sites:

```
POST: https://beta.aimatch.com/areas.xml
```

```
<?xml version="1.0"?>
```

```
<area>
  <name>new_area_test</name>
  <sites-attributes type="array">
    <site>
      <name>new_area_first_site</name>
      <description>Example of new and existing sites nested in an area</description>
      <revenue-cut>3.25</revenue-cut>
    </site>
    <site>
      <name>new_area_second_site</name>
      <revenue-cut>7.25</revenue-cut>
    </site>
    <site>
      <name>new_area_third_site</name>
      <revenue-cut>5.25</revenue-cut>
    </site>
    <site>
      <id>10</id>
    </site>
    <site>
      <id>22</id>
    </site>
  </sites-attributes>
</area>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<area>
  <created-at type="datetime">2011-03-10T15:21:15-05:00</created-at>
  <id type="integer">47</id>
  <lock-version type="integer">0</lock-version>
  <name>new_area_test</name>
  <updated-at type="datetime">2011-03-10T15:21:15-05:00</updated-at>
  <sites type="array">
    <site>
      <created-at type="datetime">2011-01-19T19:28:59-05:00</created-at>
      <description nil="true"></description>
      <id type="integer">10</id>
      <lock-version type="integer">0</lock-version>
      <name>bruciun.com</name>
      <revenue-cut type="decimal" nil="true"></revenue-cut>
      <updated-at type="datetime">2011-01-19T19:28:59-05:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-01-19T19:28:59-05:00</created-at>
      <description nil="true"></description>
      <id type="integer">22</id>
      <lock-version type="integer">0</lock-version>
      <name>felynnel.com</name>
      <revenue-cut type="decimal" nil="true"></revenue-cut>
      <updated-at type="datetime">2011-01-19T19:28:59-05:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-03-10T15:21:15-05:00</created-at>
      <description>Example of new and existing sites nested in an area</description>
      <id type="integer">131</id>
```

```
      <lock-version type="integer">0</lock-version>
      <name>new_area_second_site</name>
      <revenue-cut type="decimal">3.25</revenue-cut>
      <updated-at type="datetime">2011-03-10T15:21:15-05:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-03-10T15:21:15-05:00</created-at>
      <description nil="true"></description>
      <id type="integer">132</id>
      <lock-version type="integer">0</lock-version>
      <name>new_area_test_n_exist_site2</name>
      <revenue-cut type="decimal">7.25</revenue-cut>
      <updated-at type="datetime">2011-03-10T15:21:15-05:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-03-10T15:21:15-05:00</created-at>
      <description nil="true"></description>
      <id type="integer">133</id>
      <lock-version type="integer">0</lock-version>
      <name>new_area_third_site</name>
      <revenue-cut type="decimal">5.25</revenue-cut>
      <updated-at type="datetime">2011-03-10T15:21:15-05:00</updated-at>
    </site>
  </sites>
</area>
```

Update an Area and Its Associated Sites

Here is an example of updating an area and its associated sites:

```
PUT: https://beta.aimatch.com/areas/119.xml

 <?xml version="1.0"?>
<area>
  <name>updated_area</name>
  <sites-attributes type="array">
    <site>
      <id>119</id>
      <lock-version>0</lock-version>
      <name>updated_area_first_site</name>
      <description>Example of a site nested in an area updated</description>
      <revenue-cut>6.25</revenue-cut>
    </site>
    <site>
      <id>120</id>
      <lock-version>0</lock-version>
      <name>updated_area_second_site</name>
      <revenue-cut>10.5</revenue-cut>
    </site>
    <site>
      <id>121</id>
      <name>updated_area_third_site</name>
      <lock-version>0</lock-version>
    </site>
    <site>
      <name>updated_area_new_fourth_site</name>
```

```
    </site>
  </sites-attributes>
</area>
```

The response will not contain anything, but the status code will be 200.

Specifying the Site-ID List for Existing Sites

You can specify a site-ID list for existing sites in one of the following ways:

- by using a list of elements

```
<area>
  <id>1</id>
  <site-id-list type="array">
    <site-id>1</site-id>
    <site-id>2</site-id>
  </site-id-list>
</area>
```

- by using a comma separated string list

```
<area>
  <id>1</id>
  <site-id-list>1,2</site-id-list>
</area>
```

- by using an alternative form with site-IDs:

```
<area>
  <id>1</id>
  <site-ids type="array">
    <id>119</id>
    <id>121</id>
  </site-ids>
</area>
```

Managing Campaigns

Get a List of Campaigns

Here is an example of getting a list of campaigns:

```
GET: https://beta.aimatch.com/campaigns.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<campaigns type="array">
  <campaign>
    <id type="integer">1</id>
    <name>TestCampaign1</name>
  </campaign>
  <campaign>
    <id type="integer">2</id>
    <name>TestCampaign2</name>
  </campaign>
```

```
  <campaign>
    <id type="integer">3</id>
    <name>TestCampaign3</name>
  </campaign>
</campaigns>
```

This XML response provides 30 campaigns at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/campaigns.xml?page=3
```

Get a Specific Campaign

Here is an example of getting a specific campaign:

```
GET: https://beta.aimatch.com/campaigns/3.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<campaign>
  <campaign-type>Guaranteed</campaign-type>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <description nil="true"></description>
  <end-date type="date" nil="true"></end-date>
  <external-id nil="true"></external-id>
  <flags type="integer">0</flags>
  <id type="integer">1</id>
  <lock-version type="integer">0</lock-version>
  <name>TestCampaign3</name>
  <notes nil="true"></notes>
  <proposal-id type="integer" nil="true"></proposal-id>
  <start-date type="date" nil="true"></start-date>
  <status nil="true"></status>
  <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
  <flights type="array">
    <flight>
      <actions type="integer" nil="true"></actions>
      <bonus-actions type="integer" nil="true"></bonus-actions>
      <bonus-clicks type="integer" nil="true"></bonus-clicks>
      <bonus-impressions type="integer" nil="true"></bonus-impressions>
      <campaign-id type="integer">1</campaign-id>
      <clicks type="integer" nil="true"></clicks>
      <companion-type></companion-type>
      <cost-actions type="decimal" nil="true"></cost-actions>
      <cost-actions-local type="decimal" nil="true"></cost-actions-local>
      <cost-clicks type="decimal" nil="true"></cost-clicks>
      <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
      <cost-impressions type="decimal" nil="true"></cost-impressions>
      <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
      <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
      <creative-selection-method>Weighted</creative-selection-method>
      <currency-id type="integer">840</currency-id>
      <conversion-rate type="float">1.0</conversion-rate>
      <daily-end-time type="datetime" nil="true"></daily-end-time>
      <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
```

```
<daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
<daily-start-time type="datetime" nil="true"></daily-start-time>
<description></description>
<end-date type="datetime">2011-06-25T09:00:00-04:00</end-date>
<flat-rate-local type="decimal" nil="true"></flat-rate-local>
<flat-rate type="decimal" nil="true"></flat-rate>
<flat-rate-flag type="boolean">false</flat-rate-flag>
<flight-type>House</flight-type>
<id type="integer">1</id>
<impressions type="integer" nil="true"></impressions>
<lock-version type="integer">1</lock-version>
<name>campaign_3_flight_1</name>
<notes></notes>
<reference-rate type="integer" nil="true"></reference-rate>
<size-id type="integer">0</size-id>
<start-date type="datetime">2011-05-09T09:00:00-04:00</start-date>
<status>pending</status>
<targets-group-logic></targets-group-logic>
<updated-at type="datetime">2011-05-11T15:46:54-04:00</updated-at>
<user-info></user-info>
<weight type="integer">1</weight>
<actual-impressions type="integer">22079581</actual-impressions>
<actual-clicks type="integer">0</actual-clicks>
<actual-actions type="integer">0</actual-actions>
<actual-custom-actions type="integer">0</actual-custom-actions>
<flight-creatives type="array">
  <flight-creative>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
    <creative-id type="integer">1</creative-id>
    <flight-id type="integer">1</flight-id>
    <from-date type="date" nil="true"></from-date>
    <id type="integer">1</id>
    <position type="integer">1</position>
    <to-date type="date" nil="true"></to-date>
    <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
    <weight type="integer">1</weight>
  </flight-creative>
  <flight-creative>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-05-11T15:31:59-04:00</created-at>
    <creative-id type="integer">2</creative-id>
    <flight-id type="integer">1</flight-id>
    <from-date type="date" nil="true"></from-date>
    <id type="integer">3</id>
    <position type="integer" nil="true"></position>
    <to-date type="date" nil="true"></to-date>
    <updated-at type="datetime">2011-05-11T15:34:51-04:00</updated-at>
    <weight type="integer">1</weight>
  </flight-creative>
</flight-creatives>
<targets type="array"/>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
```

```
    <grace-period type="integer" nil="true"></grace-period>
    <grace-period-priority type="decimal" nil="true"></grace-period-priority>
    <id type="integer">1</id>
    <lock-version type="integer">0</lock-version>
    <name>Direct Sold</name>
    <preemptible type="boolean">false</preemptible>
    <prioritization-method>Schedule</prioritization-method>
    <rank type="integer">1</rank>
    <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
  </tier>
  <areas type="array">
    <area>
      <area-id type="">4</area-id>
      <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
      <flight-id type="">1</flight-id>
      <id type="integer">4</id>
      <lock-version type="integer">0</lock-version>
      <name>Article</name>
      <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
      <sites type="array"/>
    </area>
    <area>
      <area-id type="">8</area-id>
      <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
      <flight-id type="">1</flight-id>
      <id type="integer">8</id>
      <lock-version type="integer">0</lock-version>
      <name>Downloads</name>
      <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
      <sites type="array"/>
    </area>
  </areas>
  <tag-values type="array"/>
  <sites type="array">
    <site>
      <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
      <description nil="true"></description>
      <flight-id type="">1</flight-id>
      <id type="integer">1</id>
      <lock-version type="integer">0</lock-version>
      <name>reviews.cnet.com</name>
      <revenue-cut type="decimal" nil="true"></revenue-cut>
      <site-id type="">1</site-id>
      <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
      <description nil="true"></description>
      <flight-id type="">1</flight-id>
      <id type="integer">5</id>
      <lock-version type="integer">0</lock-version>
      <name>alecopy.com</name>
      <revenue-cut type="decimal" nil="true"></revenue-cut>
      <site-id type="">5</site-id>
      <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
    </site>
```

```
    </sites>
    <flight-action-policies type="array"/>
    <category-list type="array">
      <category>Soda</category>
    </category-list>
    <beacon_count type="integer">0</beacon_count>
    <top_priority type="integer">0</top_priority>
  </flight>
  <flight>
    <actions type="integer" nil="true"></actions>
    <bonus-actions type="integer" nil="true"></bonus-actions>
    <bonus-clicks type="integer" nil="true"></bonus-clicks>
    <bonus-impressions type="integer" nil="true"></bonus-impressions>
    <campaign-id type="integer">1</campaign-id>
    <clicks type="integer" nil="true"></clicks>
    <companion-type nil="true"></companion-type>
    <cost-actions type="decimal" nil="true"></cost-actions>
    <cost-actions-local type="decimal" nil="true"></cost-actions-local>
    <cost-clicks type="decimal" nil="true"></cost-clicks>
    <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
    <cost-impressions type="decimal" nil="true"></cost-impressions>
    <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
    <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
    <creative-selection-method>Weighted</creative-selection-method>
    <currency-id type="integer">840</currency-id>
    <conversion-rate type="float">1.0</conversion-rate>
    <daily-end-time type="datetime" nil="true"></daily-end-time>
    <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
    <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
    <daily-start-time type="datetime" nil="true"></daily-start-time>
    <description nil="true"></description>
    <end-date type="datetime">2011-07-25T09:55:53-04:00</end-date>
    <flat-rate type="decimal" nil="true"></flat-rate>
    <flat-rate-local type="decimal" nil="true"></flat-rate-local>
    <flat-rate-flag type="boolean">false</flat-rate-flag>
    <flight-type>House</flight-type>
    <id type="integer">2</id>
    <impressions type="integer" nil="true"></impressions>
    <lock-version type="integer">0</lock-version>
    <name>campaign_3_flight_2</name>
    <notes nil="true"></notes>
    <reference-rate type="integer" nil="true"></reference-rate>
    <size-id type="integer" nil="true"></size-id>
    <start-date type="datetime">2011-05-02T09:55:53-04:00</start-date>
    <status>pending</status>
    <targets-group-logic nil="true"></targets-group-logic>
    <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
    <user-info nil="true"></user-info>
    <weight type="integer">1</weight>
    <actual-impressions type="integer">9498624</actual-impressions>
    <actual-clicks type="integer">0</actual-clicks>
    <actual-actions type="integer">0</actual-actions>
    <actual-custom-actions type="integer">3410</actual-custom-actions>
    <flight-creatives type="array">
      <flight-creative>
        <active type="boolean">true</active>
```

```
        <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
        <creative-id type="integer">2</creative-id>
        <flight-id type="integer">2</flight-id>
        <from-date type="date" nil="true"></from-date>
        <id type="integer">2</id>
        <position type="integer">1</position>
        <to-date type="date" nil="true"></to-date>
        <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
        <weight type="integer">1</weight>
      </flight-creative>
    </flight-creatives>
    <targets type="array"/>
    <tier>
      <active type="boolean">true</active>
      <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
      <flags type="integer">0</flags>
      <grace-period type="integer">1</grace-period>
      <grace-period-priority type="decimal">0.2</grace-period-priority>
      <id type="integer">2</id>
      <lock-version type="integer">0</lock-version>
      <name>Indirect Sold</name>
      <preemptible type="boolean">false</preemptible>
      <prioritization-method>eCPM</prioritization-method>
      <rank type="integer">2</rank>
      <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
    </tier>
    <areas type="array"/>
    <tag-values type="array"/>
    <sites type="array"/>
    <flight-action-policies type="array"/>
    <category-list type="array">
      <category>Soda</category>
    </category-list>
    <beacon_count type="integer">0</beacon_count>
    <top_priority type="integer">0</top_priority>
  </flight>
</flights>
<advertiser>
  <legal-entity-name></legal-entity-name>
  <status nil="true"/>
  <id type="integer">1</id>
  <name>TestAdvertiser1</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal" nil="true"></credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status>Active</customer-status>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean" nil="true"/>
  <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
  <standard-discount type="decimal" nil="true"/>
  <region-id type="integer" nil="true"/>
```

```
    <contacts type="array">
      <contact>
        <address1 nil="true"></address1>
        <address2 nil="true"></address2>
        <city nil="true"></city>
        <contactable-id type="integer">1</contactable-id>
        <contactable-type>Advertiser</contactable-type>
        <country nil="true"></country>
        <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
        <email nil="true"></email>
        <full-name>Test Advertiser User</full-name>
        <id type="integer">1</id>
        <lock-version type="integer">0</lock-version>
        <phone nil="true"></phone>
        <state nil="true"></state>
        <title nil="true"></title>
        <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
        <website nil="true"></website>
        <zip nil="true"></zip>
      </contact>
    </contacts>
  </advertiser>
</campaign>
```

Create a Campaign with Nested Items

Here is an example of creating a campaign with nested items:

```
POST: https://beta.aimatch.com/campaigns.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<campaign>
  <campaign-type>Guaranteed</campaign-type>
  <description>TestCampaign4 description</description>
  <end-date type="date" nil="true">2012-01-31T23:59:59-05:00</end-date>
  <external-id>TestCampaign4 ext id</external-id>
  <name>TestCampaign4</name>
  <notes>TestCampaign4 notes</notes>
  <start-date type="date">2011-06-16</start-date>
  <exposure-policy-attributes>
    <clicks-lifetime type="integer">10000</clicks-lifetime>
    <clicks-period type="integer">30000</clicks-period>
    <clicks-quantity type="integer" nil="true">50</clicks-quantity>
    <conversions-lifetime type="integer" nil="true">50</conversions-lifetime>
    <conversions-period type="integer">18000</conversions-period>
    <conversions-quantity type="integer" nil="true">4</conversions-quantity>
    <impressions-lifetime type="integer" nil="true">10000</impressions-lifetime>
    <impressions-per-session type="integer" nil="true">15</impressions-per-session>
    <impressions-period type="integer">21600</impressions-period>
    <impressions-quantity type="integer" nil="true">10</impressions-quantity>
  </exposure-policy-attributes>
  <flights-attributes type="array">
    <flight>
      <actions type="integer" nil="true">1</actions>
      <bonus-actions type="integer" nil="true">2</bonus-actions>
```

```
      <bonus-clicks type="integer" nil="true">3</bonus-clicks>
      <bonus-impressions type="integer" nil="true">4</bonus-impressions>
      <clicks type="integer" nil="true">5</clicks>
      <companion-type>Strict</companion-type>
      <cost-actions-local type="decimal" nil="true">10.5</cost-actions-local>
      <cost-clicks-local type="decimal" nil="true">5.5</cost-clicks-local>
      <cost-impressions-local type="decimal" nil="true">1.5</cost-impressions-local>
      <creative-selection-method>Weighted</creative-selection-method>
      <currency-id type="integer">840</currency-id>
      <daily-end-time type="datetime" nil="true">2000-01-01T19:00:00Z</daily-end-time>
      <daily-rev-cap-local type="decimal">50.99</daily-rev-cap-local>
      <daily-start-time type="datetime" nil="true">
2000-01-01T04:00:00Z</daily-start-time>
      <description>TestCampaign4TestFlight1 description</description>
      <end-date type="datetime">2014-03-20T15:00:00-05:00</end-date>
      <flat-rate-local type="integer" nil="true"></flat-rate-local>
      <flat-rate-flag type="boolean">false</flat-rate-flag>
      <flight-type>House</flight-type>
      <impressions type="integer" nil="true">20</impressions>
      <name>TestCampaign4TestFlight1</name>
      <notes>TestCampaign4TestFligh1 notes</notes>
      <reference-rate type="integer" nil="true"></reference-rate>
      <size-id type="integer" nil="true"></size-id>
      <start-date type="datetime">2011-02-17T13:00:00+05:00</start-date>
      <status>pending</status>
      <targets-group-logic>OR</targets-group-logic>
      <tier-id type="integer">1</tier-id>
      <user-info>TestCampaign4TestFlight1 user info</user-info>
      <weight type="integer">1</weight>
      <flight-third-party-attributes>
        <external-id>TestCampaign4Flight1 3rd party ext id 1</external-id>
        <third-party-id type="integer">3</third-party-id>
      </flight-third-party-attributes>
      <flight-action-policies-attributes type="array">
        <flight-action-policy>
          <clicks-period type="integer">72000</clicks-period>
          <clicks-trump-imps type="boolean">false</clicks-trump-imps>
          <imps-period type="integer">21600</imps-period>
          <revenue-local type="decimal">50.0</revenue-local>
          <action>
            <for-all-advertisers type="boolean">false</for-all-advertisers>
            <name>camp_4_flight_1_act_policy_1</name>
          </action>
        </flight-action-policy>
      </flight-action-policies-attributes>
      <flight-creatives-attributes type="array">
        <flight-creative>
          <active type="boolean">true</active>
          <creative-id type="integer">3</creative-id>
          <from-date type="date" nil="true">2011-02-18</from-date>
          <position type="integer">1</position>
          <to-date type="date" nil="true">2013-03-19</to-date>
          <weight type="integer">2</weight>
          <exposure-policy-attributes>
            <clicks-lifetime type="integer" nil="true">30</clicks-lifetime>
            <clicks-period type="integer">360000</clicks-period>
```

```
                <clicks-quantity type="integer" nil="true">5</clicks-quantity>
                <conversions-lifetime type="integer" nil="true">5</conversions-lifetime>
                <conversions-period type="integer">36000</conversions-period>
                <conversions-quantity type="integer" nil="true">2</conversions-quantity>
                <impressions-lifetime type="integer" nil="true">1000</impressions-lifetime>
                <impressions-per-session type="integer" nil="true">100<
    /impressions-per-session>
                <impressions-period type="integer">360000</impressions-period>
                <impressions-quantity type="integer" nil="true">200</impressions-quantity>
              </exposure-policy-attributes>
            </flight-creative>
          </flight-creatives-attributes>
          <areas-attributes type="array">
            <area>
              <name>camp_4_flight_1_area_1</name>
              <sites-attributes type="array">
                <site>
                  <description nil="true">Test Campaign 4 Flight1 Area1 Site1</description>
                  <name>camp_4_flight_1_area_1_site_1.com</name>
                  <revenue-cut type="decimal" nil="true">3.68</revenue-cut>
                </site>
              </sites-attributes>
            </area>
          </areas-attributes>
          <category-list type="array">
            <category>camp4_flight_1_cat_1</category>
            <category>camp4_flight_1_cat_2</category>
          </category-list>
          <top_priority type="integer">1</top_priority>
          <beacon_count type="integer">1</beacon_count>
          <sites-attributes type="array">
            <site>
              <description nil="true">camp4 flight 1 site 1</description>
              <name>camp_4_flight_1_site_1.com</name>
              <revenue-cut type="decimal" nil="true">3.5</revenue-cut>
            </site>
          </sites-attributes>
          <targets-attributes type="array">
            <target>
              <description nil="true">Test Campaign 4 Flight1 Target 1</description>
              <name>camp_4_flight_1_target_1 </name>
              <rules_attributes type="array">
                <rule>
                  <connector nil="true"></connector>
                  <left-parentheses-count type="integer">0</left-parentheses-count>
                  <left-value>DAYOFWEEK</left-value>
                  <operator>In</operator>
                  <position type="integer">1</position>
                  <right-parentheses-count type="integer">0</right-parentheses-count>
                  <right-value>TUESDAY,FRIDAY</right-value>
                </rule>
                <rule>
                  <connector>AND</connector>
                  <left-parentheses-count type="integer">0</left-parentheses-count>
                  <left-value>SITE</left-value>
                  <operator>Equals</operator>
```

```
            <position type="integer">2</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>camp_4_flight_1_area_1_site_1.com</right-value>
          </rule>
        </rules_attributes>
      </target>
    </targets-attributes>
    <tag-values-attributes type="array">
      <tag-value>
        <tag-id type="integer">1</tag-id>
        <value>Campaign4FlightTagvalue1</value>
      </tag-value>
    </tag-values-attributes>
    <ad-hoc-target-attributes>
      <rules-attributes type="array">
        <rule>
          <connector nil="true"></connector>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>GEO_DISTANCE_MILES</left-value>
          <operator>Greater or Equal To</operator>
          <position type="integer">1</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>2.56,51.521562,-0.261612</right-value>
        </rule>
        <rule>
          <connector nil="true">AND</connector>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>GEO_DISTANCE_MILES</left-value>
          <operator>Greater or Equal To</operator>
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>2.56,51.521562,-0.261612</right-value>
        </rule>
        <rule>
          <connector nil="true">OR</connector>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>GEO_DISTANCE_MILES</left-value>
          <operator>Greater or Equal To</operator>
          <position type="integer">3</position>
          <right-parentheses-count type="integer">1</right-parentheses-count>
          <right-value>2.56,51.521562,-0.261612</right-value>
        </rule>
      </rules-attributes>
    </ad-hoc-target-attributes>
  </flight>
</flights-attributes>
<advertiser-id>1</advertiser-id>
</campaign>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<campaign>
  <campaign-type>Guaranteed</campaign-type>
  <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
  <description>TestCampaign4 description</description>
  <end-date type="date">2012-01-31</end-date>
```

```
<external-id>TestCampaign4 ext id</external-id>
<flags type="integer">0</flags>
<id type="integer">4</id>
<lock-version type="integer">0</lock-version>
<name>TestCampaign4</name>
<notes>TestCampaign4 notes</notes>
<proposal-id type="integer" nil="true"></proposal-id>
<start-date type="date">2011-06-16</start-date>
<status nil="true"></status>
<updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
<flights type="array">
  <flight>
    <actions type="integer">1</actions>
    <bonus-actions type="integer">2</bonus-actions>
    <bonus-clicks type="integer">3</bonus-clicks>
    <bonus-impressions type="integer">4</bonus-impressions>
    <campaign-id type="integer">4</campaign-id>
    <clicks type="integer">5</clicks>
    <companion-type>Strict</companion-type>
    <cost-actions type="decimal">10.5</cost-actions>
    <cost-actions-local type="decimal">10.5</cost-actions-local>
    <cost-clicks type="decimal">5.5</cost-clicks>
    <cost-clicks-local type="decimal">5.5</cost-clicks-local>
    <cost-impressions type="decimal">1.5</cost-impressions>
    <cost-impressions-local type="decimal">1.5</cost-impressions-local>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <creative-selection-method>Weighted</creative-selection-method>
    <currency-id type="integer">840</currency-id>
    <conversion-rate type="float">1.0</conversion-rate>
    <daily-end-time type="datetime">2000-01-01T19:00:00Z</daily-end-time>
    <daily-rev-cap type="decimal">50.99</daily-rev-cap>
    <daily-rev-cap-local type="decimal">50.99</daily-rev-cap-local>
    <daily-start-time type="datetime">2000-01-01T04:00:00Z</daily-start-time>
    <description>TestCampaign4TestFlight1 description</description>
    <end-date type="datetime">2014-03-20T16:00:00-04:00</end-date>
    <flat-rate type="decimal" nil="true"></flat-rate>
    <flat-rate-local type="decimal" nil="true"></flat-rate-local>
    <flat-rate-flag type="boolean">false</flat-rate-flag>
    <flight-type>House</flight-type>
    <id type="integer">5</id>
    <impressions type="integer">20</impressions>
    <lock-version type="integer">0</lock-version>
    <name>TestCampaign4TestFlight1</name>
    <notes>TestCampaign4TestFligh1 notes</notes>
    <reference-rate type="integer" nil="true"></reference-rate>
    <size-id type="integer" nil="true"></size-id>
    <start-date type="datetime">2011-02-17T03:00:00-05:00</start-date>
    <status>missing creative</status>
    <targets-group-logic>OR</targets-group-logic>
    <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
    <user-info>TestCampaign4TestFlight1 user info</user-info>
    <weight type="integer">1</weight>
    <flight-creatives type="array">
      <flight-creative>
        <active type="boolean">true</active>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
```

```
    <creative-id type="integer">3</creative-id>
    <flight-id type="integer">5</flight-id>
    <from-date type="date">2011-02-18</from-date>
    <id type="integer">5</id>
    <position type="integer">1</position>
    <to-date type="date">2013-03-19</to-date>
    <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
    <weight type="integer">2</weight>
    <exposure-policy>
      <clicks-lifetime type="integer">30</clicks-lifetime>
      <clicks-period type="integer">360000</clicks-period>
      <clicks-quantity type="integer">5</clicks-quantity>
      <conversions-lifetime type="integer">5</conversions-lifetime>
      <conversions-period type="integer">36000</conversions-period>
      <conversions-quantity type="integer">2</conversions-quantity>
      <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
      <id type="integer">2</id>
      <impressions-lifetime type="integer">1000</impressions-lifetime>
      <impressions-per-session type="integer">100</impressions-per-session>
      <impressions-period type="integer">360000</impressions-period>
      <impressions-quantity type="integer">200</impressions-quantity>
      <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
    </exposure-policy>
  </flight-creative>
</flight-creatives>
<targets type="array">
  <target>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <creator-id type="integer" nil="true"></creator-id>
    <description>Test Campaign 4 Flight1 Target 1</description>
    <id type="integer">9</id>
    <lock-version type="integer">0</lock-version>
    <name>camp_4_flight_1_target_1 </name>
    <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
    <rules type="array">
      <rule type="TargetRule">
        <connector nil="true"></connector>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
        <id type="integer">47</id>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <left-value>DAYOFWEEK</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>In</operator>
        <position type="integer">1</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>TUESDAY,FRIDAY</right-value>
        <target-id type="integer">9</target-id>
        <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
      </rule>
      <rule type="TargetRule">
        <connector>AND</connector>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
        <id type="integer">48</id>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <left-value>SITE</left-value>
        <lock-version type="integer">0</lock-version>
```

```
        <operator>Equals</operator>
        <position type="integer">2</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>camp_4_flight_1_area_1_site_1.com</right-value>
        <target-id type="integer">9</target-id>
        <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
      </rule>
    </rules>
  </target>
</targets>
<ad-hoc-target>
  <description nil="true"/>
  <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
  <id type="integer">366</id>
  <creator-id type="integer" nil="true"/>
  <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
  <name>aiMatch_ad_hoc_target_1355249089_822</name>
  <lock-version type="integer">0</lock-version>
  <rules type="array">
    <rule type="TargetRule">
      <left-value>GEO_DISTANCE_MILES</left-value>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
      <id type="integer">2883</id>
      <right-value>2.56,51.521562,-0.261612</right-value>
      <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
      <position type="integer">1</position>
      <connector nil="true"/>
      <operator>Greater or Equal To</operator>
      <target-id type="integer">366</target-id>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <lock-version type="integer">0</lock-version>
    </rule>
    <rule type="TargetRule">
      <left-value>GEO_DISTANCE_MILES</left-value>
      <left-parentheses-count type="integer">1</left-parentheses-count>
      <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
      <id type="integer">2884</id>
      <right-value>2.56,51.521562,-0.261612</right-value>
      <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
      <position type="integer">2</position>
      <connector>AND</connector>
      <operator>Greater or Equal To</operator>
      <target-id type="integer">366</target-id>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <lock-version type="integer">0</lock-version>
    </rule>
    <rule type="TargetRule">
      <left-value>GEO_DISTANCE_MILES</left-value>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
      <id type="integer">2885</id>
      <right-value>2.56,51.521562,-0.261612</right-value>
      <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
      <position type="integer">3</position>
      <connector>OR</connector>
```

```
      <operator>Greater or Equal To</operator>
      <target-id type="integer">366</target-id>
      <right-parentheses-count type="integer">1</right-parentheses-count>
      <lock-version type="integer">0</lock-version>
    </rule>
 </rules>
</ad-hoc-target>
<flight-third-party>
  <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
  <external-id>TestCampaign4Flight1 3rd party ext id 1</external-id>
  <flight-id type="integer">5</flight-id>
  <id type="integer">3</id>
  <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
  <third-party>
    <created-at type="datetime">2011-04-25T09:55:51-04:00</created-at>
    <display-name>Admeld</display-name>
    <id type="integer">3</id>
    <lock-version type="integer">0</lock-version>
    <name>admeld</name>
    <updated-at type="datetime">2011-04-25T09:55:51-04:00</updated-at>
  </third-party>
</flight-third-party>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">1</id>
  <lock-version type="integer">0</lock-version>
  <name>Direct Sold</name>
  <preemptible type="boolean">false</preemptible>
  <prioritization-method>Schedule</prioritization-method>
  <rank type="integer">1</rank>
  <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
</tier>
<areas type="array">
  <area>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <id type="integer">11</id>
    <lock-version type="integer">0</lock-version>
    <name>camp_4_flight_1_area_1</name>
    <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
    <sites type="array">
      <site>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
        <description>Test Campaign 4 Flight1 Area1 Site1</description>
        <id type="integer">113</id>
        <lock-version type="integer">0</lock-version>
        <name>camp_4_flight_1_area_1_site_1.com</name>
        <revenue-cut type="decimal">3.68</revenue-cut>
        <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
      </site>
    </sites>
  </area>
</areas>
```

```
          <tag-values type="array">
            <tag-value>
              <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
              <id type="integer">8858</id>
              <tag-id type="integer">1</tag-id>
              <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
              <value>Campaign4FlightTagvalue1</value>
            </tag-value>
          </tag-values>
          <sites type="array">
            <site>
              <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
              <description>camp4 flight 1 site 1</description>
              <id type="integer">112</id>
              <lock-version type="integer">0</lock-version>
              <name>camp_4_flight_1_site_1.com</name>
              <revenue-cut type="decimal">3.5</revenue-cut>
              <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
            </site>
          </sites>
          <flight-action-policies type="array"/>
          <category-list type="array">
            <category>camp4_flight_1_cat_1</category>
            <category>camp4_flight_1_cat_2</category>
          </category-list>
          <beacon_count type="integer">1</beacon_count>
          <top_priority type="integer">1</top_priority>
        </flight>
      </flights>
      <advertiser>
        <legal-entity-name></legal-entity-name>
        <status nil="true"/>
        <id type="integer">1</id>
        <name>TestAdvertiser1</name>
        <industry-id type="integer" nil="true"/>
        <credit-limit type="decimal" nil="true"></credit-limit>
        <account-number></account-number>
        <business-id1 nil="true"/>
        <business-id2 nil="true"/>
        <customer-status>Active</customer-status>
        <bill-direct type="boolean">true</bill-direct>
        <credit-approved type="boolean">false</credit-approved>
        <updated-at type="datetime">2011-05-27T19:30:41-04:00</updated-at>
        <lock-version type="integer">1</lock-version>
        <pre-pay type="boolean">false</pre-pay>
        <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
        <standard-discount type="decimal" nil="true"/>
        <region-id type="integer" nil="true"/>
        <contacts type="array">
          <contact>
            <address1 nil="true"></address1>
            <address2 nil="true"></address2>
            <city nil="true"></city>
            <contactable-id type="integer">1</contactable-id>
            <contactable-type>Advertiser</contactable-type>
            <country nil="true"></country>
```

```
        <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
        <email nil="true"></email>
        <full-name>Test Advertiser User</full-name>
        <id type="integer">1</id>
        <lock-version type="integer">0</lock-version>
        <phone nil="true"></phone>
        <state nil="true"></state>
        <title nil="true"></title>
        <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
        <website nil="true"></website>
        <zip nil="true"></zip>
      </contact>
    </contacts>
  </advertiser>
  <exposure-policy>
    <clicks-lifetime type="integer">10000</clicks-lifetime>
    <clicks-period type="integer">30000</clicks-period>
    <clicks-quantity type="integer">50</clicks-quantity>
    <conversions-lifetime type="integer">50</conversions-lifetime>
    <conversions-period type="integer">18000</conversions-period>
    <conversions-quantity type="integer">4</conversions-quantity>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <id type="integer">1</id>
    <impressions-lifetime type="integer">10000</impressions-lifetime>
    <impressions-per-session type="integer">15</impressions-per-session>
    <impressions-period type="integer">21600</impressions-period>
    <impressions-quantity type="integer">10</impressions-quantity>
    <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
  </exposure-policy>
</campaign>
```

Update a Campaign

Here is an example of updating a campaign:

```
PUT: https://beta.aimatch.com/areas/119.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<campaign>
  <description>TestCampaign4 description updated</description>
  <name>TestCampaign4Updated</name>
  <lock-version type="integer">0</lock-version>
  <notes>TestCampaign4 notes updated</notes>
  <start-date type="date">2011-06-17</start-date>
  <exposure-policy-attributes>
    <id type="integer">1</id>
    <clicks-lifetime type="integer">10001</clicks-lifetime>
    <clicks-period type="integer">30001</clicks-period>
    <clicks-quantity type="integer" nil="true">51</clicks-quantity>
    <conversions-lifetime type="integer" nil="true">51</conversions-lifetime>
    <conversions-period type="integer">18001</conversions-period>
    <conversions-quantity type="integer" nil="true">5</conversions-quantity>
    <impressions-lifetime type="integer" nil="true">10001</impressions-lifetime>
    <impressions-per-session type="integer" nil="true">16</impressions-per-session>
    <impressions-period type="integer">21601</impressions-period>
```

```
      <impressions-quantity type="integer" nil="true">11</impressions-quantity>
    </exposure-policy-attributes>
    <flights-attributes type="array">
      <flight>
        <id type="integer">5</id>
        <lock-version type="integer">0</lock-version>
        <actions type="integer" nil="true">11</actions>
        <bonus-actions type="integer" nil="true">12</bonus-actions>
        <bonus-clicks type="integer" nil="true">13</bonus-clicks>
        <bonus-impressions type="integer" nil="true">14</bonus-impressions>
        <clicks type="integer" nil="true">15</clicks>
        <cost-actions-local type="decimal" nil="true">11.5</cost-actions-local>
        <cost-clicks-local type="decimal" nil="true">6.5</cost-clicks-local>
        <cost-impressions-local type="decimal" nil="true">1.6</cost-impressions-local>
        <currency-id type="integer">840</currency-id>
        <description>TestCampaign4TestFlight1 description updated</description>
        <end-date type="datetime">2014-03-21T15:00:00-05:00</end-date>
        <impressions type="integer" nil="true">21</impressions>
        <name>TestCampaign4TestFlight1Updated</name>
        <notes>TestCampaign4TestFligh1 notes updated</notes>
        <start-date type="datetime">2011-02-18T13:00:00+05:00</start-date>
        <user-info>TestCampaign4TestFlight1 user info updated</user-info>
        <areas-attributes type="array">
          <area>
            <id type="integer">11</id>
            <lock-version type="integer">0</lock-version>
            <name>camp_4_flight_1_area_1_updated</name>
            <sites-attributes type="array">
              <site>
                <id type="integer">113</id>
                <lock-version type="integer">0</lock-version>
                <description nil="true">
  Test Campaign 4 Flight1 Area1 Site1 updated</description>
                <name>camp_4_flight_1_area_1_site_1_updated.com</name>
                <revenue-cut type="decimal" nil="true">4.68</revenue-cut>
              </site>
            </sites-attributes>
          </area>
        </areas-attributes>
        <ad-hoc-target-attributes>
          <description nil="true"/>
          <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
          <id type="integer">366</id>
          <creator-id type="integer" nil="true"/>
          <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
          <name>aiMatch_ad_hoc_target_1355249089_822</name>
          <lock-version type="integer">0</lock-version>
          <rules-attributes type="array">
            <rule>
              <left-value>GEO_DISTANCE_MILES</left-value>
              <left-parentheses-count type="integer">0</left-parentheses-count>
              <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
              <id type="integer">2883</id>
              <right-value>3.56,51.521562,-0.261612</right-value>
              <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
              <position type="integer">1</position>
```

```
            <connector nil="true"/>
            <operator>Greater or Equal To</operator>
            <target-id type="integer">366</target-id>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <lock-version type="integer">0</lock-version>
          </rule>
          <rule>
            <left-value>GEO_DISTANCE_MILES</left-value>
            <left-parentheses-count type="integer">1</left-parentheses-count>
            <id type="integer">2884</id>
            <right-value>3.56,51.521562,-0.261612</right-value>
            <position type="integer">2</position>
            <connector>AND</connector>
            <operator>Greater or Equal To</operator>
            <target-id type="integer">366</target-id>
            <right-parentheses-count type="integer">0</right-parentheses-count>
          </rule>
          <rule>
            <left-value>GEO_DISTANCE_MILES</left-value>
            <left-parentheses-count type="integer">0</left-parentheses-count>
            <id type="integer">2885</id>
            <right-value>3.56,51.521562,-0.261612</right-value>
            <position type="integer">3</position>
            <connector>OR</connector>
            <operator>Greater or Equal To</operator>
            <target-id type="integer">366</target-id>
            <right-parentheses-count type="integer">1</right-parentheses-count>
          </rule>
        </rules-attributes>
      </ad-hoc-target-attributes>
    </flight>
  </flights-attributes>
</campaign>
```

The response will not contain anything, but the status code will be 200.

When you specify the updated campaign, you receive the following::

```
GET: https://beta.aimatch.com/campaigns/4.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<campaign>
  <campaign-type>Guaranteed</campaign-type>
  <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
  <description>TestCampaign4 description updated</description>
  <end-date type="date">2012-01-31</end-date>
  <external-id>TestCampaign4 ext id</external-id>
  <flags type="integer">0</flags>
  <id type="integer">4</id>
  <lock-version type="integer">1</lock-version>
  <name>TestCampaign4Updated</name>
  <notes>TestCampaign4 notes updated</notes>
  <proposal-id type="integer" nil="true"></proposal-id>
  <start-date type="date">2011-06-17</start-date>
  <status nil="true"></status>
  <updated-at type="datetime">2011-05-27T19:47:12-04:00</updated-at>
  <flights type="array">
    <flight>
```

```
<actions type="integer">11</actions>
<bonus-actions type="integer">12</bonus-actions>
<bonus-clicks type="integer">13</bonus-clicks>
<bonus-impressions type="integer">14</bonus-impressions>
<campaign-id type="integer">4</campaign-id>
<clicks type="integer">15</clicks>
<companion-type>Strict</companion-type>
<cost-actions type="decimal">11.5</cost-actions>
<cost-actions-local type="decimal">11.5</cost-actions-local>
<cost-clicks type="decimal">6.5</cost-clicks>
<cost-clicks-local type="decimal">6.5</cost-clicks-local>
<cost-impressions type="decimal">1.6</cost-impressions>
<cost-impressions-local type="decimal">1.6</cost-impressions-local>
<created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
<creative-selection-method>Weighted</creative-selection-method>
<currency-id type="integer">840</currency-id>
<conversion-rate type="float">1.0</conversion-rate>
<daily-end-time type="datetime">2000-01-01T19:00:00Z</daily-end-time>
<daily-rev-cap type="decimal">50.99</daily-rev-cap>
<daily-rev-cap-local type="decimal">50.99</daily-rev-cap-local>
<daily-start-time type="datetime">2000-01-01T04:00:00Z</daily-start-time>
<description>TestCampaign4TestFlight1 description updated</description>
<end-date type="datetime">2014-03-21T16:00:00-04:00</end-date>
<flat-rate type="decimal" nil="true"></flat-rate>
<flat-rate-local type="decimal" nil="true"></flat-rate-local>
<flat-rate-flag type="boolean">false</flat-rate-flag>
<flight-type>House</flight-type>
<id type="integer">5</id>
<impressions type="integer">21</impressions>
<lock-version type="integer">1</lock-version>
<name>TestCampaign4TestFlight1Updated</name>
<notes>TestCampaign4TestFligh1 notes updated</notes>
<reference-rate type="integer" nil="true"></reference-rate>
<size-id type="integer" nil="true"></size-id>
<start-date type="datetime">2011-02-18T03:00:00-05:00</start-date>
<status>pending</status>
<targets-group-logic>OR</targets-group-logic>
<updated-at type="datetime">2011-05-27T19:47:12-04:00</updated-at>
<user-info>TestCampaign4TestFlight1 user info updated</user-info>
<weight type="integer">1</weight>
<flight-creatives type="array">
  <flight-creative>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <creative-id type="integer">3</creative-id>
    <flight-id type="integer">5</flight-id>
    <from-date type="date">2011-02-18</from-date>
    <id type="integer">5</id>
    <position type="integer">1</position>
    <to-date type="date">2013-03-19</to-date>
    <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
    <weight type="integer">2</weight>
    <exposure-policy>
      <clicks-lifetime type="integer">30</clicks-lifetime>
      <clicks-period type="integer">360000</clicks-period>
      <clicks-quantity type="integer">5</clicks-quantity>
```

```
      <conversions-lifetime type="integer">5</conversions-lifetime>
      <conversions-period type="integer">36000</conversions-period>
      <conversions-quantity type="integer">2</conversions-quantity>
      <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
      <id type="integer">2</id>
      <impressions-lifetime type="integer">1000</impressions-lifetime>
      <impressions-per-session type="integer">100</impressions-per-session>
      <impressions-period type="integer">360000</impressions-period>
      <impressions-quantity type="integer">200</impressions-quantity>
      <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
    </exposure-policy>
  </flight-creative>
</flight-creatives>
<targets type="array">
  <target>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <creator-id type="integer" nil="true"></creator-id>
    <description>Test Campaign 4 Flight1 Target 1</description>
    <flight-id type="">5</flight-id>
    <id type="integer">9</id>
    <lock-version type="integer">1</lock-version>
    <name>camp_4_flight_1_target_1 </name>
    <target-id type="">9</target-id>
    <updated-at type="datetime">2011-05-27T19:47:12-04:00</updated-at>
    <rules type="array">
      <rule type="TargetRule">
        <connector nil="true"></connector>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
        <id type="integer">47</id>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <left-value>DAYOFWEEK</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>In</operator>
        <position type="integer">1</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>TUESDAY,FRIDAY</right-value>
        <target-id type="integer">9</target-id>
        <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
      </rule>
      <rule type="TargetRule">
        <connector>AND</connector>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
        <id type="integer">48</id>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <left-value>SITE</left-value>
        <lock-version type="integer">1</lock-version>
        <operator>Equals</operator>
        <position type="integer">2</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>camp_4_flight_1_area_1_site_1_updated.com</right-value>
        <target-id type="integer">9</target-id>
        <updated-at type="datetime">2011-05-27T19:47:12-04:00</updated-at>
      </rule>
    </rules>
  </target>
</targets>
```

```
<flight-third-party>
  <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
  <external-id>TestCampaign4Flight1 3rd party ext id 1</external-id>
  <flight-id type="integer">5</flight-id>
  <id type="integer">3</id>
  <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
  <third-party>
    <created-at type="datetime">2011-04-25T09:55:51-04:00</created-at>
    <display-name>Admeld</display-name>
    <id type="integer">3</id>
    <lock-version type="integer">0</lock-version>
    <name>admeld</name>
    <updated-at type="datetime">2011-04-25T09:55:51-04:00</updated-at>
  </third-party>
</flight-third-party>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">1</id>
  <lock-version type="integer">0</lock-version>
  <name>Direct Sold</name>
  <preemptible type="boolean">false</preemptible>
  <prioritization-method>Schedule</prioritization-method>
  <rank type="integer">1</rank>
  <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
</tier>
<areas type="array">
  <area>
    <area-id type="">11</area-id>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <flight-id type="">5</flight-id>
    <id type="integer">11</id>
    <lock-version type="integer">1</lock-version>
    <name>camp_4_flight_1_area_1_updated</name>
    <updated-at type="datetime">2011-05-27T19:47:12-04:00</updated-at>
    <sites type="array">
      <site>
        <area-id type="">11</area-id>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
        <description>Test Campaign 4 Flight1 Area1 Site1 updated</description>
        <id type="integer">113</id>
        <lock-version type="integer">1</lock-version>
        <name>camp_4_flight_1_area_1_site_1_updated.com</name>
        <revenue-cut type="decimal">4.68</revenue-cut>
        <site-id type="">113</site-id>
        <updated-at type="datetime">2011-05-27T19:47:12-04:00</updated-at>
      </site>
    </sites>
  </area>
</areas>
<tag-values type="array">
  <tag-value>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
```

```
        <id type="integer">8858</id>
        <tag-id type="integer">1</tag-id>
        <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
        <value>Campaign4FlightTagvalue1</value>
      </tag-value>
    </tag-values>
    <sites type="array">
      <site>
        <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
        <description>camp4 flight 1 site 1</description>
        <flight-id type="">5</flight-id>
        <id type="integer">112</id>
        <lock-version type="integer">0</lock-version>
        <name>camp_4_flight_1_site_1.com</name>
        <revenue-cut type="decimal">3.5</revenue-cut>
        <site-id type="">112</site-id>
        <updated-at type="datetime">2011-05-27T19:30:53-04:00</updated-at>
      </site>
    </sites>
    <flight-action-policies type="array"/>
    <category-list type="array">
      <category>camp4_flight_1_cat_1</category>
      <category>camp4_flight_1_cat_2</category>
    </category-list>
    <beacon_count type="integer">1</beacon_count>
    <top_priority type="integer">1</top_priority>
  </flight>
</flights>
<advertiser>
  <legal-entity-name></legal-entity-name>
  <status nil="true"/>
  <id type="integer">1</id>
  <name>TestAdvertiser</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal" nil="true"></credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status>Active</customer-status>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-05-27T19:30:41-04:00</updated-at>
  <lock-version type="integer">1</lock-version>
  <pre-pay type="boolean">false</pre-pay>
  <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
  <standard-discount type="decimal" nil="true"/>
  <region-id type="integer" nil="true"/>
  <contacts type="array">
    <contact>
      <address1 nil="true"></address1>
      <address2 nil="true"></address2>
      <city nil="true"></city>
      <contactable-id type="integer">1</contactable-id>
      <contactable-type>Advertiser</contactable-type>
      <country nil="true"></country>
      <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
```

```
        <email nil="true"></email>
        <full-name>Test Advertiser User</full-name>
        <id type="integer">1</id>
        <lock-version type="integer">0</lock-version>
        <phone nil="true"></phone>
        <state nil="true"></state>
        <title nil="true"></title>
        <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
        <website nil="true"></website>
        <zip nil="true"></zip>
      </contact>
    </contacts>
  </advertiser>
  <exposure-policy>
    <clicks-lifetime type="integer">10001</clicks-lifetime>
    <clicks-period type="integer">30001</clicks-period>
    <clicks-quantity type="integer">51</clicks-quantity>
    <conversions-lifetime type="integer">51</conversions-lifetime>
    <conversions-period type="integer">18001</conversions-period>
    <conversions-quantity type="integer">5</conversions-quantity>
    <created-at type="datetime">2011-05-27T19:30:53-04:00</created-at>
    <id type="integer">1</id>
    <impressions-lifetime type="integer">10001</impressions-lifetime>
    <impressions-per-session type="integer">16</impressions-per-session>
    <impressions-period type="integer">21601</impressions-period>
    <impressions-quantity type="integer">11</impressions-quantity>
    <updated-at type="datetime">2011-05-27T19:47:12-04:00</updated-at>
  </exposure-policy>
</campaign>
```


Managing Flights

Get a List of Flights

Here is an example of getting a list of flights:

```
GET: https://beta.aimatch.com/flights.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<flights type="array">
  <flight>
    <id type="integer">1</id>
    <name>Flight1</name>
  </flight>
  <flight>
    <id type="integer">2</id>
    <name>Flight2</name>
  </flight>
  <flight>
    <id type="integer">3</id>
    <name>Flight3</name>
  </flight>
```

```
</flights>
```

This XML response provides 30 flights at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/flights.xml?page=3
```

Get a Specific Flight

Here is an example of getting a specific flight:

```
GET: https://beta.aimatch.com/flights/1.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<flight>
  <actions type="integer" nil="true"></actions>
  <bonus-actions type="integer" nil="true"></bonus-actions>
  <bonus-clicks type="integer" nil="true"></bonus-clicks>
  <bonus-impressions type="integer" nil="true"></bonus-impressions>
  <campaign-id type="integer">1</campaign-id>
  <clicks type="integer" nil="true"></clicks>
  <companion-type></companion-type>
  <cost-actions type="decimal" nil="true"></cost-actions>
  <cost-actions-local type="decimal" nil="true"></cost-actions-local>
  <cost-clicks type="decimal" nil="true"></cost-clicks>
  <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
  <cost-impressions type="decimal" nil="true"></cost-impressions>
  <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <creative-selection-method>Weighted</creative-selection-method>
  <currency-id type="integer">840</currency-id>
  <conversion-rate type="float">1.0</conversion-rate>
  <daily-end-time type="datetime" nil="true"></daily-end-time>
  <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
  <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
  <daily-start-time type="datetime" nil="true"></daily-start-time>
  <description></description>
  <end-date type="datetime">2011-06-25T09:00:00-04:00</end-date>
  <flat-rate type="decimal" nil="true"></flat-rate>
  <flat-rate-local type="decimal" nil="true"></flat-rate-local>
  <flat-rate-flag type="boolean">false</flat-rate-flag>
  <flight-type>House</flight-type>
  <id type="integer">1</id>
  <impressions type="integer" nil="true"></impressions>
  <lock-version type="integer">1</lock-version>
  <name>campaign_3_flight_1</name>
  <notes></notes>
  <reference-rate type="integer" nil="true"></reference-rate>
  <size-id type="integer">0</size-id>
  <start-date type="datetime">2011-05-09T09:00:00-04:00</start-date>
  <status>pending</status>
  <targets-group-logic></targets-group-logic>
  <updated-at type="datetime">2011-05-11T15:46:54-04:00</updated-at>
  <user-info></user-info>
```

```
<weight type="integer">1</weight>
<actual-impressions type="integer">22079581</actual-impressions>
<actual-clicks type="integer">0</actual-clicks>
<actual-actions type="integer">0</actual-actions>
<actual-custom-actions type="integer">0</actual-custom-actions>
<flight-creatives type="array">
  <flight-creative>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
    <creative-id type="integer">1</creative-id>
    <flight-id type="integer">1</flight-id>
    <from-date type="date" nil="true"></from-date>
    <id type="integer">1</id>
    <position type="integer">1</position>
    <to-date type="date" nil="true"></to-date>
    <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
    <weight type="integer">1</weight>
  </flight-creative>
  <flight-creative>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-05-11T15:31:59-04:00</created-at>
    <creative-id type="integer">2</creative-id>
    <flight-id type="integer">1</flight-id>
    <from-date type="date" nil="true"></from-date>
    <id type="integer">3</id>
    <position type="integer" nil="true"></position>
    <to-date type="date" nil="true"></to-date>
    <updated-at type="datetime">2011-05-11T15:34:51-04:00</updated-at>
    <weight type="integer">1</weight>
  </flight-creative>
</flight-creatives>
<targets type="array"/>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">1</id>
  <lock-version type="integer">0</lock-version>
  <name>Direct Sold</name>
  <preemptible type="boolean">false</preemptible>
  <prioritization-method>Schedule</prioritization-method>
  <rank type="integer">1</rank>
  <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
</tier>
<areas type="array">
  <area>
    <area-id type="">4</area-id>
    <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
    <flight-id type="">1</flight-id>
    <id type="integer">4</id>
    <lock-version type="integer">0</lock-version>
    <name>Article</name>
    <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
    <sites type="array"/>
```

```
      </area>
      <area>
        <area-id type="">8</area-id>
        <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
        <flight-id type="">1</flight-id>
        <id type="integer">8</id>
        <lock-version type="integer">0</lock-version>
        <name>Downloads</name>
        <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
        <sites type="array"/>
      </area>
    </areas>
    <tag-values type="array"/>
    <sites type="array">
      <site>
        <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
        <description nil="true"></description>
        <flight-id type="">1</flight-id>
        <id type="integer">1</id>
        <lock-version type="integer">0</lock-version>
        <name>reviews.cnet.com</name>
        <revenue-cut type="decimal" nil="true"></revenue-cut>
        <site-id type="">1</site-id>
        <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
      </site>
      <site>
        <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
        <description nil="true"></description>
        <flight-id type="">1</flight-id>
        <id type="integer">5</id>
        <lock-version type="integer">0</lock-version>
        <name>alecopy.com</name>
        <revenue-cut type="decimal" nil="true"></revenue-cut>
        <site-id type="">5</site-id>
        <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
      </site>
    </sites>
    <flight-action-policies type="array"/>
    <category-list type="array">
      <category>Soda</category>
    </category-list>
    <beacon_count type="integer">0</beacon_count>
    <top_priority type="integer">0</top_priority>
  </flight>
```


Managing Creatives

Get a List of Creatives

Here is an example of getting a list of creatives:

```
GET: https://beta.aimatch.com/creative.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<creatives type="array">
  <creative>
    <id type="integer">1</id>
    <name>TestCreative1</name>
  </creative>
  <creative>
    <id type="integer">2</id>
    <name>TestCreative2</name>
  </creative>
  <creative>
    <id type="integer">3</id>
    <name>TestCreative3</name>
  </creative>
</creatives>
```

This XML response provides 30 creatives at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/creative.xml?page=3
```

Get a Specific Creative

Here is an example of getting a specific creative:

```
GET: https://beta.aimatch.com/creative/3.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <ad-text nil="true"></ad-text>
  <advertiser-id type="integer">2</advertiser-id>
  <alt></alt>
  <alt-image-id type="integer" nil="true"></alt-image-id>
  <alt-image-reference nil="true"></alt-image-reference>
  <alt-image-type nil="true"></alt-image-type>
  <alt-image-url nil="true"></alt-image-url>
  <click-url></click-url>
  <created-at type="datetime">2011-05-25T13:18:09-04:00</created-at>
  <description></description>
  <duration type="integer" nil="true"></duration>
  <filename nil="true"></filename>
  <id type="integer">3</id>
  <image-url></image-url>
  <location-type>url</location-type>
  <lock-version type="integer">0</lock-version>
  <map nil="true"></map>
  <media-id type="integer" nil="true"></media-id>
  <media-reference></media-reference>
  <name>TestCreative3</name>
  <notes></notes>
  <request-headers></request-headers>
  <request-url></request-url>
  <template-id type="integer" nil="true"></template-id>
  <updated-at type="datetime">2011-05-25T13:18:09-04:00</updated-at>
```

```
  <user-info></user-info>
  <format-field-values type="array">
    <format-field-value type="CreativeFormatFieldValue">
      <created-at type="datetime">2011-05-25T13:18:09-04:00</created-at>
      <creative-id type="integer">3</creative-id>
      <format-field-id type="integer">14</format-field-id>
      <id type="integer">1</id>
      <updated-at type="datetime">2011-05-25T13:18:09-04:00</updated-at>
      <value>123</value>
      <format-field type="CreativeFormatFieldValue">
        <created-at type="datetime">2011-04-25T09:55:48-04:00</created-at>
        <description>example: code before but not including "/jview/"
in ad tag insertion order</description>
        <id type="integer">14</id>
        <name>Atlas_unique_identifier</name>
        <required type="boolean">true</required>
        <updated-at type="datetime">2011-04-25T09:55:48-04:00</updated-at>
      </format-field>
    </format-field-value>
    <format-field-value type="CreativeFormatFieldValue">
      <created-at type="datetime">2011-05-25T13:18:09-04:00</created-at>
      <creative-id type="integer">3</creative-id>
      <format-field-id type="integer">15</format-field-id>
      <id type="integer">2</id>
      <updated-at type="datetime">2011-05-25T13:18:09-04:00</updated-at>
      <value>abc</value>
      <format-field type="CreativeFormatFieldValue">
        <created-at type="datetime">2011-04-25T09:55:48-04:00</created-at>
        <description>example: code after but not including "/jview/"</description>
        <id type="integer">15</id>
        <name>Atlas_unique_code</name>
        <required type="boolean">true</required>
        <updated-at type="datetime">2011-04-25T09:55:48-04:00</updated-at>
      </format-field>
    </format-field-value>
  </format-field-values>
  <format>
    <id type="integer">13</id>
    <name>Standard-Atlas_javascript</name>
  </format>
  <size>
    <id type="integer">7</id>
    <name>120X60</name>
  </size>
</creative>
```


Create a Creative with a format_field (a Nested Item)

Here is an example of creating a creative with a format_field (a nested item):

```
POST: https://beta.aimatch.com/creative.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <advertiser-id type="integer">2</advertiser-id>
  <format-id type="integer">8</format-id>
```

```
    <location-type>url</location-type>
    <name>TestCreative4</name>
    <size-id type="integer">10</size-id>
    <format-field-values-attributes type="array">
      <format-field-value>
        <format-field-id type="integer">7</format-field-id>
        <value>PID123</value>
      </format-field-value>
    </format-field-values-attributes>
  </creative>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <ad-text nil="true"></ad-text>
  <advertiser-id type="integer">2</advertiser-id>
  <alt nil="true"></alt>
  <alt-image-id type="integer" nil="true"></alt-image-id>
  <alt-image-reference nil="true"></alt-image-reference>
  <alt-image-type nil="true"></alt-image-type>
  <alt-image-url nil="true"></alt-image-url>
  <click-url nil="true"></click-url>
  <created-at type="datetime">2011-05-27T20:09:56-04:00</created-at>
  <description nil="true"></description>
  <duration type="integer" nil="true"></duration>
  <filename nil="true"></filename>
  <id type="integer">4</id>
  <image-url nil="true"></image-url>
  <location-type>url</location-type>
  <lock-version type="integer">0</lock-version>
  <map nil="true"></map>
  <media-id type="integer" nil="true"></media-id>
  <media-reference nil="true"></media-reference>
  <name>TestCreative4</name>
  <notes nil="true"></notes>
  <request-headers nil="true"></request-headers>
  <request-url nil="true"></request-url>
  <template-id type="integer" nil="true"></template-id>
  <updated-at type="datetime">2011-05-27T20:09:56-04:00</updated-at>
  <user-info nil="true"></user-info>
  <format-field-values type="array">
    <format-field-value type="CreativeFormatFieldValue">
      <created-at type="datetime">2011-05-27T20:09:56-04:00</created-at>
      <creative-id type="integer">4</creative-id>
      <format-field-id type="integer">7</format-field-id>
      <id type="integer">3</id>
      <updated-at type="datetime">2011-05-27T20:09:56-04:00</updated-at>
      <value>PID123</value>
      <format-field type="CreativeFormatFieldValue">
        <created-at type="datetime">2011-04-25T09:55:47-04:00</created-at>
        <description nil="true"></description>
        <id type="integer">7</id>
        <name>PID</name>
        <required type="boolean">true</required>
        <updated-at type="datetime">2011-04-25T09:55:47-04:00</updated-at>
      </format-field>
```

```
    </format-field-value>
  </format-field-values>
  <format>
    <id type="integer">8</id>
    <name>Standard-Pointroll_javascript</name>
  </format>
  <size>
    <id type="integer">10</id>
    <name>728x90</name>
  </size>
</creative>
```

Create a Creative Providing Advertiser Name and Format Name Instead of Advertiser ID or Format ID

Here is an example of creating a creative that provides advertiser name and format name instead of advertiser ID or format ID:

```
POST: https://beta.aimatch.com/creative.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <advertiser-name>TestAdvertiser1</advertiser-name>
  <format-name>Standard-Pointroll_javascript</format-name>
  <location-type>url</location-type>
  <name>TestCreativeWithAdvertiserName</name>
  <size-id type="integer">10</size-id>
</creative>
```

Response (HTTP status: 201 Created):

Update a Creative

Here is an example of updating a creative:

```
PUT: https://beta.aimatch.com/creative/10.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <lock-version type="integer">0</lock-version>
  <description>test creative 4 updated</description>
  <name>TestCreative4Updated</name>
  <size-id type="integer">9</size-id>
  <format-field-values-attributes type="array">
    <format-field-value>
      <id type="integer">3</id>
      <value>PID456</value>
    </format-field-value>
  </format-field-values-attributes>
</creative>
```

The response will not contain anything, but the status code will be 200.

The creative now looks like this:

```
GET: https://beta.aimatch.com/creative/10.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
```

```
  <ad-text nil="true"></ad-text>
  <advertiser-id type="integer">2</advertiser-id>
  <alt nil="true"></alt>
  <alt-image-id type="integer" nil="true"></alt-image-id>
  <alt-image-reference nil="true"></alt-image-reference>
  <alt-image-type nil="true"></alt-image-type>
  <alt-image-url nil="true"></alt-image-url>
  <click-url nil="true"></click-url>
  <created-at type="datetime">2011-05-27T20:09:56-04:00</created-at>
  <description>test creative 4 updated</description>
  <duration type="integer" nil="true"></duration>
  <filename nil="true"></filename>
  <id type="integer">4</id>
  <image-url nil="true"></image-url>
  <location-type>url</location-type>
  <lock-version type="integer">1</lock-version>
  <map nil="true"></map>
  <media-id type="integer" nil="true"></media-id>
  <media-reference nil="true"></media-reference>
  <name>TestCreative4Updated</name>
  <notes nil="true"></notes>
  <request-headers nil="true"></request-headers>
  <request-url nil="true"></request-url>
  <template-id type="integer" nil="true"></template-id>
  <updated-at type="datetime">2011-05-27T20:14:14-04:00</updated-at>
  <user-info nil="true"></user-info>
  <format-field-values type="array">
    <format-field-value type="CreativeFormatFieldValue">
      <created-at type="datetime">2011-05-27T20:09:56-04:00</created-at>
      <creative-id type="integer">4</creative-id>
      <format-field-id type="integer">7</format-field-id>
      <id type="integer">3</id>
      <updated-at type="datetime">2011-05-27T20:14:14-04:00</updated-at>
      <value>PID456</value>
      <format-field type="CreativeFormatFieldValue">
        <created-at type="datetime">2011-04-25T09:55:47-04:00</created-at>
        <description nil="true"></description>
        <id type="integer">7</id>
        <name>PID</name>
        <required type="boolean">true</required>
        <updated-at type="datetime">2011-04-25T09:55:47-04:00</updated-at>
      </format-field>
    </format-field-value>
  </format-field-values>
  <format>
    <id type="integer">8</id>
    <name>Standard-Pointroll_javascript</name>
  </format>
  <size>
    <id type="integer">9</id>
    <name>125X125</name>
  </size>
</creative>
```

Create a New Creative with a Media File Already in the Proper Location on the Disk

Here is an example of creating a creative with a media file already in the proper location on the disk:

```
POST: https://beta.aimatch.com/creative.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <advertiser-id type="integer">2</advertiser-id>
  <format-id type="integer">6</format-id>
  <location-type>asset</location-type>
  <name>CreativeWithMediaOnDiskAlready</name>
  <media-attributes>
    <asset-content-type>image/gif</asset-content-type>
    <asset-file-name>OnDiskAlready.gif</asset-file-name>
    <asset-file-size type="integer">261</asset-file-size>
    <asset-updated-at type="datetime">2011-04-13T13:11:01-04:00</asset-updated-at>
  </media-attributes>
  <size-id type="integer">10</size-id>
</creative>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <ad-text nil="true"></ad-text>
  <advertiser-id type="integer">2</advertiser-id>
  <alt nil="true"></alt>
  <alt-image-id type="integer" nil="true"></alt-image-id>
  <alt-image-reference nil="true"></alt-image-reference>
  <alt-image-type nil="true"></alt-image-type>
  <alt-image-url nil="true"></alt-image-url>
  <click-url nil="true"></click-url>
  <created-at type="datetime">2011-04-13T13:16:48-04:00</created-at>
  <description nil="true"></description>
  <duration type="integer" nil="true"></duration>
  <filename nil="true"></filename>
  <id type="integer">684</id>
  <image-url nil="true"></image-url>
  <location-type>asset</location-type>
  <lock-version type="integer">0</lock-version>
  <map nil="true"></map>
  <media-id type="integer">181</media-id>
  <media-reference>OnDiskAlready.gif</media-reference>
  <name>CreativeWithMediaOnDiskAlready</name>
  <notes nil="true"></notes>
  <request-headers nil="true"></request-headers>
  <request-url nil="true"></request-url>
  <size-id type="integer">5</size-id>
  <template-id type="integer" nil="true"></template-id>
  <updated-at type="datetime">2011-04-13T13:16:48-04:00</updated-at>
  <user-info nil="true"></user-info>
  <media>
    <advertiser-id type="integer">2</advertiser-id>
```

```
    <created-at type="datetime">2011-04-13T13:16:48-04:00</created-at>
    <id type="integer">181</id>
    <lock-version type="integer">0</lock-version>
    <updated-at type="datetime">2011-04-13T13:16:48-04:00</updated-at>
    <asset-file-name>OnDiskAlready.gif</asset-file-name>
    <asset-content-type>image/gif</asset-content-type>
    <asset-updated-at type="datetime">2011-04-13T13:11:01-04:00</asset-updated-at>
    <asset-file-size type="integer">261</asset-file-size>
    <asset_url type="string">/system/media/TestAdvertiser2/
OnDiskAlready.gif</asset_url>
  </media>
  <format-field-values type="array"/>
  <format>
    <id type="integer">6</id>
    <name>TestFormat6</name>
  </format>
  <size>
    <id type="integer">10</id>
    <name>728x90</name>
  </size>
</creative>
```


Creating a New Creative with a Base64-Encoded Media File and an Alt-image File Uploaded in a Single Request

Here is an example of creating a new creative with a base64-encoded media file and an alt-image file uploaded in a single request

```
POST: https://beta.aimatch.com/creative.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <advertiser-id type="integer">2</advertiser-id>
  <format-id type="integer">6</format-id>
  <location-type>upload</location-type>
  <name>new_creative_using_base64_media_upload</name>
  <size-id type="integer">6</size-id>
  <media-attributes>
    <asset type="file" content-type="image/gif" name="x.gif">
      <![CDATA[R0lGODlhEQAWAMQAAE1NTVNTU1lZWWNjY2xsbHNzc3p6eoaGhouLi5SUlJqa
mqGhoampqbOzs7m5uczMzNXV1d3d3ePj4+vr6/T09P7+/gAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAACH5BAkKABYALAAAAAARABYAAAWCYCWOZGme
aJoqEPooI5UAQ2Q+ApBQVQMIgRrpMQjkGJWJ4Re0VXBGwIGXNBgDBAkxOiUp
mYQiEEH1WoFXMopSMAIXqgnBHUikJG30+xSZA8NXcCQRYgEGWgGBI4RXBhMi
Dol7jECOJA2JAQANMpqWJZgBBU4JnyYNWSRlJxIqrq8jIQA7
]]>
    </asset>
  </media-attributes>
  <alt-image-attributes>
    <asset type="file" content-type="image/gif" name="x_alt_image.gif">
      <![CDATA[R0lGODlhEQAWAMQAAE1NTVNTU1lZWWNjY2xsbHNzc3p6eoaGhouLi5SUlJqa
mqGhoampqbOzs7m5uczMzNXV1d3d3ePj4+vr6/T09P7+/gAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAACH5BAkKABYALAAAAAARABYAAAWCYCWOZGme
aJoqEPooI5UAQ2Q+ApBQVQMIgRrpMQjkGJWJ4Re0VXBGwIGXNBgDBAkxOiUp
mYQiEEH1WoFXMopSMAIXqgnBHUikJG30+xSZA8NXcCQRYgEGWgGBI4RXBhMi
Dol7jECOJA2JAQANMpqWJZgBBU4JnyYNWSRlJxIqrq8jIQA7
```

```
]]>
    </asset>
  </alt-image-attributes>
</creative>
```

Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<creative>
  <ad-text nil="true"></ad-text>
  <advertiser-id type="integer">2</advertiser-id>
  <alt nil="true"></alt>
  <alt-image-id type="integer">170</alt-image-id>
  <alt-image-reference nil="true"></alt-image-reference>
  <alt-image-type nil="true"></alt-image-type>
  <alt-image-url nil="true"></alt-image-url>
  <click-url nil="true"></click-url>
  <created-at type="datetime">2011-04-12T21:43:45-04:00</created-at>
  <description nil="true"></description>
  <duration type="integer" nil="true"></duration>
  <filename nil="true"></filename>
  <id type="integer">677</id>
  <image-url nil="true"></image-url>
  <location-type>asset</location-type>
  <lock-version type="integer">0</lock-version>
  <map nil="true"></map>
  <media-id type="integer">169</media-id>
  <media-reference>x5.gif</media-reference>
  <name>new_creative_using_base64_media_upload</name>
  <notes nil="true"></notes>
  <request-headers nil="true"></request-headers>
  <request-url nil="true"></request-url>
  <template-id type="integer" nil="true"></template-id>
  <updated-at type="datetime">2011-04-12T21:43:45-04:00</updated-at>
  <user-info nil="true"></user-info>
  <alt-image>
    <advertiser-id type="integer">2</advertiser-id>
    <asset-content-type>image/gif</asset-content-type>
    <asset-file-name>x_alt_image.gif</asset-file-name>
    <asset-file-size type="integer">261</asset-file-size>
    <asset-updated-at type="datetime">2011-04-12T21:43:45-04:00</asset-updated-at>
    <created-at type="datetime">2011-04-12T21:43:45-04:00</created-at>
    <id type="integer">170</id>
    <lock-version type="integer">0</lock-version>
    <updated-at type="datetime">2011-04-12T21:43:45-04:00</updated-at>
  </alt-image>
  <media>
    <advertiser-id type="integer">2</advertiser-id>
    <asset-content-type>image/gif</asset-content-type>
    <asset-file-name>x.gif</asset-file-name>
    <asset-file-size type="integer">261</asset-file-size>
    <asset-updated-at type="datetime">2011-04-12T21:43:45-04:00</asset-updated-at>
    <created-at type="datetime">2011-04-12T21:43:45-04:00</created-at>
    <id type="integer">169</id>
    <lock-version type="integer">0</lock-version>
    <updated-at type="datetime">2011-04-12T21:43:45-04:00</updated-at>
  </media>
```

```
    <format-field-values type="array"/>
    <format>
      <id type="integer">6</id>
      <name>TestFormat6</name>
    </format>
    <size>
      <id type="integer">6</id>
      <name>120X240</name>
    </size>
  </creative>
  response: 201
```

Managing Creative Formats

Get a List of Creative Formats

Here is an example of getting a list of creative formats:

```
GET: https://beta.aimatch.com/formats.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<formats type="array">
  <format>
    <id type="integer">2</id>
    <name>Animated_GIF</name>
  </format>
  <format>
    <id type="integer">12</id>
    <name>Atlas_iFrame</name>
  </format>
  <format>
    <id type="integer">7</id>
    <name>Doubleclick</name>
  </format>
</formats>
```

This XML response provides 30 creative formats at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/formats.xml?page=7
```

Get a Specific Creative Format

Here is an example of getting a list of creative formats:

```
GET: https://beta.aimatch.com/formats/7.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<creative-format>
  <created-at type="datetime">2015-04-13T22:58:05-04:00</created-at>
  <description nil="true"/>
```

```
  <filename>CreativeFormat4.tmpl</filename>
  <id type="integer">1</id>
  <is-system-format type="boolean">false</is-system-format>
  <lock-version type="integer">0</lock-version>
  <mime-type-id type="integer" nil="true"/>
  <name>CreativeFormat4</name>
  <request-body nil="true"/>
  <request-headers nil="true"/>
  <request-method nil="true"/>
  <request-url nil="true"/>
  <response-template nil="true"/>
  <response-type nil="true"/>
  <updated-at type="datetime">2015-04-13T22:58:05-04:00</updated-at>
  <format-fields type="array">
    <format-field>
      <created-at type="datetime">2015-04-13T22:58:05-04:00</created-at>
      <description nil="true"/>
      <format-id type="integer">1</format-id>
      <id type="integer">1</id>
      <media type="boolean">false</media>
      <name>Field1</name>
      <position type="integer">0</position>
      <required type="boolean">false</required>
      <updated-at type="datetime">2015-04-13T22:58:05-04:00</updated-at>
    </format-field>
  </format-fields>
  <exposure-policy>
    <clicks-lifetime type="integer">1</clicks-lifetime>
    <clicks-period type="integer">3600</clicks-period>
    <clicks-quantity type="integer">1</clicks-quantity>
    <conversions-lifetime type="integer">1</conversions-lifetime>
    <conversions-period type="integer">3600</conversions-period>
    <conversions-quantity type="integer">1</conversions-quantity>
    <created-at type="datetime">2015-04-13T22:58:05-04:00</created-at>
    <id type="integer">1</id>
    <impressions-lifetime type="integer">1</impressions-lifetime>
    <impressions-per-session type="integer">1</impressions-per-session>
    <impressions-period type="integer">3600</impressions-period>
    <impressions-quantity type="integer">1</impressions-quantity>
    <updated-at type="datetime">2015-04-13T22:58:05-04:00</updated-at>
  </exposure-policy>
  <template-file-url>/system/templates/formats/CreativeFormat4.tmpl?1428980285<
/template-file-url>
</creative-format>
```


Get a Specific Format by Name

Here is an example of getting a specific format by name:

```
GET: http://beta.aimatch.com/creative_formats/CreativeFormat5.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<creative-format>
  <created-at type="datetime">2015-04-13T22:58:06-04:00</created-at>
  <description nil="true"/>
```

```
    <filename>CreativeFormat5.tmpl</filename>
    <id type="integer">1</id>
    <is-system-format type="boolean">false</is-system-format>
    <lock-version type="integer">0</lock-version>
    <mime-type-id type="integer" nil="true"/>
    <name>CreativeFormat5</name>
    <request-body nil="true"/>
    <request-headers nil="true"/>
    <request-method nil="true"/>
    <request-url nil="true"/>
    <response-template nil="true"/>
    <response-type nil="true"/>
    <updated-at type="datetime">2015-04-13T22:58:06-04:00</updated-at>
    <format-fields type="array">
      <format-field>
        <created-at type="datetime">2015-04-13T22:58:06-04:00</created-at>
        <description nil="true"/>
        <format-id type="integer">1</format-id>
        <id type="integer">1</id>
        <media type="boolean">false</media>
        <name>Field2</name>
        <position type="integer">0</position>
        <required type="boolean">false</required>
        <updated-at type="datetime">2015-04-13T22:58:06-04:00</updated-at>
      </format-field>
    </format-fields>
    <exposure-policy>
      <clicks-lifetime type="integer">1</clicks-lifetime>
      <clicks-period type="integer">3600</clicks-period>
      <clicks-quantity type="integer">1</clicks-quantity>
      <conversions-lifetime type="integer">1</conversions-lifetime>
      <conversions-period type="integer">3600</conversions-period>
      <conversions-quantity type="integer">1</conversions-quantity>
      <created-at type="datetime">2015-04-13T22:58:06-04:00</created-at>
      <id type="integer">1</id>
      <impressions-lifetime type="integer">1</impressions-lifetime>
      <impressions-per-session type="integer">1</impressions-per-session>
      <impressions-period type="integer">3600</impressions-period>
      <impressions-quantity type="integer">1</impressions-quantity>
      <updated-at type="datetime">2015-04-13T22:58:06-04:00</updated-at>
    </exposure-policy>
    <template-file-url>/system/templates/formats/CreativeFormat5.tmpl?1428980286<
  /template-file-url>
  </creative-format>
```


Create a Format with an Uploaded Template

Here is an example of creating a format with a template:

```
POST: http://beta.aimatch.com/creative_formats.xml
```

Here is the XML request:


```
<?xml version="1.0"?>
<creative-format>
  <name>Format1</name>
  <template-type>upload</template-type>
```

```
  <template-file type="file" content-type="text/html" name="format.html">
<![CDATA[PHA+VGVzdCBUZW1wbGF0ZTwvcD4=]]></template-file>
</creative-format>
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<creative-format>
  <created-at type="datetime">2015-04-13T22:58:07-04:00</created-at>
  <description nil="true"/>
  <filename>format.html</filename>
  <id type="integer">1</id>
  <is-system-format type="boolean">false</is-system-format>
  <lock-version type="integer">0</lock-version>
  <mime-type-id type="integer" nil="true"/>
  <name>Format1</name>
  <request-body nil="true"/>
  <request-headers nil="true"/>
  <request-method nil="true"/>
  <request-url nil="true"/>
  <response-template nil="true"/>
  <response-type nil="true"/>
  <updated-at type="datetime">2015-04-13T22:58:07-04:00</updated-at>
  <format-fields type="array"/>
  <template-file-url>/system/templates/formats/format.html?1428980287
</template-file-url>
</creative-format>
```

Create a Format with Template Text

Here is an example of creating a format with template text:

```
POST: http://beta.aimatch.com/creative_formats.xml
```

Here is the XML request:

```
<?xml version="1.0"?>
  <creative-format>
  <name>Format2</name>
  <template-type>text</template-type>
  <template-text><![CDATA[<p>Test Template</p>]]></template-text>
</creative-format>
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<creative-format>
  <created-at type="datetime">2015-04-13T22:58:08-04:00</created-at>
  <description nil="true"/>
  <filename>Format2.tmpl</filename>
  <id type="integer">1</id>
  <is-system-format type="boolean">false</is-system-format>
  <lock-version type="integer">0</lock-version>
  <mime-type-id type="integer" nil="true"/>
  <name>Format2</name>
  <request-body nil="true"/>
  <request-headers nil="true"/>
  <request-method nil="true"/>
  <request-url nil="true"/>
```

```
  <response-template nil="true"/>
  <response-type nil="true"/>
  <updated-at type="datetime">2015-04-13T22:58:08-04:00</updated-at>
  <format-fields type="array"/>
  <template-file-url>/system/templates/formats/Format2.tmpl?1428980288<
/template-file-url>
</creative-format>
```

Create a Format with a Network Template

Here is an example of creating a format with a network template:

```
POST: http://beta.aimatch.com/creative_formats.xml
```

Here is the XML request:

```
<?xml version="1.0"?>
<creative-format>
  <name>Format3</name>
  <template-type>network</template-type>
  <request-url>http://example.com/%%QUERY%%</request-url>
  <request-headers>User-Agent: %%USERAGENT%%</request-headers>
</creative-format>
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<creative-format>
  <created-at type="datetime">2015-04-13T22:58:09-04:00</created-at>
  <description nil="true"/>
  <filename nil="true"/>
  <id type="integer">1</id>
  <is-system-format type="boolean">false</is-system-format>
  <lock-version type="integer">0</lock-version>
  <mime-type-id type="integer" nil="true"/>
  <name>Format3</name>
  <request-body nil="true"/>
  <request-headers>User-Agent: %%USERAGENT%%</request-headers>
  <request-method nil="true"/>
  <request-url>http://example.com/%%QUERY%%</request-url>
  <response-template nil="true"/>
  <response-type nil="true"/>
  <updated-at type="datetime">2015-04-13T22:58:09-04:00</updated-at>
  <format-fields type="array"/>
  <template-file-url/>
</creative-format>
```

Update Format

Here is an example of updating a format:

```
PUT: http://beta.aimatch.com/creative_formats/1.xml
```

Here is the XML request:

```
<?xml version="1.0"?>
<creative-format>
  <name>CreativeFormat6</name>
  <description>Adipisci incidunt voluptas qui soluta.</description>
```

```
  <lock-version>0</lock-version>
</creative-format>
```

Update Format by Name

Here is an example of updating a format by name:

```
PUT: http://beta.aimatch.com/creative_formats/CreativeFormat7.xml
```

Here is the XML request:

```
<?xml version="1.0"?>
<creative-format>
  <name>CreativeFormat7</name>
  <description>Dignissimos voluptate enim velit perferendis qui.</description>
  <lock-version>0</lock-version>
</creative-format>
```

Update Format and Nested Format Fields

Here is an example of updating a format and any nested format fields:

```
PUT: http://beta.aimatch.com/creative_formats/1.xml
```

Here is the XML request:

```
<?xml version="1.0"?>
<creative-format>
  <lock-version>0</lock-version>
  <format-fields-attributes type="array">
    <format-field>
      <id>1</id>
      <name>Field6</name>
      <description>Minima laudantium quod expedita distinctio culpa delectus.
</description>
    </format-field>
  </format-fields-attributes>
</creative-format>
```

Update Format and a Nested Exposure Policy

Here is an example of updating a format and a nested exposure policy:

```
PUT: http://beta.aimatch.com/creative_formats/1.xml
```

Here is the XML request:

```
<?xml version="1.0"?>
<creative-format>
  <lock-version>0</lock-version>
  <exposure-policy-attributes>
    <id>1</id>
    <impressions-quantity>2</impressions-quantity>
  </exposure-policy-attributes>
</creative-format>
```

Update Format and Nested Format Tag Tokens

Here is an example of updating a format and nested format tag tokens:

```
PUT: http://beta.aimatch.com/creative_formats/1.xml
```

Here is the XML request:

```
<?xml version="1.0"?>
<creative-format>
  <lock-version>0</lock-version>
  <format-tag-tokens-attributes>
    <id>1</id>
    <token>one.two.three</token>
  </format-tag-tokens-attributes>
</creative-format>
```

Delete Format

Here is an example of deleting a format:

```
DELETE http://beta.aimatch.com/creative_formats/1.xml
```

Managing Media

Get a List of Media

Here is an example of getting a list of media:

```
GET: https://beta.aimatch.com/media.xml
```

Here is the XML response:

```
<media type="array">
  <media>
    <id type="integer">1</id>
    <asset_url type="string">/system/media/TestAdvertiser2/TestMedia1.gif</asset_url>
  </media>
  <media>
    <id type="integer">2</id>
    <asset_url type="string">/system/media/TestAdvertiser2/TestMedia2.gif</asset_url>
  </media>
  <media>
    <id type="integer">3</id>
    <asset_url type="string">/system/media/TestAdvertiser2/TestMedia3.gif</asset_url>
  </media>
</media>
```

This XML response provides 30 media at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/media.xml?page=3
```

Get Specific Media

Here is an example of getting specific media:

```
GET: https://beta.aimatch.com/media/3.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<media>
  <advertiser-id type="integer">2</advertiser-id>
  <asset-content-type>image/gif</asset-content-type>
  <asset-file-name>x.gif</asset-file-name>
  <asset-file-size type="integer">267</asset-file-size>
  <asset-updated-at type="datetime">2010-07-02T11:19:07-04:00</asset-updated-at>
      <public-asset-url>https://content.aimatch.com/customer/2/x.gif</public-asset-url>
  <created-at type="datetime">2010-10-04T16:27:10-04:00</created-at>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <updated-at type="datetime">2010-10-04T16:27:10-04:00</updated-at>
</media>
```


Create New Media with a Media File Already in the Proper Location on the Disk

Here is an example of creating new media with a media file already in the proper location on the disk:

```
POST: https://beta.aimatch.com/media.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<media>
  <asset-content-type>image/gif</asset-content-type>
  <asset-file-name>OnDiskAlready.gif</asset-file-name>
  <asset-file-size type="integer">261</asset-file-size>
  <asset-updated-at type="datetime">2011-04-13T13:11:01-04:00</asset-updated-at>
</media>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<media>
  <advertiser-id type="integer">2</advertiser-id>
  <created-at type="datetime">2011-04-13T13:16:48-04:00</created-at>
  <id type="integer">181</id>
  <lock-version type="integer">0</lock-version>
  <updated-at type="datetime">2011-04-13T13:16:48-04:00</updated-at>
  <asset-file-name>OnDiskAlready.gif</asset-file-name>
  <asset-content-type>image/gif</asset-content-type>
  <asset-updated-at type="datetime">2011-04-13T13:11:01-04:00</asset-updated-at>
  <asset-file-size type="integer">261</asset-file-size>
  <asset_url type="string">/system/media/TestAdvertiser2/OnDiskAlready.gif</asset_url>
</media>
```

Create New Media with a Base64-encoded GIF File Uploaded

Here is an example of creating new media with a base64-encoded GIF file uploaded:

```
POST: https://beta.aimatch.com/media.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<media>
  <advertiser-id type="integer">2</advertiser-id>
  <asset type="file" content-type="image/gif" name="x.gif">
    <![CDATA[R0lGODlhEQAWAMQAAE1NTVNTU1lZWWNjY2xsbHNzc3p6eoaGhouLi5SUlJqa
mqGhoampqbOzs7m5uczMzNXV1d3d3ePj4+vr6/T09P7+/gAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAACH5BAkKABYALAAAAAARABYAAAWCYCWOZGme
aJoqEPooI5UAQ2Q+ApBQVQMIgRrpMQjkGJWJ4Re0VXBGwIGXNBgDBAkxOiUp
mYQiEEH1WoFXMopSMAIXqgnBHUikJG30+xSZA8NXcCQRYgEGWgGBI4RXBhMi
Dol7jECOJA2JAQANMpqWJZgBBU4JnyYNWSRlJxIqrq8jIQA7
]]>
  </asset>
</media>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<media>
  <advertiser-id type="integer">2</advertiser-id>
  <asset-content-type>image/gif</asset-content-type>
  <asset-file-name>x.gif</asset-file-name>
  <asset-file-size type="integer">261</asset-file-size>
  <asset-updated-at type="datetime">2011-04-12T21:43:45-04:00</asset-updated-at>
  <created-at type="datetime">2011-04-12T21:43:45-04:00</created-at>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <updated-at type="datetime">2011-04-12T21:43:45-04:00</updated-at>
</media>
```


Get Specific Media That Includes the File Contents Base64-encoded

Here is an example of getting specific media that includes the file contents base64-encoded:

```
GET: https://beta.aimatch.com/media/3.xml?encode_attachments=true
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<media>
  <id type="integer">3</id>
  <asset type="file" name="x.gif" content-type="image/gif">
    <![CDATA[R0lGODlhEQAWAMQAAE1NTVNTU1lZWWNjY2xsbHNzc3p6eoaGhouLi5SUlJqa
mqGhoampqbOzs7m5uczMzNXV1d3d3ePj4+vr6/T09P7+/gAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAACH5BAkKABYALAAAAAARABYAAAWCYCWOZGme
aJoqEPooI5UAQ2Q+ApBQVQMIgRrpMQjkGJWJ4Re0VXBGwIGXNBgDBAkxOiUp
mYQiEEH1WoFXMopSMAIXqgnBHUikJG30+xSZA8NXcCQRYgEGWgGBI4RXBhMi
Dol7jECOJA2JAQANMpqWJZgBBU4JnyYNWSRlJxIqrq8jIQA7
]]>
  </asset>
</media>
```

Managing Advertiser Creative Templates

Get a List of Advertiser Creative Templates

Here is an example of getting a list of advertiser creative templates:

```
GET: https://beta.aimatch.com/advertiser_creative_templates.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<advertiser-creative-templates type="array">
  <advertiser-creative-template>
    <id type="integer">1</id>
    <asset_url type="string">/system/templates/creatives/TestAdvertiser2/
test_1.html</asset_url>
  </advertiser-creative-template>
  <advertiser-creative-template>
    <id type="integer">2</id>
    <asset_url type="string">/system/templates/creatives/TestAdvertiser2/
test_2.html</asset_url>
  </advertiser-creative-template>
  <advertiser-creative-template>
    <id type="integer">3</id>
    <asset_url type="string">/system/templates/creatives/TestAdvertiser2/
test_3.html</asset_url>
  </advertiser-creative-template>
</advertiser-creative-templates>
```


This XML response provides 30 advertiser creative templates at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/advertiser_creative_templates.xml?page=3
```

Get a Specific Advertiser Creative Template

Here is an example of getting a specific advertiser creative template:

```
GET: https://beta.aimatch.com/advertiser_creative_templates/3.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<advertiser-creative-template>
  <advertiser-id type="integer">2</advertiser-id>
  <asset-content-type>text/html</asset-content-type>
  <asset-file-name>test_3.html</asset-file-name>
  <asset-file-size type="integer">84</asset-file-size>
  <asset-updated-at type="datetime">2011-04-15T15:29:14-04:00</asset-updated-at>
  <created-at type="datetime">2011-04-15T15:29:15-04:00</created-at>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <mime-type-id type="integer" nil="true"></mime-type-id>
  <updated-at type="datetime">2011-04-15T15:29:15-04:00</updated-at>
```

```
</advertiser-creative-template>
```

Create a New Advertiser Creative Template with the Text Specified

Here is an example of creating a new advertiser creative template with the text specified:

```
POST: https://beta.aimatch.com/advertiser_creative_templates.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<advertiser_creative_template>
  <template_type>text</template_type>
  <template_name>test_4.txt</template_name>
  <advertiser_id>2</advertiser_id>
  <template_text>Example text template.</template_text>
</advertiser_creative_template>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<advertiser-creative-template>
  <advertiser-id type="integer">2</advertiser-id>
  <asset-content-type>text/html</asset-content-type>
  <asset-file-name>test_4.txt</asset-file-name>
  <asset-file-size type="integer">22</asset-file-size>
  <asset-updated-at type="datetime">2011-04-15T15:47:48-04:00</asset-updated-at>
  <created-at type="datetime">2011-04-15T15:47:48-04:00</created-at>
  <id type="integer">4</id>
  <lock-version type="integer">0</lock-version>
  <mime-type-id type="integer" nil="true"></mime-type-id>
  <updated-at type="datetime">2011-04-15T15:47:48-04:00</updated-at>
</advertiser-creative-template>
```


Creating a New Advertiser Creative Template with Base64-encoded HTML File

Here is an example of creating a new advertiser creative template with base64-encoded HTML file:

```
POST: https://beta.aimatch.com/advertiser_creative_templates.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<advertiser-creative-template>
  <template_type>upload</template_type>
  <template_name>test_5.html</template_name>
  <advertiser_id type="integer">2</advertiser_id>
  <asset type="file" name="test_5.html" content-type="text/html">
    <![CDATA[PGh0bWw+CjxoZWFkPgo8dGl0bGU+aGVsbG8gd29ybGQ8L3RpdGxlPgo8L2hl
YWQ+Cjxib2R5PgpoZWxsbyB3b3JsZCEKPC9ib2R5Pgo8aHRtbD4K
]]>
  </asset>
</advertiser-creative-template>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<advertiser-creative-template>
  <advertiser-id type="integer">2</advertiser-id>
  <asset-content-type>text/html</asset-content-type>
  <asset-file-name>test_5.html</asset-file-name>
```

```
  <asset-file-size type="integer">84</asset-file-size>
  <asset-updated-at type="datetime">2011-04-15T15:29:14-04:00</asset-updated-at>
  <created-at type="datetime">2011-04-15T15:29:15-04:00</created-at>
  <id type="integer">5</id>
  <lock-version type="integer">0</lock-version>
  <mime-type-id type="integer" nil="true"></mime-type-id>
  <updated-at type="datetime">2011-04-15T15:29:15-04:00</updated-at>
</advertiser-creative-template>
```


Get a Specific Advertiser Creative Template That Includes the File Contents Base64–encoded

Here is an example of getting a specific advertiser creative template including the file contents base64–encoded:

```
GET: https://beta.aimatch.com/advertiser_creative_templates/
4.xml?encode_attachments=true
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<advertiser-creative-template>
  <id type="integer">5</id>
  <asset type="file" name="test_5.html" content-type="text/html">
    <![CDATA[PGh0bWw+CjxoZWFkPgo8dGl0bGU+aGVsbG8gd29ybGQ8L3RpdGxlPgo8L2hl
YWQ+Cjxib2R5PgpoZWxsbyB3b3JsZCEKPC9ib2R5Pgo8aHRtbD4K
]]>
  </asset>
</advertiser-creative-template>
```


Managing Sites

Get a List of Sites

Here is an example of getting a list of sites:

```
GET: https://beta.aimatch.com/sites.xml
```

Here is the XML response:


```
<sites type="array">
  <site>
    <id type="integer">4</id>
    <name>TestSite0</name>
  </site>
  <site>
    <id type="integer">5</id>
    <name>TestSite1</name>
  </site>
  <site>
    <id type="integer">14</id>
    <name>TestSite10</name>
  </site>
</sites>
```

This XML response provides 30 sites at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/sites.xml?page=3
```

Get a Specific Site

Here is an example of getting a specific site:

```
GET: https://beta.aimatch.com/sites/14.xml
```

Here is the XML response:

```
<site>
  <created-at type="datetime">2010-08-31T13:49:39-04:00</created-at>
  <description>example site</description>
  <id type="integer">14</id>
  <lock-version type="integer">0</lock-version>
  <name>TestSite10</name>
  <revenue-cut type="decimal">1.4</revenue-cut>
  <updated-at type="datetime">2010-08-31T13:49:39-04:00</updated-at>
</site>
```

Create a Site

Here is an example of creating a site:

```
POST: https://beta.aimatch.com/sites.xml
```

```
<site>
  <name>My-New-Site</name>
  <description>example new site</description>
  <revenue-cut>1.79</revenue-cut>
</site>
```

Response (HTTP status: 201 Created):

```
<site>
  <created-at type="datetime">2010-09-14T11:41:48-04:00</created-at>
  <description>example new site</description>
  <id type="integer">57</id>
  <lock-version type="integer">0</lock-version>
  <name>My-New-Site</name>
  <revenue-cut>1.79</revenue-cut>
  <updated-at type="datetime">2010-09-14T11:41:48-04:00</updated-at>
</site>
```

Update a Site

Here is an example of updating a site:

```
PUT: https://beta.aimatch.com/sites/14.xml
```

```
<site>
  <lock-version type="integer">0</lock-version>
  <name>TestSite10</name>
  <description>example updating a site</description>
  <revenue-cut>2.5</revenue-cut>
```

```
</site>
```

The response will not contain anything, but the status code will be 200.

Managing Tags, TagValues, GeoTags, GeoTagValues, and Tag Groups

Get a List of Tags

Here is an example of getting a list of tags:

```
GET: https://beta.aimatch.com/tags.xml
```

Here is the XML response:

```
<tags type="array">
  <tag>
    <id type="integer">1</id>
    <name>TestTag1</name>
  </tag>
  <tag>
    <id type="integer">2</id>
    <name>TestTag2</name>
  </tag>
  <tag>
    <id type="integer">3</id>
    <name>TestTag3</name>
  </tag>
</tags>
```

This XML response provides 30 tags at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/tags.xml?page=3
```

Get a List of Tags in a Tag Group

Here is an example of getting a list of tags in a tag group:

```
GET: https://beta.aimatch.com/tags.xml?search%5Btag_group_name_like%5D=DMP
```

Here is the XML response:

```
<<?xml version="1.0" encoding="UTF-8"?>
<tags type="array">
  <tag>
    <datatype>String</datatype>
    <id type="integer">3</id>
    <name>Tag1</name>
    <tag-group-id type="integer">1</tag-group-id>
  </tag>
  <tag>
    <datatype>String</datatype>
    <id type="integer">4</id>
    <name>Tag2</name>
```

```
    <tag-group-id type="integer">1</tag-group-id>
  </tag>
  <tag>
    <datatype>String</datatype>
    <id type="integer">5</id>
    <name>Tag3</name>
    <tag-group-id type="integer">1</tag-group-id>
  </tag>
</tags>
```

This XML response provides 30 tags at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/tags.xml?page=3&search%5Btag_group_name_like%5D=DMP
```

Get a Specific Tag

Here is an example of getting a specific tag:

```
GET: https://beta.aimatch.com/tags/3.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2011-02-24T16:55:53-05:00</created-at>
  <dynamic-target type="boolean">false</dynamic-target>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTag3</name>
  <updated-at type="datetime">2011-02-24T16:55:53-05:00</updated-at>
  <sites type="array">
    <site>
      <created-at type="datetime">2011-02-24T17:43:41-05:00</created-at>
      <description>tag xml site 1</description>
      <id type="integer">123</id>
      <lock-version type="integer">0</lock-version>
      <name>TestTag3Site1</name>
      <revenue-cut type="decimal">5.25</revenue-cut>
      <site-id type="">123</site-id>
      <tag-id type="">3</tag-id>
      <updated-at type="datetime">2011-02-24T17:43:41-05:00</updated-at>
    </site>
  </sites>
  <values type="array">
    <value type="TagValue">
      <id type="integer">59</id>
      <value>TestTag3Value1</value>
    </value>
    <value type="TagValue">
      <id type="integer">60</id>
      <value>TestTag3Value1</value>
    </value>
    <value type="TagValue">
      <id type="integer">61</id>
      <value>TestTag3Value2</value>
```

```
    </value>
    <value type="TagValue">
      <id type="integer">62</id>
      <value>TestTag3Value3</value>
    </value>
  </values>
</tag>
```


Get a Specific Tag Where the Tag Has Thousands of Tag Values

Here is an example of getting a specific tag where the tag has thousands of tag values:

```
GET: https://beta.aimatch.com/tags/3.xml?format_tag_values=csv
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2011-02-24T16:55:53-05:00</created-at>
  <dynamic-target type="boolean">false</dynamic-target>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTag3</name>
  <updated-at type="datetime">2011-02-24T16:55:53-05:00</updated-at>
  <sites type="array">
    <site>
      <created-at type="datetime">2011-02-24T17:43:41-05:00</created-at>
      <description>tag xml site 1</description>
      <id type="integer">123</id>
      <lock-version type="integer">0</lock-version>
      <name>TestTag3Site1</name>
      <revenue-cut type="decimal">5.25</revenue-cut>
      <site-id type="">123</site-id>
      <tag-id type="">3</tag-id>
      <updated-at type="datetime">2011-02-24T17:43:41-05:00</updated-at>
    </site>
  </sites>
  <values-names>TestTag3Value1,TestTag3Value2,TestTag3Value3,TestTag3Value4,...,
testtag3value99999</value-names>
</tag>
```


Create a Tag

Here is an example of creating a tag:

```
POST: https://beta.aimatch.com/tags.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <dynamic-target type="boolean">false</dynamic-target>
  <name>TestTag4</name>
  <values-attributes type="array">
    <value>
      <value>TestTag4Value1</value>
    </value>
    <value>
      <value>TestTag4Value2</value>
```

```
    </values>
    <value>
      <value>TestTag4Value3</value>
    </value>
  </values-attributes>
  <sites-attributes type="array">
    <site>
      <description nil="true">Example of a site nested in an tag</description>
      <name>TestTag4Site1</name>
      <revenue-cut type="decimal">6.25</revenue-cut>
    </site>
    <site>
      <name>TestTag4Site2</name>
      <revenue-cut type="decimal">10.5</revenue-cut>
    </site>
  </sites-attributes>
</tag>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2011-03-03T14:08:55-05:00</created-at>
  <dynamic-target type="boolean">false</dynamic-target>
  <id type="integer">4</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTag4</name>
  <updated-at type="datetime">2011-03-03T14:08:55-05:00</updated-at>
  <sites type="array">
    <site>
      <created-at type="datetime">2011-03-03T14:08:55-05:00</created-at>
      <description>Example of a site nested in an tag</description>
      <id type="integer">126</id>
      <lock-version type="integer">0</lock-version>
      <name>TestTag4Site1</name>
      <revenue-cut type="decimal">6.25</revenue-cut>
      <updated-at type="datetime">2011-03-03T14:08:55-05:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-03-03T14:08:55-05:00</created-at>
      <description nil="true"></description>
      <id type="integer">127</id>
      <lock-version type="integer">0</lock-version>
      <name>TestTag4Site2</name>
      <revenue-cut type="decimal">10.5</revenue-cut>
      <updated-at type="datetime">2011-03-03T14:08:55-05:00</updated-at>
    </site>
  </sites>
  <values type="array">
    <value type="TagValue">
      <id type="integer">64</id>
      <value>TestTag4Value1</value>
    </value>
    <value type="TagValue">
      <id type="integer">65</id>
      <value>TestTag4Value2</value>
    </value>
```

```
    <value type="TagValue">
      <id type="integer">66</id>
      <value>TestTag4Value3</value>
    </value>
  </values>
</tag>
```

Creating a Tag with Bulk Value Creation

Here is an example of creating a tag with bulk value creation:

```
POST: https://beta.aimatch.com/tags.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <name>TestTag5</name>
  <new-value-names>TestTag5Value1,TestTag5Value2,TestTag5Value3</new-value-names>
</tag>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2011-12-01T07:15:47+13:00</created-at>
  <dynamic-target type="boolean">false</dynamic-target>
  <id type="integer">8</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTag5</name>
  <tagsum-logging type="boolean">false</tagsum-logging>
  <updated-at type="datetime">2011-12-01T07:15:47+13:00</updated-at>
  <sites type="array"/>
  <values type="array">
    <value type="TagValue">
      <id type="integer">10571</id>
      <value>TestTag5Value1</value>
    </value>
    <value type="TagValue">
      <id type="integer">10572</id>
      <value>TestTag5Value2</value>
    </value>
    <value type="TagValue">
      <id type="integer">10573</id>
      <value>TestTag5Value3</value>
    </value>
  </values>
</tag>
```

Creating a Tag with Bulk Value Creation and Specifying That You Want the Results to Contain the Tag Values as CSV

Here is an example of creating a tag with bulk value creation and specifying that you want the results to contain the tag values as CSV:

```
POST: https://beta.aimatch.com/tags.xml?format_tag_values=csv
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <name>TestTag6</name>
```

```
  <new-value-names>TestTag5Value1,TestTag5Value2,TestTag5Value3</new-value-names>
</tag>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2011-12-01T07:15:47+13:00</created-at>
  <dynamic-target type="boolean">false</dynamic-target>
  <id type="integer">8</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTag5</name>
  <tagsum-logging type="boolean">false</tagsum-logging>
  <updated-at type="datetime">2011-12-01T07:15:47+13:00</updated-at>
  <sites type="array"/>
  <value-names>TestTag5Value1,TestTag5Value2TestTag5Value3</value-names>
</tag>
```

Bulk Creating Tag Values for an Existing Tag

Here is an example of bulk creating tag values for an existing tag:

```
POST: https://beta.aimatch.com/tags/4/values.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <new_value_names>New Tag Value1|Tag Value 1,New Tag Value2|Tag Value 2</new_value_name
</tag>
```

The response will not contain anything, but the status code will be 200.

The tag now looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
  <tag>
    <created-at type="datetime">2017-09-13T16:37:34-04:00</created-at>
    <datatype>String</datatype>
    <dynamic-target type="boolean">true</dynamic-target>
    <id type="integer">4</id>
    <lock-version type="integer">0</lock-version>
    <name>TestTag4</name>
    <tag-group-id type="integer">4</tag-group-id>
    <tagsum-logging type="boolean">false</tagsum-logging>
    <updated-at type="datetime">2017-09-21T15:11:05-04:00</updated-at>
    <values type="array">
      <value type="TagValue">
        <display-name>tag value 1</display-name>
        <id type="integer">120</id>
        <value>new tag value1</value>
      </value>
      <value type="TagValue">
        <display-name>tag value 2</display-name>
        <id type="integer">121</id>
        <value>new tag value2</value>
      </value>
    </values>
    <sites-attributes type="array">
      <site>
        <description>Example of a site nested in an tag updated</description>
```

```
      <id type="integer">126</id>
      <lock-version type="integer">0</lock-version>
      <name>TestTag4Site1</name>
      <revenue-cut type="decimal">6.25</revenue-cut>
    </site>
    <site>
      <description nil="true">TestTag4Site2 description</description>
      <id type="integer">127</id>
      <lock-version type="integer">0</lock-version>
      <revenue-cut type="decimal">11.5</revenue-cut>
    </site>
  </sites-attributes>
</tag>
```

Updating a Tag and Its Associated Sites

Here is an example of updating a tag and its associated sites:

```
PUT: https://beta.aimatch.com/tags/4.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <dynamic-target type="boolean">true</dynamic-target>
  <id type="integer">4</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTag4</name>
  <sites-attributes type="array">
    <site>
      <description>Example of a site nested in an tag updated</description>
      <id type="integer">126</id>
      <lock-version type="integer">0</lock-version>
      <name>TestTag4Site1</name>
      <revenue-cut type="decimal">6.25</revenue-cut>
    </site>
    <site>
      <description nil="true">TestTag4Site2 description</description>
      <id type="integer">127</id>
      <lock-version type="integer">0</lock-version>
      <revenue-cut type="decimal">11.5</revenue-cut>
    </site>
  </sites-attributes>
</tag>
```

The response will not contain anything, but the status code will be 200.

The tag now looks like this:

```
GET: https://beta.aimatch.com/tags/4.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2011-03-03T14:08:55-05:00</created-at>
  <dynamic-target type="boolean">true</dynamic-target>
  <id type="integer">4</id>
  <lock-version type="integer">1</lock-version>
  <name>TestTag4</name>
  <updated-at type="datetime">2011-03-03T14:18:17-05:00</updated-at>
  <sites type="array">
```

```
    <site>
      <created-at type="datetime">2011-03-03T14:08:55-05:00</created-at>
      <description>Example of a site nested in an tag updated</description>
      <id type="integer">126</id>
      <lock-version type="integer">1</lock-version>
      <name>TestTag4Site1</name>
      <revenue-cut type="decimal">6.25</revenue-cut>
      <site-id type="">126</site-id>
      <tag-id type="">4</tag-id>
      <updated-at type="datetime">2011-03-03T14:18:17-05:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-03-03T14:08:55-05:00</created-at>
      <description>TestTag4Site2 description</description>
      <id type="integer">127</id>
      <lock-version type="integer">1</lock-version>
      <name>TestTag4Site2</name>
      <revenue-cut type="decimal">11.5</revenue-cut>
      <site-id type="">127</site-id>
      <tag-id type="">4</tag-id>
      <updated-at type="datetime">2011-03-03T14:18:17-05:00</updated-at>
    </site>
  </sites>
  <values type="array">
    <value type="TagValue">
      <id type="integer">64</id>
      <value>TestTag4Value1</value>
    </value>
    <value type="TagValue">
      <id type="integer">65</id>
      <value>TestTag4Value2</value>
    </value>
    <value type="TagValue">
      <id type="integer">66</id>
      <value>TestTag4Value3</value>
    </value>
  </values>
</tag>
```

Updating a Tag and Its Associated Tag Values

Here is an example of updating a tag and its associated tag values:

```
PUT: https://beta.aimatch.com/tags/4.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <name>TestTag4</name>
  <values-attributes type="array">
    <value>
      <value>NewTagValue1</value>
      <display-name>New Tag Value 1</display-name>
    </value>
    <value>
      <value>NewTagValue2</value>
    </value>
```

```
  </values-attributes>
</tag>
```

The response will not contain anything, but the status code will be 200.

The tag now looks like this:

```
GET: https://beta.aimatch.com/tags/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2017-09-21T14:41:35-04:00</created-at>
  <datatype>String</datatype>
  <dynamic-target type="boolean">false</dynamic-target>
  <id type="integer">16</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTag4</name>
  <tag-group-id type="integer" nil="true"/>
  <tagsum-logging type="boolean">false</tagsum-logging>
  <updated-at type="datetime">2017-09-21T14:41:35-04:00</updated-at>
  <values type="array">
    <value type="TagValue">
      <display-name>New Tag Value 1</display-name>
      <id type="integer">124</id>
      <value>NewTagValue1</value>
    </value>
    <value type="TagValue">
      <display-name nil="true"/>
      <id type="integer">125</id>
      <value>NewTagValue2</value>
    </value>
  </values>
  <sites type="array"/>
</tag>
```


Update a Tag (Using the Name instead of the ID) to Add New Tag Values in Bulk

Here is an example of updating a tag (using the name instead of the ID) in order to add new tag values in bulk:

```
PUT: https://beta.aimatch.com/tags/by_name.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <name>TestTag5</name>
  <new-value-names>TestTag5Value4,TestTag5Value5,TestTag5Value6</new-value-names>
</tag>
```

The response will not contain anything, but the status code will be 200.

Here is an example of the updated tag:

```
GET: https://beta.aimatch.com/tags/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <created-at type="datetime">2011-12-01T07:15:47+13:00</created-at>
  <dynamic-target type="boolean">true</dynamic-target>
  <id type="integer">8</id>
```

```
  <lock-version type="integer">1</lock-version>
  <name>TestTag5</name>
  <tagsum-logging type="boolean">true</tagsum-logging>
  <updated-at type="datetime">2011-12-01T07:24:11+13:00</updated-at>
  <sites type="array"/>
  <values type="array">
    <value type="TagValue">
      <id type="integer">10571</id>
      <value>TestTag5Value1</value>
    </value>
    <value type="TagValue">
      <id type="integer">10572</id>
      <value>TestTag5Value2</value>
    </value>
    <value type="TagValue">
      <id type="integer">10573</id>
      <value>TestTag5Value3</value>
    </value>
    <value type="TagValue">
      <id type="integer">10574</id>
      <value>TestTag5Value4</value>
    </value>
    <value type="TagValue">
      <id type="integer">10575</id>
      <value>TestTag5Value5</value>
    </value>
    <value type="TagValue">
      <id type="integer">10576</id>
      <value>TestTag5Value6</value>
    </value>
  </values>
</tag>
```


Updating a Tag's Tag Group

Here is an example of updating a tag's tag group:

```
PUT: https://beta.aimatch.com/tags/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<tag>
  <tag-group-id type="integer">5</tag-group-id>
</tag>
```


The response will not contain anything, but the status code will be 200.

Here is an example of the updated tag:

```
GET: https://beta.aimatch.com/tags/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
  <tag>
    <created-at type="datetime">2017-09-13T16:37:34-04:00</created-at>
    <datatype>String</datatype>
    <dynamic-target type="boolean">false</dynamic-target>
    <id type="integer">4</id>
    <lock-version type="integer">5</lock-version>
    <name>Tag2</name>
```

```
    <tag-group-id type="integer">4</tag-group-id>
    <tagsum-logging type="boolean">false</tagsum-logging>
    <updated-at type="datetime">2017-09-21T15:11:05-04:00</updated-at>
    <values type="array">
      <value type="TagValue">
        <display-name nil="true"/>
        <id type="integer">11</id>
        <value>TagValue11</value>
      </value>
    </values>
    <sites type="array"/>
  </tag>
```

Deleting a Tag Value

Here is an example of deleting a tag value:

```
DELETE: https://beta.aimatch.com/tags/4/values/120.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<hash>
  <id type="integer">120</id>
</hash>
response: 200
```

Specifying the Site-ID List (the Sites Must Already Exist)

You can specify a site-ID list in one of the following ways:

- by using a list of elements

```
<tag>
  <name>TestTag10</name>
  <site-id-list type="array">
    <site-id type="integer">1</site-id>
    <site-id type="integer">2</site-id>
  </site-id-list>
</tag>
```

- by using a comma separated string list

```
<tag>
  <name>TestTag10</name>
  <site-id-list>126,127</site-id-list>
</tag>
```

- by using an alternative form with site-IDs:

```
<tag>
  <site-ids type="array">
    <id>126</id>
    <id>127</id>
  </site-ids>
</tag>
```

Alternative Way of Specifying New Tag Values That Allow Bulk Tag Value Submission

Here is an example of using a comma separated string list to specify new tag values that allow bulk tag value submission:

```
<tag>
  <name>TestTag10</name>
  <new-value-names>TestTag5Value4,TestTag5Value5,TestTag5Value6</new-value-names>
</tag>
```

Get Tag-Values for a Tag

Here is an example getting tag-values for a tag:

```
GET: https://beta.aimatch.com/tags/4/values.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<tag-values type="array">
  <tag-value>
    <id type="integer">64</id>
    <value>TestTag4Value1</value>
  </tag-value>
  <tag-value>
    <id type="integer">65</id>
    <value>TestTag4Value2</value>
  </tag-value>
  <tag-value>
    <id type="integer">66</id>
    <value>TestTag4Value3</value>
  </tag-value>
</tag-values>
```


Get a List of Geo-Tags

Here is an example getting a list of geo-tags:

```
GET: https://beta.aimatch.com/geo_tags.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<geo-tags type="array">
  <geo-tag>
    <id type="integer">1</id>
    <name>geo_city</name>
  </geo-tag>
  <geo-tag>
    <id type="integer">2</id>
    <name>geo_region</name>
  </geo-tag>
  <geo-tag>
    <id type="integer">3</id>
    <name>geo_country</name>
  </geo-tag>
```

```
  <geo-tag>
    <id type="integer">4</id>
    <name>geo_metro-code</name>
  </geo-tag>
  <geo-tag>
    <id type="integer">5</id>
    <name>geo_conn-speed</name>
  </geo-tag>
  <geo-tag>
    <id type="integer">6</id>
    <name>geo_zip-code-text</name>
  </geo-tag>
  <geo-tag>
    <id type="integer">7</id>
    <name>geo_metro_code</name>
  </geo-tag>
  <geo-tag>
    <id type="integer">8</id>
    <name>geo_conn_speed</name>
  </geo-tag>
  <geo-tag>
    <id type="integer">9</id>
    <name>geo_zip_code_text</name>
  </geo-tag>
</geo-tags>
```

Get GeoTagValues for a Geo-Tag

Here is an example getting GeoTagValues for a Geo-Tag:

```
GET: https://beta.aimatch.com/geo_tags/5/values.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<geo-tag-values type="array">
  <geo-tag-value>
    <id type="integer">26459</id>
    <value>broadband</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26464</id>
    <value>cable</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26460</id>
    <value>dialup</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26463</id>
    <value>dsl</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26462</id>
    <value>mobile</value>
  </geo-tag-value>
  <geo-tag-value>
```

```
    <id type="integer">26458</id>
    <value>oc12</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26468</id>
    <value>oc3</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26469</id>
    <value>satellite</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26465</id>
    <value>t1</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26467</id>
    <value>t3</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26466</id>
    <value>wireless</value>
  </geo-tag-value>
  <geo-tag-value>
    <id type="integer">26461</id>
    <value>xdsl</value>
  </geo-tag-value>
</geo-tag-values>
```


Get a List of Tag Groups

Here is an example getting a list of tag groups:

```
GET https://beta.aimatch.com/ads/tag_groups.json
```

Here is the JSON response:


```
{
    "name": "tag_groups",
    "start": 1,
    "count": 2,
    "limit": 30,
    "version": "1.0",
    "links": [
        {
        "method": "GET",
        "rel": "self",
        "href": "http://dc.aimatch.com/ads/tag_groups.json",
        "uri": "/ads/tag_groups.json",
        "type": "application/vnd.sas.marketing.ads.collection+json"
        }
    ],
    "items": [
        {
            "id": 4,
            "name": "ExampleName1",
            "links": [
```

```
                {
                    "rel": "self",
                    "method": "get",
                    "href": "http://dc.aimatch.com/ads/tag_groups/4",
                    "uri": "/ads/tag_groups/4",
                    "type": "application/vnd.sas.marketing.ads.tag_group+json"
                },
                {
                    "rel": "up",
                    "method": "get",
                    "href": "http://dc.aimatch.com/ads/tag_groups",
                    "uri": "/ads/tag_groups",
                    "type": "application/vnd.sas.marketing.ads.collection+json"
                }
            ],
        },
        {
            "id": 3,
            "name": "ExampleName2",
            "links": [
                {
                    "rel": "self",
                    "method": "get",
                    "href": "http://dc.aimatch.com/ads/tag_groups/3",
                    "uri": "/ads/tag_groups/3",
                    "type": "application/vnd.sas.marketing.ads.tag_group+json"
                },
                {
                    "rel": "up",
                    "method": "get",
                    "href": "http://dc.aimatch.com/ads/tag_groups",
                    "uri": "/ads/tag_groups",
                    "type": "application/vnd.sas.marketing.ads.collection+json"
                }
            ],
        }
    ],
}
```

Get a Specific Tag Group

Here is an example of getting a specific tag group:

```
GET https://beta.aimatch.com/ads/tag_groups/1.json
```

Here is the JSON response:

```
{
    "id": 4,
    "name": "ExampleName",
    "links": [
        {
            "rel": "self",
            "method": "get",
            "href": "http://dc.aimatch.com/ads/tag_groups/4",
            "uri": "/ads/tag_groups/4",
            "type": "application/vnd.sas.marketing.ads.tag_group+json"
```

```
            },
            {
                "rel": "up",
                "method": "get",
                "href": "http://dc.aimatch.com/ads/tag_groups",
                "uri": "/ads/tag_groups",
                "type": "application/vnd.sas.marketing.ads.collection+json"
            }
        ]
    }
```

Create a Tag Group

Here is an example of creating a tag group:

```
POST https://beta.aimatch.com/ads/tag_groups.json
```

```
{
    "tag_group": {
        "name": "ExampleName"
    }
}
```

Update a Tag Group

Here is an example of updating a tag group

```
PUT https://beta.aimatch.com/ads/tag_groups/1.json


{
    "tag_group": {
        "name": "NOPE2"
    }
}
```

Delete a Tag Group

Here is an example of deleting a tag group:

```
DELETE https://beta.aimatch.com/ads/tag_groups/1.json
```

Managing Targets

Get a List of Targets

Here is an example of getting a list of targets:

```
GET: https://beta.aimatch.com/targets.xml
```

Here is the XML response:

```
<targets type="array">
  <target>
```

```
    <id type="integer">1</id>
    <name>TestTarget1</name>
  </target>
  <target>
    <id type="integer">2</id>
    <name>TestTarget2</name>
  </target>
  <target>
    <id type="integer">3</id>
    <name>TestTarget3</name>
  </target>
</targets>
```

This XML response provides 30 targets at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of URL with a page query parameter:

```
https://beta.aimatch.com/targets.xml?page=3
```

Get a Specific Target

Here is an example of getting a specific target:

```
GET: https://beta.aimatch.com/targets/3.xml
```

Here is the XML response:

```
<target>
  <created-at type="datetime">2011-02-17T15:48:34-05:00</created-at>
  <creator-id type="integer" nil="true"></creator-id>
  <description>TestTarget3 description</description>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTarget3</name>
  <updated-at type="datetime">2011-02-17T15:48:34-05:00</updated-at>
  <rules type="array">
    <rule type="TargetRule">
      <connector nil="true"></connector>
      <created-at type="datetime">2011-02-17T15:48:34-05:00</created-at>
      <id type="integer">9</id>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>DAYOFWEEK</left-value>
      <lock-version type="integer">0</lock-version>
      <operator>In</operator>
      <position type="integer">1</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>TUESDAY,FRIDAY</right-value>
      <target-id type="integer">3</target-id>
      <updated-at type="datetime">2011-02-17T15:48:34-05:00</updated-at>
    </rule>
    <rule type="TargetRule">
      <connector>AND</connector>
      <created-at type="datetime">2011-02-17T15:48:34-05:00</created-at>
      <id type="integer">10</id>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>SITE</left-value>
      <lock-version type="integer">0</lock-version>
```

```
      <operator>Equals</operator>
      <position type="integer">2</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>TestSite1</right-value>
      <target-id type="integer">3</target-id>
      <updated-at type="datetime">2011-02-17T15:48:34-05:00</updated-at>
    </rule>
  </rules>
</target>
```

Create a Target

Here is an example of creating a target:

```
POST: https://beta.aimatch.com/targets.xml
```

```
<target>
  <description>TestTarget4 description</description>
  <name>TestTarget4</name>
  <rules-attributes type="array">
    <rule>
      <connector nil="true"></connector>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>DAYOFWEEK</left-value>
      <operator>In</operator>
      <position type="integer">1</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY</right-value>
    </rule>
    <rule>
      <connector>AND</connector>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>SITE</left-value>
      <operator>Equals</operator>
      <position type="integer">2</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>TestSite2</right-value>
    </rule>
  </rules-attributes>
</target>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<target>
  <created-at type="datetime">2011-03-03T13:31:14-05:00</created-at>
  <creator-id type="integer">1</creator-id>
  <description>TestTarget4 description</description>
  <id type="integer">4</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTarget4</name>
  <updated-at type="datetime">2011-03-03T13:31:14-05:00</updated-at>
  <rules type="array">
    <rule type="TargetRule">
      <connector nil="true"></connector>
      <created-at type="datetime">2011-03-03T13:31:14-05:00</created-at>
      <id type="integer">23</id>
```

```
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>DAYOFWEEK</left-value>
      <lock-version type="integer">0</lock-version>
      <operator>In</operator>
      <position type="integer">1</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY</right-value>
      <target-id type="integer">4</target-id>
      <updated-at type="datetime">2011-03-03T13:31:14-05:00</updated-at>
    </rule>
    <rule type="TargetRule">
      <connector>AND</connector>
      <created-at type="datetime">2011-03-03T13:31:14-05:00</created-at>
      <id type="integer">24</id>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>SITE</left-value>
      <lock-version type="integer">0</lock-version>
      <operator>Equals</operator>
      <position type="integer">2</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>TestSite2</right-value>
      <target-id type="integer">4</target-id>
      <updated-at type="datetime">2011-03-03T13:31:14-05:00</updated-at>
    </rule>
  </rules>
</target>
```


Update a Target

Here is an example of updating a target:

```
PUT: https://beta.aimatch.com/targets/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<target>
  <description>updated TestTarget4 description</description>
  <id type="integer">4</id>
  <name>TestTarget4</name>
  <rules-attributes type="array">
    <rule>
      <connector nil="true"></connector>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <id type="integer">23</id>
      <left-value>DAYOFWEEK</left-value>
      <operator>In</operator>
      <position type="integer">1</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>SUNDAY,WEDNESDAY</right-value>
    </rule>
    <rule>
      <id type="integer">24</id>
      <connector>OR</connector>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>SITE</left-value>
      <operator>Equals</operator>
      <position type="integer">2</position>
```

```
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>TestSite3</right-value>
    </rule>
  </rules-attributes>
</target>
```

The response will not contain anything, but the status code will be 200.

The target now looks like this:

```
GET: https://beta.aimatch.com/targets/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<target>
  <created-at type="datetime">2011-03-03T13:31:14-05:00</created-at>
  <creator-id type="integer">1</creator-id>
  <description>updated TestTarget4 description</description>
  <id type="integer">4</id>
  <lock-version type="integer">1</lock-version>
  <name>TestTarget4</name>
  <updated-at type="datetime">2011-03-03T13:48:22-05:00</updated-at>
  <rules type="array">
    <rule type="TargetRule">
      <connector nil="true"></connector>
      <created-at type="datetime">2011-03-03T13:31:14-05:00</created-at>
      <id type="integer">23</id>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>DAYOFWEEK</left-value>
      <lock-version type="integer">1</lock-version>
      <operator>In</operator>
      <position type="integer">1</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>SUNDAY,WEDNESDAY</right-value>
      <target-id type="integer">10</target-id>
      <updated-at type="datetime">2011-03-03T13:48:22-05:00</updated-at>
    </rule>
    <rule type="TargetRule">
      <connector>OR</connector>
      <created-at type="datetime">2011-03-03T13:31:14-05:00</created-at>
      <id type="integer">24</id>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>SITE</left-value>
      <lock-version type="integer">1</lock-version>
      <operator>Equals</operator>
      <position type="integer">2</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>TestSite3</right-value>
      <target-id type="integer">10</target-id>
      <updated-at type="datetime">2011-03-03T13:48:22-05:00</updated-at>
    </rule>
  </rules>
</target>
```

Managing Tiers

Get a List of Tiers

Here is an example of getting a list of tiers:

```
GET: http://beta.aimatch.com/tiers.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
  <tiers type="array">
    <tier>
      <name>Tier1</name>
      <id type="integer">1</id>
    </tier>
  </tiers>
```

This XML response provides 30 tiers at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of URL with a page query parameter:

```
http://beta.aimatch.com/tiers.xml?page=2
```

Get a Specific Tier

Here is an example of getting a specific tier:

```
GET: http://beta.aimatch.com/tiers/3.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
  <tier>
    <prioritization-method>CPM</prioritization-method>
    <grace-period type="integer" nil="true"/>
    <flags type="integer">0</flags>
    <created-at type="datetime">2014-07-02T16:20:49-04:00</created-at>
    <updated-at type="datetime">2014-07-02T16:20:49-04:00</updated-at>
    <preemptible type="boolean">false</preemptible>
    <name>Tier2</name>
    <grace-period-priority type="decimal" nil="true"/>
    <rank type="integer">1</rank>
    <lock-version type="integer">0</lock-version>
    <active type="boolean">true</active>
    <id type="integer">3</id>
    <override-raq-rule type="boolean">false</override-raq-rule>
    <eliminate-duplicate-flights type="boolean">false</eliminate-duplicate-flights>
    <eliminate-duplicate-advertisers type="boolean">false</eliminate-duplicate-advertise
    <multi-variant type="boolean">false</multi-variant>
  </tier>
```

Create a Tier

Here is an example of creating a tier:

```
POST: http://beta.aimatch.com/tiers.xml
```

```
<?xml version="1.0"?>
  <tier>
    <name>Tier3</name>
    <active>true</active>
    <prioritization-method>Schedule</prioritization-method>
    <grace-period>10</grace-period>
    <grace-period-priority>0.5</grace-period-priority>
    <override-raq-rule>true</override-raq-rule>
    <eliminate-duplicate-flights>true</eliminate-duplicate-flights>
    <eliminate-duplicate-advertisers>true</eliminate-duplicate-advertisers>
    <multi-variant>true</multi-variant>
    <target-id>1</target-id>
    <rank>1</rank>
    <preemptible>true</preemptible>
  </tier>
```

Response (HTTP status: 201 Created):

```
Location: http://www.example.com/tiers/1
```

Update a Tier

Here is an example of updating a tier:

```
PUT: http://beta.aimatch.com/tiers/4.xml
```

```
<?xml version="1.0"?>
  <tier>
    <id>4</id>
    <name>Tier5</name>
  </tier>
```

The response will not contain anything, but the status code will be 200.

Managing Supertags

Get a List of Supertags

Here is an example of getting a list of supertags:

```
GET: http://beta.aimatch.com/supertag_values.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<supertag-values type="array">
  <supertag-value>
    <id type="integer">6</id>
    <pathinfo>/site=pb</pathinfo>
    <value>hottub1</value>
```

```
    </supertag-value>
    <supertag-value>
      <id type="integer">14</id>
      <pathinfo/>
      <value>hottub10</value>
    </supertag-value>
    <supertag-value>
      <id type="integer">15</id>
      <pathinfo/>
      <value>hottub11</value>
    </supertag-value>
  </supertag-values>
```

This XML response provides 30 supertags at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of URL with a page query parameter:

```
http://beta.aimatch.com/supertag_values.xml?page=2
```

Get a Specific Supertag

Here is an example of getting a specific supertag:

```
GET: http://beta.aimatch.com/supertag_values/6.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<supertag-value>
  <created-at type="datetime">2015-02-26T08:37:54+13:00</created-at>
  <id type="integer">6</id>
  <lock-version type="integer">0</lock-version>
  <pathinfo>/site=pb</pathinfo>
  <updated-at type="datetime">2015-02-26T08:37:54+13:00</updated-at>
  <value>hottub1</value>
</supertag-value>
```

Create a Supertag

Here is an example of creating a supertag:

```
POST: http://beta.aimatch.com/supertag_values.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<supertag-value>
  <pathinfo>site=sp/area=a</pathinfo>
  <value>super1</value>
</supertag-value>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<supertag-value>
  <created-at type="datetime">2015-02-28T09:05:21+13:00</created-at>
  <id type="integer">36</id>
  <lock-version type="integer">0</lock-version>
  <pathinfo>site=sp/area=a</pathinfo>
  <updated-at type="datetime">2015-02-28T09:05:21+13:00</updated-at>
  <value>super1</value>
```

```
</supertag-value>
```

Update a Supertag

Here is an example of updating a supertag:

```
PUT: http://beta.aimatch.com/supertag_values/36.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<supertag-value>
  <pathinfo>site=sp/area=b</pathinfo>
  <value>super1</value>
</supertag-value>
```

The response will not contain anything, but the status code will be 200.

The supertag now looks like this:

```
GET: http://beta.aimatch.com/supertag_values/36.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<supertag-value>
  <created-at type="datetime">2015-02-28T09:05:21+13:00</created-at>
  <id type="integer">36</id>
  <lock-version type="integer">1</lock-version>
  <pathinfo>site=sp/area=b</pathinfo>
  <updated-at type="datetime">2015-02-28T09:16:17+13:00</updated-at>
  <value>super1</value>
</supertag-value>
```

Create or Update Supertags in Batch

Here is an example of creating or updating supertags in batch:

```
POST: https://beta.aimatch.com/supertag_values.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<supertag-values type="array">
  <supertag-value>
    <pathinfo>site=sp/area=a</pathinfo>
    <value>super1</value>
  </supertag-value>
  <supertag-value>
    <pathinfo></pathinfo>
    <value>super2</value>
  </supertag-value>
</supertag-values>
response: 200 if no errors; otherwise 422 with errors
```

Delete a Supertag

Here is an example of deleting a supertag:

```
DELETE: http://beta.aimatch.com/supertag_values/36.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<hash>
  <id type="integer">36</id>
</hash>
```

```
response: 200
```

Managing Check Availability

Check Availability with a Line Item ID

```
POST: http://beta.aimatch.com/
check_inventory.xml?entity_name=line_item&flight_breakdown=1
```

The value of flight_breakdown can be 0 or 1. When flight_breakdown = 1, the response includes the contending flights.

```
<line-item>
  <id type="integer">9</id>
</line-item>
```

Here is the XML response:

```
<InventoryResponse>
  <JobId>82105f2947a6432f3819169763d7f921</JobId>
  <ProgressbarEnabled type="boolean">false</ProgressbarEnabled>
  <Email>[User email]</Email>
</InventoryResponse>
```

You can periodically issue the following request to check and retrieve the result.

```
GET: http://beta.aimatch.com/
check_inventory.xml?job_id=82105f2947a6432f3819169763d7f921
```

The XML response when the job does not run in the background:

```
<InventoryResponse>
  <Available type="integer">100000000</Available>
  <Total type="integer">200000000</Total>
  <Error nil="true"/>
  <Flights type="array">
    <Flight>
      <Id type="integer">16</Id>
      <CampaignId type="integer">10</CampaignId>
      <Name>men_and_pro_globo_3</Name>
      <Reserved type="integer">123456</Reserved>
      <Url>/flights/16/edit</Url>
    </Flight>
    <Flight>
      <Id type="integer">9</Id>
      <CampaignId type="integer">11</CampaignId>
      <Name>gran_health_burleigh_1</Name>
      <Reserved type="integer">123432</Reserved>
      <Url>/flights/9/edit</Url>
    </Flight>
  </Flights>
</InventoryResponse>
```

Check Availability Based on Line Item Attributes, No ID

```
POST: http://beta.aimatch.com/
```

```
check_inventory.xml?entity_name=line_item&flight_breakdown=1
```


```
<line-item>
  <start-date-date type="date">2015-01-28</start-date-date>
  <end-date-date type="date">2015-02-28</end-date-date>
  <product-id type="integer">2</product-id>
  <size-id type="integer">2</size-id>
  <site-ids type="array">
    <id>1</id>
    <id>11</id>
  </site-ids>
  <area-ids type="array"></area-ids>
  <tag-value-ids type="array">
    <id>1</id>
    <id>2</id>
    <id>41688</id>
  </tag-value-ids>
  <geo-tag-value-ids type="array">
    <id>5162</id>
    <id>146168</id>
  </geo-tag-value-ids>
  <targets-group-logic>AND</targets-group-logic>
  <target-ids type="array">
    <id>12</id>
    <id>52</id>
  </target-ids>
  <ad-hoc-target-attributes>
    <id type="integer">366</id>
  </ad-hoc-target-attributes>
  <exposure-policy-attributes>
    <id type="integer">2</id>
    <impressions-quantity type="integer">3</impressions-quantity>
    <impressions-period type="integer">10800</impressions-period>
    <impressions-per-session type="integer">10</impressions-per-session>
    <impressions-lifetime type="integer">100</impressions-lifetime>
  </exposure-policy-attributes>
</line-item>
```


Here is the XML response:


```
<InventoryResponse>
  <JobId>77b3dad97aa10862ec12097f633205a3</JobId>
  <ProgressbarEnabled type="boolean">false</ProgressbarEnabled>
  <Email>[User email]</Email>
</InventoryResponse>
```


You can periodically issue the following request to check and retrieve the result.

```
GET: http://beta.aimatch.com/
check_inventory.xml?job_id=77b3dad97aa10862ec12097f633205a3
```

The XML response when the job does not run in the background:


```
<InventoryResponse>
  <Available type="integer">100000000</Available>
  <Total type="integer">200000000</Total>
  <Error nil="true"/>
  <Flights type="array">
    <Flight>
```

```
      <Id type="integer">16</Id>
      <CampaignId type="integer">10</CampaignId>
      <Name>men_and_pro_globo_3</Name>
      <Reserved type="integer">123456</Reserved>
      <Url>/flights/16/edit</Url>
    </Flight>
    <Flight>
      <Id type="integer">9</Id>
      <CampaignId type="integer">11</CampaignId>
      <Name>gran_health_burleigh_1</Name>
      <Reserved type="integer">123432</Reserved>
      <Url>/flights/9/edit</Url>
    </Flight>
  </Flights>
</InventoryResponse>
```


Managing Products

Get a List of Products

Here is an example of getting a list of products:

```
GET: https://beta.aimatch.com/products.xml
```

Here is the XML response:


```
<products type="array">
  <product>
    <id type="integer">1</id>
    <name>TestProduct1</name>
  </product>
  <product>
    <id type="integer">2</id>
    <name>TestProduct2</name>
  </product>
  <product>
    <id type="integer">3</id>
    <name>TestProduct3</name>
  </product>
</products>
```


This XML response provides 30 products at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/products.xml?page=3
```

Get a Specific Product

Here is an example of getting a specific product:

```
GET: https://beta.aimatch.com/products/3.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<product>
  <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
  <description nil="true"></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal" nil="true"></floor-rate>
  <goal-rate type="decimal" nil="true"></goal-rate>
  <id type="integer">3</id>
  <list-rate type="decimal">5.25</list-rate>
  <lock-version type="integer">1</lock-version>
  <name>TestProduct3</name>
  <rate-style>additional</rate-style>
  <rate-type nil="true"></rate-type>
  <targeting-type>None</targeting-type>
  <targets-group-logic>OR</targets-group-logic>
  <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
  <tier>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
    <flags type="integer">0</flags>
    <grace-period type="integer" nil="true"></grace-period>
    <grace-period-priority type="decimal" nil="true"></grace-period-priority>
    <id type="integer">3</id>
    <lock-version type="integer">0</lock-version>
    <name>House</name>
    <preemptible type="boolean">false</preemptible>
    <prioritization-method>Schedule</prioritization-method>
    <rank type="integer">3</rank>
    <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
  </tier>
  <sizes type="array">
    <size>
      <id type="integer">10</id>
      <name>160x230</name>
    </size>
  </sizes>
  <targets type="array">
    <target>
      <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
      <creator-id type="integer" nil="true"></creator-id>
      <description nil="true"></description>
      <id type="integer">3</id>
      <lock-version type="integer">0</lock-version>
      <name>rfp_3_prop_1_prod_1_tar_1</name>
      <product-id type="">3</product-id>
      <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
      <rules type="array">
        <rule type="TargetRule">
          <connector nil="true"></connector>
          <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
          <id type="integer">35</id>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>AREA</left-value>
          <lock-version type="integer">0</lock-version>
          <operator>Equals</operator>
          <position type="integer">1</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
```

```
          <right-value>Blog</right-value>
          <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
        </rule>
        <rule type="TargetRule">
          <connector>AND</connector>
          <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
          <id type="integer">36</id>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>DAYOFWEEK</left-value>
          <lock-version type="integer">0</lock-version>
          <operator>In</operator>
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
          <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
        </rule>
      </rules>
    </target>
  </targets>
  <dynamic-targets type="array"/>
</product>
```

Create a Product with Nested Items

Here is an example of creating a product with nested items:

```
POST: https://beta.aimatch.com/products.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<product>
  <name>TestProduct4</name>
  <description>Test Product 4</description>
  <list-rate type="decimal">3.21</list-rate>
  <rate-style>additional</rate-style>
  <rate-type>CPM</rate-type>
  <tier-id type="integer">3</tier-id>
  <targets-group-logic>OR</targets-group-logic>
  <targets-attributes type="array">
    <target>
      <description>prod 4 target 1 description</description>
      <name>prod_4_tar_1</name>
      <rules-attributes type="array">
        <rule>
          <connector nil="true"></connector>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>AREA</left-value>
          <operator>Equals</operator>
          <position type="integer">1</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>Blog</right-value>
        </rule>
        <rule>
          <connector>AND</connector>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>DAYOFWEEK</left-value>
          <operator>In</operator>
```

```
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
        </rule>
      </rules-attributes>
    </target>
    <target>
      <description>prod 4 target 2 description</description>
      <name>prod_4_tar_2</name>
      <rules-attributes type="array">
        <rule>
          <connector nil="true"></connector>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>AREA</left-value>
          <operator>Equals</operator>
          <position type="integer">1</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>Blog</right-value>
        </rule>
        <rule>
          <connector>AND</connector>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>DAYOFWEEK</left-value>
          <operator>In</operator>
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>SUNDAY,SATURDAY</right-value>
        </rule>
      </rules-attributes>
    </target>
  </targets-attributes>
  <dynamic-targets-attributes type="array">
    <dynamic-target>
      <left-value-id type="integer">16</left-value-id>
      <left-value-type>Tag</left-value-type>
    </dynamic-target>
    <dynamic-target>
      <left-value-id type="integer">17</left-value-id>
      <left-value-type>Tag</left-value-type>
    </dynamic-target>
  </dynamic-targets-attributes>
  <size-ids type="array">
    <id>10></id>
    <id>0></id>
  </size-ids>
</product>
```


Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<product>
  <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
  <description>Test Product 4</description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal" nil="true"></floor-rate>
  <goal-rate type="decimal" nil="true"></goal-rate>
```

```
<id type="integer">4</id>
<list-rate type="decimal">3.21</list-rate>
<lock-version type="integer">0</lock-version>
<name>TestProduct4</name>
<rate-style>additional</rate-style>
<rate-type>CPM</rate-type>
<targeting-type>None</targeting-type>
<targets-group-logic>OR</targets-group-logic>
<updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <name>House</name>
  <preemptible type="boolean">false</preemptible>
  <prioritization-method>Schedule</prioritization-method>
  <rank type="integer">3</rank>
  <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
</tier>
<sizes type="array">
  <size>
    <id type="integer">0</id>
    <name>(none specified)</name>
  </size>
  <size>
    <id type="integer">10</id>
    <name>160x230</name>
  </size>
</sizes>
<targets type="array">
  <target>
    <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
    <creator-id type="integer" nil="true"></creator-id>
    <description>prod 4 target 1 description</description>
    <id type="integer">5</id>
    <lock-version type="integer">0</lock-version>
    <name>prod_4_tar_1</name>
    <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
    <rules type="array">
      <rule type="TargetRule">
        <connector nil="true"></connector>
        <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
        <id type="integer">39</id>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <left-value>AREA</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>Equals</operator>
        <position type="integer">1</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>Blog</right-value>
        <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
      </rule>
```

```
        <rule type="TargetRule">
          <connector>AND</connector>
          <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
          <id type="integer">40</id>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>DAYOFWEEK</left-value>
          <lock-version type="integer">0</lock-version>
          <operator>In</operator>
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
          <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
        </rule>
      </rules>
    </target>
    <target>
      <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
      <creator-id type="integer" nil="true"></creator-id>
      <description>prod 4 target 2 description</description>
      <id type="integer">6</id>
      <lock-version type="integer">0</lock-version>
      <name>prod_4_tar_2</name>
      <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
      <rules type="array">
        <rule type="TargetRule">
          <connector nil="true"></connector>
          <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
          <id type="integer">41</id>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>AREA</left-value>
          <lock-version type="integer">0</lock-version>
          <operator>Equals</operator>
          <position type="integer">1</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>Blog</right-value>
          <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
        </rule>
        <rule type="TargetRule">
          <connector>AND</connector>
          <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
          <id type="integer">42</id>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>DAYOFWEEK</left-value>
          <lock-version type="integer">0</lock-version>
          <operator>In</operator>
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>SUNDAY,SATURDAY</right-value>
          <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
        </rule>
      </rules>
    </target>
  </targets>
  <dynamic-targets type="array"/>
</product>
```

Update a Product

Here is an example of updating a product:

```
PUT: https://beta.aimatch.com/products/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<product>
  <name>TestProduct4 Updated</name>
  <description>Test Product 4 updated</description>
  <floor-rate type="decimal">2.25</floor-rate>
  <goal-rate type="decimal">4.32</goal-rate>
  <list-rate type="decimal">4.39</list-rate>
  <rate-type>CPC</rate-type>
  <targets-group-logic>AND</targets-group-logic>
  <targets-attributes type="array">
    <target>
      <id type="integer">5</id>
      <description>prod 4 target 1 description updated</description>
      <name>prod_4_tar_1_updated</name>
    </target>
  </targets-attributes>
</product>
```


The response will not contain anything, but the status code will be 200.

The product now looks like this:

```
GET: https://beta.aimatch.com/products/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<product>
  <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
  <description>Test Product 4 updated</description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal">2.25</floor-rate>
  <goal-rate type="decimal">4.32</goal-rate>
  <id type="integer">4</id>
  <list-rate type="decimal">4.39</list-rate>
  <lock-version type="integer">1</lock-version>
  <name>TestProduct4 Updated</name>
  <rate-style>additional</rate-style>
  <rate-type>CPC</rate-type>
  <targeting-type>None</targeting-type>
  <targets-group-logic>AND</targets-group-logic>
  <updated-at type="datetime">2011-05-26T15:18:37-04:00</updated-at>
  <tier>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
    <flags type="integer">0</flags>
    <grace-period type="integer" nil="true"></grace-period>
    <grace-period-priority type="decimal" nil="true"></grace-period-priority>
    <id type="integer">3</id>
    <lock-version type="integer">0</lock-version>
    <name>House</name>
    <preemptible type="boolean">false</preemptible>
    <prioritization-method>Schedule</prioritization-method>
```

```
      <rank type="integer">3</rank>
      <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
    </tier>
    <sizes type="array">
      <size>
        <id type="integer">0</id>
        <name>(none specified)</name>
      </size>
      <size>
        <id type="integer">10</id>
        <name>160x230</name>
      </size>
    </sizes>
    <targets type="array">
      <target>
        <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
        <creator-id type="integer" nil="true"></creator-id>
        <description>prod 4 target 1 description updated</description>
        <id type="integer">5</id>
        <lock-version type="integer">1</lock-version>
        <name>prod_4_tar_1_updated</name>
        <product-id type="">4</product-id>
        <updated-at type="datetime">2011-05-26T15:18:37-04:00</updated-at>
        <rules type="array">
          <rule type="TargetRule">
            <connector nil="true"></connector>
            <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
            <id type="integer">39</id>
            <left-parentheses-count type="integer">0</left-parentheses-count>
            <left-value>AREA</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>Equals</operator>
            <position type="integer">1</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>Blog</right-value>
            <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
          </rule>
          <rule type="TargetRule">
            <connector>AND</connector>
            <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
            <id type="integer">40</id>
            <left-parentheses-count type="integer">1</left-parentheses-count>
            <left-value>DAYOFWEEK</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>In</operator>
            <position type="integer">2</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
            <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
          </rule>
        </rules>
      </target>
      <target>
        <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
        <creator-id type="integer" nil="true"></creator-id>
        <description>prod 4 target 2 description</description>
```

```
        <id type="integer">6</id>
        <lock-version type="integer">0</lock-version>
        <name>prod_4_tar_2</name>
        <product-id type="">4</product-id>
        <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
        <rules type="array">
          <rule type="TargetRule">
            <connector nil="true"></connector>
            <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
            <id type="integer">41</id>
            <left-parentheses-count type="integer">0</left-parentheses-count>
            <left-value>AREA</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>Equals</operator>
            <position type="integer">1</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>Blog</right-value>
            <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
          </rule>
          <rule type="TargetRule">
            <connector>AND</connector>
            <created-at type="datetime">2011-05-26T15:02:10-04:00</created-at>
            <id type="integer">42</id>
            <left-parentheses-count type="integer">1</left-parentheses-count>
            <left-value>DAYOFWEEK</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>In</operator>
            <position type="integer">2</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>SUNDAY,SATURDAY</right-value>
            <updated-at type="datetime">2011-05-26T15:02:10-04:00</updated-at>
          </rule>
        </rules>
      </target>
    </targets>
    <dynamic-targets type="array"/>
  </product>
```


Managing RFPs

Get a List of RFPs

Here is an example of getting a list of RFPs:

```
GET: https://beta.aimatch.com/rfps.xml
```

Here is the XML response:


```
<rfps type="array">
  <rfp>
    <id type="integer">1</id>
    <name>TestRFP1</name>
    <asset_url type="string">/system/rfps/1/TestRFP1.doc</asset_url>
  </rfp>
```

```
    <rfp>
      <id type="integer">2</id>
      <name>TestRFP2</name>
      <asset_url type="string" nil="true"></asset_url>
    </rfp>
    <rfp>
      <id type="integer">3</id>
      <name>TestRFP3</name>
      <asset_url type="string" nil="true"></asset_url>
    </rfp>
  </rfps>
```

This XML response provides 30 RFPs at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/rfps.xml?page=3
```

Get a Specific RFP

Here is an example of getting a specific RFP:

```
GET: https://beta.aimatch.com/rfps/3.xml
```

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<rfp>
  <asset-content-type nil="true"></asset-content-type>
  <asset-file-name nil="true"></asset-file-name>
  <asset-file-size type="integer" nil="true"></asset-file-size>
  <asset-updated-at type="datetime" nil="true"></asset-updated-at>
  <bill-to>advertiser</bill-to>
  <budget type="decimal" nil="true"></budget>
  <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
  <details-text>request for proposal 3 text updated</details-text>
  <details-type>text</details-type>
  <id type="integer">3</id>
  <lock-version type="integer">1</lock-version>
  <name>TestRFP3</name>
  <notes nil="true"></notes>
  <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
  <proposals type="array">
    <proposal>
      <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
      <discount type="decimal" nil="true"></discount>
      <end-date type="date" nil="true"></end-date>
      <expiration-date type="date">2012-03-18</expiration-date>
      <external-id nil="true"></external-id>
      <id type="integer">3</id>
      <lock-version type="integer">1</lock-version>
      <name>rfp_3_prop_1_updated</name>
      <notes nil="true"></notes>
      <po-number nil="true"></po-number>
      <probability type="integer" nil="true"></probability>
      <rfp-id type="integer">3</rfp-id>
      <start-date type="date" nil="true"></start-date>
```

```
<updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
<version type="integer">3</version>
<terms-and-conditions type="array">
  <terms-and-conditions>
    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
    <id type="integer">3</id>
    <lock-version type="integer">1</lock-version>
    <name>rfp3_prop_1_t_and_c_1_updated</name>
    <proposal-id type="">3</proposal-id>
    <terms-and-conditions-id type="">3</terms-and-conditions-id>
    <terms-and-conditions-text>T and C text updated</terms-and-conditions-text>
    <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
  </terms-and-conditions>
</terms-and-conditions>
<line-items type="array">
  <line-item>
    <actions type="integer" nil="true"></actions>
    <additional-fees type="decimal" nil="true"></additional-fees>
    <additional-fees-local type="decimal" nil="true"></additional-fees-local>
    <bonus-actions type="integer" nil="true"></bonus-actions>
    <bonus-clicks type="integer" nil="true"></bonus-clicks>
    <bonus-impressions type="integer" nil="true"></bonus-impressions>
    <clicks type="integer" nil="true"></clicks>
    <cost-actions type="decimal" nil="true"></cost-actions>
    <cost-actions-local type="decimal" nil="true"></cost-actions-local>
    <cost-clicks type="decimal" nil="true"></cost-clicks>
    <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
    <cost-impressions type="decimal" nil="true"></cost-impressions>
    <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
    <currency-id type="integer">840</currency-id>
    <conversion-rate type="float">1.0</conversion-rate>
    <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
    <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
    <end-date type="datetime" nil="true"></end-date>
    <flat-rate type="decimal" nil="true"></flat-rate>
    <flat-rate-local type="decimal" nil="true"></flat-rate-local>
    <flat-rate-flag type="boolean">false</flat-rate-flag>
    <id type="integer">3</id>
    <impressions type="integer">65</impressions>
    <lock-version type="integer">1</lock-version>
    <name>rfp3_prop_1_li_1_updated</name>
    <proposal-id type="integer">3</proposal-id>
    <size-id type="integer" nil="true"></size-id>
    <start-date type="datetime">2011-05-31T00:00:00-04:00</start-date>
    <status nil="true"></status>
    <targets-group-logic nil="true"></targets-group-logic>
    <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
    <version type="integer">2</version>
    <areas type="array"/>
    <tag-values type="array"/>
    <sites type="array">
      <site>
        <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
        <description nil="true"></description>
        <id type="integer">106</id>
```

```
      <line-item-id type="">3</line-item-id>
      <lock-version type="integer">0</lock-version>
      <name>rfp3_prop_1_li_1_site_1</name>
      <revenue-cut type="decimal" nil="true"></revenue-cut>
      <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
    </site>
    <site>
      <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
      <description nil="true"></description>
      <id type="integer">107</id>
      <line-item-id type="">3</line-item-id>
      <lock-version type="integer">0</lock-version>
      <name>rfp3_prop_1_li_1_site_2</name>
      <revenue-cut type="decimal" nil="true"></revenue-cut>
      <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
    </site>
  </sites>
  <targets type="array"/>
  <product>
    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
    <description></description>
    <discount-rate type="decimal" nil="true"></discount-rate>
    <floor-rate type="decimal">2.25</floor-rate>
    <goal-rate type="decimal">4.32</goal-rate>
    <id type="integer">3</id>
    <list-rate type="decimal">5.25</list-rate>
    <lock-version type="integer">3</lock-version>
    <name>TestProduct3</name>
    <rate-style>additional</rate-style>
    <rate-type>CPM</rate-type>
    <targeting-type>None</targeting-type>
    <targets-group-logic>OR</targets-group-logic>
    <updated-at type="datetime">2011-05-26T15:14:39-04:00</updated-at>
    <tier>
      <active type="boolean">true</active>
      <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
      <flags type="integer">0</flags>
      <grace-period type="integer" nil="true"></grace-period>
      <grace-period-priority type="decimal" nil="true"></grace-period-priority>
      <id type="integer">3</id>
      <lock-version type="integer">0</lock-version>
      <name>House</name>
      <preemptible type="boolean">false</preemptible>
      <prioritization-method>Schedule</prioritization-method>
      <rank type="integer">3</rank>
      <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
    </tier>
    <targets type="array">
      <target>
        <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
        <creator-id type="integer" nil="true"></creator-id>
        <description nil="true"></description>
        <id type="integer">3</id>
        <lock-version type="integer">0</lock-version>
        <name>rfp_3_prop_1_prod_1_tar_1</name>
        <product-id type="">3</product-id>
```

```
                <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
                <rules type="array">
                  <rule type="TargetRule">
                    <connector nil="true"></connector>
                    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
                    <id type="integer">35</id>
                    <left-parentheses-count type="integer">0</left-parentheses-count>
                    <left-value>AREA</left-value>
                    <lock-version type="integer">0</lock-version>
                    <operator>Equals</operator>
                    <position type="integer">1</position>
                    <right-parentheses-count type="integer">0</right-parentheses-count>
                    <right-value>Blog</right-value>
                    <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
                  </rule>
                  <rule type="TargetRule">
                    <connector>AND</connector>
                    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
                    <id type="integer">36</id>
                    <left-parentheses-count type="integer">1</left-parentheses-count>
                    <left-value>DAYOFWEEK</left-value>
                    <lock-version type="integer">0</lock-version>
                    <operator>In</operator>
                    <position type="integer">2</position>
                    <right-parentheses-count type="integer">0</right-parentheses-count>
                    <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
                    <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
                  </rule>
                </rules>
              </target>
            </targets>
            <dynamic-targets type="array"/>
            <sizes type="array">
              <size>
                <id type="integer">0</id>
                <name>(none specified)</name>
              </size>
            </sizes>
          </product>
          <category-list type="array"/>
        </line-item>
      </line-items>
    </proposal>
  </proposals>
  <advertiser>
    <id type="integer">2</id>
    <name>TestAdvertiser3</name>
  </advertiser>
  <contact>
    <full-name>Test Advertiser User 3</full-name>
    <id type="integer">2</id>
  </contact>
</rfp>
```

Create an RFP with Only the Minimum Required Fields

Here is an example of creating an RFP with only the minimum required fields:

```
POST: https://beta.aimatch.com/rfps.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rfp>
  <name>TestRFP4</name>
  <details-type>text</details-type>
  <details-text>request for proposal 4 text</details-text>
  <advertiser-id>2</advertiser-id>
  <contact-id type="integer">2</contact-id>
</rfp>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<rfp>
  <asset-content-type nil="true"></asset-content-type>
  <asset-file-name nil="true"></asset-file-name>
  <asset-file-size type="integer" nil="true"></asset-file-size>
  <asset-updated-at type="datetime" nil="true"></asset-updated-at>
  <bill-to>advertiser</bill-to>
  <budget type="decimal" nil="true"></budget>
  <created-at type="datetime">2011-05-26T17:02:30-04:00</created-at>
  <details-text>request for proposal 4 text</details-text>
  <details-type>text</details-type>
  <id type="integer">4</id>
  <lock-version type="integer">0</lock-version>
  <name>TestRFP4</name>
  <notes nil="true"></notes>
  <updated-at type="datetime">2011-05-26T17:02:30-04:00</updated-at>
  <proposals type="array"/>
  <advertiser>
    <id type="integer">2</id>
    <name>TestAdvertiser3</name>
  </advertiser>
  <contact>
    <full-name>Test Advertiser User 3</full-name>
    <id type="integer">2</id>
  </contact>
</rfp>
```

Creating an RFP with Nested Items

Here is an example of creating an RFP with nested items:

```
POST: https://beta.aimatch.com/rfps.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rfp>
  <name>TestRFP5</name>
  <details-type>text</details-type>
  <details-text>request for proposal 5 text</details-text>
  <advertiser-id>2</advertiser-id>
  <contact-id type="integer">2</contact-id>
```

```
<proposals-attributes type="array">
  <proposal>
    <name>rfp5_prop1</name>
    <expiration-date type="date">2012-03-17</expiration-date>
    <line-items-attributes type="array">
      <line-item>
        <name>rfp5_prop1_li1</name>
        <impressions type="integer" nil="true">20</impressions>
        <sites-attributes type="array">
          <site>
            <name>rfp5_prop1_li1_site1</name>
          </site>
          <site>
            <name>rfp5_prop1_li1_site2</name>
          </site>
        </sites-attributes>
        <start-date type="datetime">2011-05-31T00:00:00-04:00</start-date>
        <product-attributes>
          <name>rfp5_prop1_prod1</name>
          <list-rate type="decimal">1.25</list-rate>
          <rate-style>additional</rate-style>
          <rate-type>CPM</rate-type>
          <tier-id type="integer">3</tier-id>
          <targets-group-logic>OR</targets-group-logic>
          <targets-attributes type="array">
            <target>
              <description>rfp 5 proposal 1 product 1 target 1 description
</description>
              <name>rfp5_prop1_prod1_tar1</name>
              <rules-attributes type="array">
                <rule>
                  <connector nil="true"></connector>
                  <left-parentheses-count type="integer">0</left-parentheses-count>
                  <left-value>AREA</left-value>
                  <operator>Equals</operator>
                  <position type="integer">1</position>
                  <right-parentheses-count type="integer">0</right-parentheses-count>
                  <right-value>Blog</right-value>
                </rule>
                <rule>
                  <connector>AND</connector>
                  <left-parentheses-count type="integer">1</left-parentheses-count>
                  <left-value>DAYOFWEEK</left-value>
                  <operator>In</operator>
                  <position type="integer">2</position>
                  <right-parentheses-count type="integer">0</right-parentheses-count>
                  <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
                </rule>
              </rules-attributes>
            </target>
          </targets-attributes>
          <size-ids type="array">
            <id>0</id>
          </size-ids>
        </product-attributes>
      </line-item>
```

```
      </line-items-attributes>
      <terms-and-conditions-attributes type="array">
        <terms-and-conditions>
          <name>rfp5_prop1_t_and_c1</name>
          <terms-and-conditions-text>rfp5 proposal 1 T and C 1 text<
/terms-and-conditions-text>
        <terms-and-conditions>
      </terms-and-conditions-attributes>
    </proposal>
  </proposals-attributes>
</rfp>
```

Response (HTTP status: 201 Created):


```
<?xml version="1.0" encoding="UTF-8"?>
<rfp>
  <asset-content-type nil="true"></asset-content-type>
  <asset-file-name nil="true"></asset-file-name>
  <asset-file-size type="integer" nil="true"></asset-file-size>
  <asset-updated-at type="datetime" nil="true"></asset-updated-at>
  <bill-to>advertiser</bill-to>
  <budget type="decimal" nil="true"></budget>
  <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
  <details-text>request for proposal 5 text</details-text>
  <details-type>text</details-type>
  <id type="integer">5</id>
  <lock-version type="integer">0</lock-version>
  <name>TestRFP5</name>
  <notes nil="true"></notes>
  <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
  <proposals type="array">
    <proposal>
      <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
      <discount type="decimal" nil="true"></discount>
      <end-date type="date" nil="true"></end-date>
      <expiration-date type="date">2012-03-17</expiration-date>
      <external-id nil="true"></external-id>
      <id type="integer">4</id>
      <lock-version type="integer">0</lock-version>
      <name>rfp5_prop1</name>
      <notes nil="true"></notes>
      <po-number nil="true"></po-number>
      <probability type="integer" nil="true"></probability>
      <rfp-id type="integer">5</rfp-id>
      <start-date type="date" nil="true"></start-date>
      <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
      <version type="integer">2</version>
      <terms-and-conditions type="array">
        <terms-and-conditions>
          <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
          <id type="integer">5</id>
          <lock-version type="integer">0</lock-version>
          <name>rfp5_prop1_t_and_c1</name>
          <terms-and-conditions-text>rfp5 proposal 1 T and C 1 text<
/terms-and-conditions-text>
          <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
        </terms-and-conditions>
```

```
</terms-and-conditions>
<line-items type="array">
  <line-item>
    <actions type="integer" nil="true"></actions>
    <additional-fees type="decimal" nil="true"></additional-fees>
    <additional-fees-local type="decimal" nil="true"></additional-fees-local>
    <bonus-actions type="integer" nil="true"></bonus-actions>
    <bonus-clicks type="integer" nil="true"></bonus-clicks>
    <bonus-impressions type="integer" nil="true"></bonus-impressions>
    <clicks type="integer" nil="true"></clicks>
    <cost-actions type="decimal" nil="true"></cost-actions>
    <cost-actions-local type="decimal" nil="true"></cost-actions-local>
    <cost-clicks type="decimal" nil="true"></cost-clicks>
    <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
    <cost-impressions type="decimal" nil="true"></cost-impressions>
    <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
    <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
    <currency-id type="integer">840</currency-id>
    <conversion-rate type="float">1.0</conversion-rate>
    <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
    <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
    <end-date type="datetime" nil="true"></end-date>
    <flat-rate type="decimal" nil="true"></flat-rate>
    <flat-rate-local type="decimal" nil="true"></flat-rate-local>
    <flat-rate-flag type="boolean">false</flat-rate-flag>
    <id type="integer">4</id>
    <impressions type="integer">20</impressions>
    <lock-version type="integer">0</lock-version>
    <name>rfp5_prop1_li1</name>
    <proposal-id type="integer">4</proposal-id>
    <size-id type="integer" nil="true"></size-id>
    <start-date type="datetime">2011-05-31T00:00:00-04:00</start-date>
    <status nil="true"></status>
    <targets-group-logic nil="true"></targets-group-logic>
    <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
    <version type="integer">1</version>
    <areas type="array"/>
    <tag-values type="array"/>
    <sites type="array">
      <site>
        <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
        <description nil="true"></description>
        <id type="integer">108</id>
        <line-item-id type="">4</line-item-id>
        <lock-version type="integer">0</lock-version>
        <name>rfp5_prop1_li1_site1</name>
        <revenue-cut type="decimal" nil="true"></revenue-cut>
        <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
      </site>
      <site>
        <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
        <description nil="true"></description>
        <id type="integer">109</id>
        <line-item-id type="">4</line-item-id>
        <lock-version type="integer">0</lock-version>
        <name>rfp5_prop1_li1_site2</name>
```

```
          <revenue-cut type="decimal" nil="true"></revenue-cut>
          <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
        </site>
      </sites>
      <targets type="array"/>
      <product>
        <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
        <description nil="true"></description>
        <discount-rate type="decimal" nil="true"></discount-rate>
        <floor-rate type="decimal" nil="true"></floor-rate>
        <goal-rate type="decimal" nil="true"></goal-rate>
        <id type="integer">5</id>
        <list-rate type="decimal">1.25</list-rate>
        <lock-version type="integer">0</lock-version>
        <name>rfp5_prop1_prod1</name>
        <rate-style>additional</rate-style>
        <rate-type>CPM</rate-type>
        <targeting-type>None</targeting-type>
        <targets-group-logic>OR</targets-group-logic>
        <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
        <tier>
          <active type="boolean">true</active>
          <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
          <flags type="integer">0</flags>
          <grace-period type="integer" nil="true"></grace-period>
          <grace-period-priority type="decimal" nil="true"><
/grace-period-priority>
          <id type="integer">3</id>
          <lock-version type="integer">0</lock-version>
          <name>House</name>
          <preemptible type="boolean">false</preemptible>
          <prioritization-method>Schedule</prioritization-method>
          <rank type="integer">3</rank>
          <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
        </tier>
        <targets type="array">
          <target>
            <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
            <creator-id type="integer" nil="true"></creator-id>
            <description>rfp 5 proposal 1 product 1 target 1 description
</description>
            <id type="integer">7</id>
            <lock-version type="integer">0</lock-version>
            <name>rfp5_prop1_prod1_tar1</name>
            <product-id type="">5</product-id>
            <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
            <rules type="array">
              <rule type="TargetRule">
                <connector nil="true"></connector>
                <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
                <id type="integer">43</id>
                <left-parentheses-count type="integer">0</left-parentheses-count>
                <left-value>AREA</left-value>
                <lock-version type="integer">0</lock-version>
                <operator>Equals</operator>
                <position type="integer">1</position>
```

```
                <right-parentheses-count type="integer">0</right-parentheses-count>
                <right-value>Blog</right-value>
                <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
              </rule>
              <rule type="TargetRule">
                <connector>AND</connector>
                <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
                <id type="integer">44</id>
                <left-parentheses-count type="integer">1</left-parentheses-count>
                <left-value>DAYOFWEEK</left-value>
                <lock-version type="integer">0</lock-version>
                <operator>In</operator>
                <position type="integer">2</position>
                <right-parentheses-count type="integer">0</right-parentheses-count>
                <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
                <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
              </rule>
            </rules>
          </target>
        </targets>
        <dynamic-targets type="array"/>
        <sizes type="array">
          <size>
            <id type="integer">0</id>
            <name>(none specified)</name>
          </size>
        </sizes>
      </product>
      <category-list type="array"/>
    </line-item>
  </line-items>
</proposal>
</proposals>
<advertiser>
  <id type="integer">2</id>
  <name>TestAdvertiser3</name>
</advertiser>
<contact>
  <full-name>Test Advertiser User 3</full-name>
  <id type="integer">2</id>
</contact>
</rfp>
```

Update an RFP

Here is an example of updating an RFP:

```
PUT: https://beta.aimatch.com/rfps/5.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<rfp>
  <id type="integer">5</id>
  <name>TestRFP5Updated</name>
  <details-type>text</details-type>
  <details-text>request for proposal 5 text updated</details-text>
  <proposals-attributes type="array">
```

```
    <proposal>
      <id type="integer">4</id>
      <lock-version type="integer">0</lock-version>
      <name>rfp5_prop1_updated</name>
      <expiration-date type="date">2013-03-18</expiration-date>
      <line-items-attributes type="array">
        <line-item>
          <id type="integer">4</id>
          <lock-version type="integer">0</lock-version>
          <name>rfp5_prop1_li1_updated</name>
          <impressions type="integer" nil="true">30</impressions>
          <start-date type="datetime">2011-06-01T00:00:00-04:00</start-date>
          <product-attributes>
            <id type="integer">5</id>
            <name>rfp5_prop1_prod1_updated</name>
            <list-rate type="decimal">2.36</list-rate>
            <rate-type>CPC</rate-type>
          </product-attributes>
        </line-item>
      </line-items-attributes>
      <terms-and-conditions-attributes type="array">
        <terms-and-conditions>
          <id type="integer">5</id>
          <name>rfp5_prop1_t_and_c1_updated</name>
          <terms-and-conditions-text>rfp5 proposal 1 T and C 1 text updated
</terms-and-conditions-text>
        <terms-and-conditions>
      </terms-and-conditions-attributes>
    </proposal>
  </proposals-attributes>
</rfp>
```

The response will not contain anything, but the status code will be 200.

The RFP now looks like this:

```
GET: https://beta.aimatch.com/rfps/5.xml


<?xml version="1.0" encoding="UTF-8"?>
<rfp>
  <asset-content-type nil="true"></asset-content-type>
  <asset-file-name nil="true"></asset-file-name>
  <asset-file-size type="integer" nil="true"></asset-file-size>
  <asset-updated-at type="datetime" nil="true"></asset-updated-at>
  <bill-to>advertiser</bill-to>
  <budget type="decimal" nil="true"></budget>
  <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
  <details-text>request for proposal 5 text updated</details-text>
  <details-type>text</details-type>
  <id type="integer">5</id>
  <lock-version type="integer">1</lock-version>
  <name>TestRFP5Updated</name>
  <notes nil="true"></notes>
  <updated-at type="datetime">2011-05-26T17:21:22-04:00</updated-at>
  <proposals type="array">
    <proposal>
      <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
```

```
      <discount type="decimal" nil="true"></discount>
      <end-date type="date" nil="true"></end-date>
      <expiration-date type="date">2013-03-18</expiration-date>
      <external-id nil="true"></external-id>
      <id type="integer">4</id>
      <lock-version type="integer">1</lock-version>
      <name>rfp5_prop1_updated</name>
      <notes nil="true"></notes>
      <po-number nil="true"></po-number>
      <probability type="integer" nil="true"></probability>
      <rfp-id type="integer">5</rfp-id>
      <start-date type="date" nil="true"></start-date>
      <updated-at type="datetime">2011-05-26T17:21:22-04:00</updated-at>
      <version type="integer">3</version>
      <terms-and-conditions type="array">
        <terms-and-conditions>
          <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
          <id type="integer">5</id>
          <lock-version type="integer">1</lock-version>
          <name>rfp5_prop1_t_and_c1_updated</name>
          <proposal-id type="">4</proposal-id>
          <terms-and-conditions-id type="">5</terms-and-conditions-id>
          <terms-and-conditions-text>rfp5 proposal 1 T and C 1 text updated<
/terms-and-conditions-text>
          <updated-at type="datetime">2011-05-26T17:21:22-04:00</updated-at>
        </terms-and-conditions>
      </terms-and-conditions>
      <line-items type="array">
        <line-item>
          <actions type="integer" nil="true"></actions>
          <additional-fees type="decimal" nil="true"></additional-fees>
          <additional-fees-local type="decimal" nil="true"></additional-fees-local>
          <bonus-actions type="integer" nil="true"></bonus-actions>
          <bonus-clicks type="integer" nil="true"></bonus-clicks>
          <bonus-impressions type="integer" nil="true"></bonus-impressions>
          <clicks type="integer" nil="true"></clicks>
          <cost-actions type="decimal" nil="true"></cost-actions>
          <cost-actions-local type="decimal" nil="true"></cost-actions-local>
          <cost-clicks type="decimal" nil="true"></cost-clicks>
          <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
          <cost-impressions type="decimal" nil="true"></cost-impressions>
          <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
          <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
          <currency-id type="integer">840</currency-id>
          <conversion-rate type="float">1.0</conversion-rate>
          <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
          <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
          <end-date type="datetime" nil="true"></end-date>
          <flat-rate type="decimal" nil="true"></flat-rate>
          <flat-rate-local type="decimal" nil="true"></flat-rate-local>
          <flat-rate-flag type="boolean">false</flat-rate-flag>
          <id type="integer">4</id>
          <impressions type="integer">30</impressions>
          <lock-version type="integer">1</lock-version>
          <name>rfp5_prop1_li1_updated</name>
          <proposal-id type="integer">4</proposal-id>
```

```
<size-id type="integer" nil="true"></size-id>
<start-date type="datetime">2011-06-01T00:00:00-04:00</start-date>
<status nil="true"></status>
<targets-group-logic nil="true"></targets-group-logic>
<updated-at type="datetime">2011-05-26T17:21:22-04:00</updated-at>
<version type="integer">2</version>
<areas type="array"/>
<tag-values type="array"/>
<sites type="array">
  <site>
    <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">108</id>
    <line-item-id type="">4</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>rfp5_prop1_li1_site1</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
  </site>
  <site>
    <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">109</id>
    <line-item-id type="">4</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>rfp5_prop1_li1_site2</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
  </site>
</sites>
<targets type="array"/>
<product>
  <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
  <description nil="true"></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal" nil="true"></floor-rate>
  <goal-rate type="decimal" nil="true"></goal-rate>
  <id type="integer">5</id>
  <list-rate type="decimal">2.36</list-rate>
  <lock-version type="integer">1</lock-version>
  <name>rfp5_prop1_prod1_updated</name>
  <rate-style>additional</rate-style>
  <rate-type>CPC</rate-type>
  <targeting-type>None</targeting-type>
  <targets-group-logic>OR</targets-group-logic>
  <updated-at type="datetime">2011-05-26T17:21:22-04:00</updated-at>
  <tier>
    <active type="boolean">true</active>
    <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
    <flags type="integer">0</flags>
    <grace-period type="integer" nil="true"></grace-period>
    <grace-period-priority type="decimal" nil="true"><
/grace-period-priority>
    <id type="integer">3</id>
    <lock-version type="integer">0</lock-version>
    <name>House</name>
```

```
              <preemptible type="boolean">false</preemptible>
              <prioritization-method>Schedule</prioritization-method>
              <rank type="integer">3</rank>
              <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
            </tier>
            <targets type="array">
              <target>
                <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
                <creator-id type="integer" nil="true"></creator-id>
                <description>rfp 5 proposal 1 product 1 target 1 description<
/description>
                <id type="integer">7</id>
                <lock-version type="integer">0</lock-version>
                <name>rfp5_prop1_prod1_tar1</name>
                <product-id type="">5</product-id>
                <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
                <rules type="array">
                  <rule type="TargetRule">
                    <connector nil="true"></connector>
                    <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
                    <id type="integer">43</id>
                    <left-parentheses-count type="integer">0</left-parentheses-count>
                    <left-value>AREA</left-value>
                    <lock-version type="integer">0</lock-version>
                    <operator>Equals</operator>
                    <position type="integer">1</position>
                    <right-parentheses-count type="integer">0</right-parentheses-count>
                    <right-value>Blog</right-value>
                    <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
                  </rule>
                  <rule type="TargetRule">
                    <connector>AND</connector>
                    <created-at type="datetime">2011-05-26T17:11:38-04:00</created-at>
                    <id type="integer">44</id>
                    <left-parentheses-count type="integer">1</left-parentheses-count>
                    <left-value>DAYOFWEEK</left-value>
                    <lock-version type="integer">0</lock-version>
                    <operator>In</operator>
                    <position type="integer">2</position>
                    <right-parentheses-count type="integer">0</right-parentheses-count>
                    <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
                    <updated-at type="datetime">2011-05-26T17:11:38-04:00</updated-at>
                  </rule>
                </rules>
              </target>
            </targets>
            <dynamic-targets type="array"/>
            <sizes type="array">
              <size>
                <id type="integer">0</id>
                <name>(none specified)</name>
              </size>
            </sizes>
          </product>
          <category-list type="array"/>
        </line-item>
```

```
      </line-items>
    </proposal>
  </proposals>
  <advertiser>
    <id type="integer">2</id>
    <name>TestAdvertiser3</name>
  </advertiser>
  <contact>
    <full-name>Test Advertiser User 3</full-name>
    <id type="integer">2</id>
  </contact>
</rfp>
```

Managing Proposals

Get a List of Proposals

Here is an example of getting a list of proposals:

```
GET: https://beta.aimatch.com/proposals.xml
```

Here is the XML response:

```
<proposals type="array">
  <proposal>
    <id type="integer">1</id>
    <name>TestProposal1</name>
  </proposal>
  <proposal>
    <id type="integer">2</id>
    <name>TestProposal2</name>
  </proposal>
  <proposal>
    <id type="integer">3</id>
    <name>TestProposal3</name>
  </proposal>
</proposals>
```

This XML response provides 30 proposals at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/proposals.xml?page=3
```

To view details other than just the ID and name, add an include_details query parameter to the end of the URL.

Here is an example of a URL with an include_details query parameter:

```
https://beta.aimatch.com/proposals.xml?include_details=true
```

Get a Specific Proposal

Here is an example of getting a specific proposal:

```
GET: https://beta.aimatch.com/proposals/3.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<proposal>
  <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
  <discount type="decimal" nil="true"></discount>
  <end-date type="date" nil="true"></end-date>
  <expiration-date type="date">2012-03-18</expiration-date>
  <external-id nil="true"></external-id>
  <id type="integer">3</id>
  <lock-version type="integer">1</lock-version>
  <name>TestProposal3</name>
  <notes nil="true"></notes>
  <po-number nil="true"></po-number>
  <probability type="integer" nil="true"></probability>
  <start-date type="date" nil="true"></start-date>
  <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
  <version type="integer">3</version>
  <agency-contact>
    <id type="integer">16</id>
    <phone/>
    <address1/>
    <address2/>
    <country>United States</country>
    <updated-at type="datetime">2012-01-16T13:42:55-05:00</updated-at>
    <contactable-id type="integer">5</contactable-id>
    <website/>
    <city/>
    <full-name>frank1</full-name>
    <created-at type="datetime">2012-01-16T13:42:55-05:00</created-at>
    <email/>
    <state/>
    <contactable-type>Agency</contactable-type>
    <lock-version type="integer">0</lock-version>
    <title/>
    <zip/>
  </agency-contact>
  <advertiser-contact>
    <id type="integer">20</id>
    <phone/>
    <address1/>
    <address2/>
    <country>Australia</country>
    <updated-at type="datetime">2012-08-31T14:40:55-04:00</updated-at>
    <contactable-id type="integer">3</contactable-id>
    <website/>
    <city/>
    <full-name>FD</full-name>
    <created-at type="datetime">2012-08-31T14:40:55-04:00</created-at>
    <email/>
    <state/>
    <contactable-type>Advertiser</contactable-type>
    <lock-version type="integer">7</lock-version>
    <title/>
    <zip/>
  </advertiser-contact>
```

```
<rfp>
  <bill-to>advertiser</bill-to>
  <budget type="decimal" nil="true"></budget>
  <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
  <details-text>request for proposal 3 text</details-text>
  <details-type>text</details-type>
  <id type="integer">3</id>
  <lock-version type="integer">1</lock-version>
  <name>rfp_3_updated</name>
  <notes nil="true"></notes>
  <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
  <contact>
    <full-name>Test Advertiser User</full-name>
    <id type="integer">2</id>
  </contact>
  <advertiser>
    <id type="integer">2</id>
    <name>TestAdvertiser</name>
  </advertiser>
  <asset_url type="string" nil="true"></asset_url>
</rfp>
<line-items type="array">
  <line-item>
    <actions type="integer" nil="true"></actions>
    <additional-fees type="decimal" nil="true"></additional-fees>
    <additional-fees-local type="decimal" nil="true"></additional-fees-local>
    <bonus-actions type="integer" nil="true"></bonus-actions>
    <bonus-clicks type="integer" nil="true"></bonus-clicks>
    <bonus-impressions type="integer" nil="true"></bonus-impressions>
    <clicks type="integer" nil="true"></clicks>
    <cost-actions type="decimal" nil="true"></cost-actions>
    <cost-actions-local type="decimal" nil="true"></cost-actions-local>
    <cost-clicks type="decimal" nil="true"></cost-clicks>
    <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
    <cost-impressions type="decimal" nil="true"></cost-impressions>
    <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
    <currency-id type="integer">840</currency-id>
    <conversion-rate type="float">1.0</conversion-rate>
    <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
    <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
    <end-date type="datetime" nil="true"></end-date>
    <flat-rate type="decimal" nil="true"></flat-rate>
    <flat-rate-local type="decimal" nil="true"></flat-rate-local>
    <flat-rate-flag type="boolean">false</flat-rate-flag>
    <id type="integer">3</id>
    <impressions type="integer">65</impressions>
    <lock-version type="integer">1</lock-version>
    <name>rfp3_prop_1_li_1_updated</name>
    <price-check-status>OK</price-check-status>
    <product-line nil="true"></product-line>
    <proposal-id type="integer">3</proposal-id>
    <size-id type="integer" nil="true"></size-id>
    <start-date type="datetime">2011-05-31T00:00:00-04:00</start-date>
    <status nil="true"></status>
    <targets-group-logic nil="true"></targets-group-logic>
```

```
<updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
<version type="integer">2</version>
<targets type="array"/>
<areas type="array"/>
<tag-values type="array"/>
<sites type="array">
  <site>
    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">106</id>
    <line-item-id type="">3</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>rfp3_prop_1_li_1_site_1</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
  </site>
  <site>
    <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">107</id>
    <line-item-id type="">3</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>rfp3_prop_1_li_1_site_2</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
  </site>
</sites>
<billable-decision>As is</billable-decision>
<product>
  <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
  <description></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal">2.25</floor-rate>
  <goal-rate type="decimal">4.32</goal-rate>
  <id type="integer">3</id>
  <list-rate type="decimal">5.25</list-rate>
  <lock-version type="integer">3</lock-version>
  <name>TestProduct3</name>
  <rate-style>additional</rate-style>
  <rate-type>CPM</rate-type>
  <targeting-type>None</targeting-type>
  <targets-group-logic>OR</targets-group-logic>
  <updated-at type="datetime">2011-05-26T15:14:39-04:00</updated-at>
  <targets type="array">
    <target>
      <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
      <creator-id type="integer" nil="true"></creator-id>
      <description nil="true"></description>
      <id type="integer">3</id>
      <lock-version type="integer">0</lock-version>
      <name>rfp_3_prop_1_prod_1_tar_1</name>
      <product-id type="">3</product-id>
      <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
      <rules type="array">
        <rule type="TargetRule">
          <connector nil="true"></connector>
```

```
            <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
            <id type="integer">35</id>
            <left-parentheses-count type="integer">0</left-parentheses-count>
            <left-value>AREA</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>Equals</operator>
            <position type="integer">1</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>Blog</right-value>
            <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
          </rule>
          <rule type="TargetRule">
            <connector>AND</connector>
            <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
            <id type="integer">36</id>
            <left-parentheses-count type="integer">1</left-parentheses-count>
            <left-value>DAYOFWEEK</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>In</operator>
            <position type="integer">2</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
            <updated-at type="datetime">2011-05-24T13:36:55-04:00</updated-at>
          </rule>
        </rules>
      </target>
    </targets>
    <dynamic-targets type="array"/>
    <tier>
      <active type="boolean">true</active>
      <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
      <flags type="integer">0</flags>
      <grace-period type="integer" nil="true"></grace-period>
      <grace-period-priority type="decimal" nil="true"></grace-period-priority>
      <id type="integer">3</id>
      <lock-version type="integer">0</lock-version>
      <name>House</name>
      <preemptible type="boolean">false</preemptible>
      <prioritization-method>Schedule</prioritization-method>
      <rank type="integer">3</rank>
      <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
    </tier>
    <sizes type="array">
      <size>
        <id type="integer">5</id>
        <width type="integer">120</width>
        <updated-at type="datetime">2011-03-17T15:40:16-04:00</updated-at>
        <created-at type="datetime">2010-06-07T14:31:17-04:00</created-at>
        <name>120X240</name>
        <lock-version type="integer">5</lock-version>
        <height type="integer">240</height>
      </size>
    </sizes>
  </product>
  <category-list type="array"/>
</line-item>
```

```
  </line-items>
  <terms-and-conditions type="array">
    <terms-and-conditions>
      <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
      <id type="integer">3</id>
      <lock-version type="integer">1</lock-version>
      <name>rfp3_prop_1_t_and_c_1_updated</name>
      <proposal-id type="">3</proposal-id>
      <terms-and-conditions-text>T and C text updated</terms-and-conditions-text>
      <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
    </terms-and-conditions>
  </terms-and-conditions>
</proposal>
```


Create a Proposal with Nested Resources That Include Line Items

Here is an example of creating a proposal with nested resources that include line items:

```
POST: https://beta.aimatch.com/proposals.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<proposal>
  <name>TestProposal1</name>
  <expiration-date>2012-12-01</expiration-date>
  <start-date>2012-11-25</start-date>
  <end-date>2013-01-07</end-date>
  <external-id>999</external-id>
  <advertiser-id>1</advertiser-id>
  <agency-id>1</agency-id>
  <trafficker-id>3</trafficker-id>
  <terms-and-conditions-ids type="array">
    <id>1</id>
    <id>2</id>
  </terms-and-conditions-ids>
  <line-items-attributes type="array">
    <line-item>
      <name>test_prop_1_li_1</name>
      <start-date>2012-11-25</start-date>
      <end-date>2012-12-31</end-date>
      <cost-impressions-local type="decimal">0.05</cost-impressions-local>
      <product-attributes>
        <name>test_prop_1_prod_1</name>
        <list-rate type="decimal">1.25</list-rate>
        <rate-style>additional</rate-style>
        <rate-type>CPM</rate-type>
        <tier-id type="integer">1</tier-id>
        <targets-group-logic>OR</targets-group-logic>
        <targets-attributes type="array">
          <target>
            <description></description>
            <name>test_prop_1_prod_1_tar_1</name>
            <rules-attributes type="array">
              <rule>
                <connector nil="true"></connector>
                <left-parentheses-count type="integer">0</left-parentheses-count>
                <left-value>AREA</left-value>
```

```
          <operator>Equals</operator>
          <position type="integer">1</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>Blog</right-value>
        </rule>
        <rule>
          <connector>AND</connector>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>DAYOFWEEK</left-value>
          <operator>In</operator>
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
        </rule>
      </rules-attributes>
    </target>
  </targets-attributes>
  <size-ids type="array">
    <id>2</id>
    <id>0</id>
  </size-ids>
</product-attributes>
<sites-attributes type="array">
  <site>
    <name>test_prop_1_li_1_site_1</name>
  </site>
  <site>
    <name>test_prop_1_li_1_site_2</name>
  </site>
</sites-attributes>
<size-id>7</size-id>
<tag-value-ids type="array">
  <id type="integer">1</id>
  <id type="integer">2</id>
  <id type="integer">3</id>
</tag-value-ids>
<impressions>100000</impressions>
<ad-hoc-target-attributes>
  <rules-attributes type="array">
    <rule>
      <connector nil="true"></connector>
      <left-parentheses-count type="integer">0</left-parentheses-count>
      <left-value>GEO_DISTANCE_MILES</left-value>
      <operator>Greater or Equal To</operator>
      <position type="integer">1</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>2.56,51.521562,-0.261612</right-value>
    </rule>
    <rule>
      <connector nil="true">AND</connector>
      <left-parentheses-count type="integer">1</left-parentheses-count>
      <left-value>GEO_DISTANCE_MILES</left-value>
      <operator>Greater or Equal To</operator>
      <position type="integer">2</position>
      <right-parentheses-count type="integer">0</right-parentheses-count>
      <right-value>2.56,51.521562,-0.261612</right-value>
```

```
          </rule>
          <rule>
            <connector nil="true">OR</connector>
            <left-parentheses-count type="integer">0</left-parentheses-count>
            <left-value>GEO_DISTANCE_MILES</left-value>
            <operator>Greater or Equal To</operator>
            <position type="integer">3</position>
            <right-parentheses-count type="integer">1</right-parentheses-count>
            <right-value>2.56,51.521562,-0.261612</right-value>
          </rule>
        </rules-attributes>
      </ad-hoc-target-attributes>
    </line-item>
  </line-items-attributes>
</proposal>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<proposal>
  <bill-to>advertiser</bill-to>
  <converted-at type="date" nil="true"></converted-at>
  <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
  <credit-check-status>OK</credit-check-status>
  <discount type="decimal" nil="true"></discount>
  <end-date type="date">2013-01-07</end-date>
  <expiration-date type="date">2012-12-01</expiration-date>
  <external-id>999</external-id>
  <id type="integer">1</id>
  <internal-notes nil="true"></internal-notes>
  <lock-version type="integer">1</lock-version>
  <name>TestProposal1</name>
  <notes nil="true"></notes>
  <po-number>000001</po-number>
  <probability type="integer" nil="true"></probability>
  <start-date type="date">2012-11-25</start-date>
  <status nil="true"></status>
  <trafficker-id type="integer">3</trafficker-id>
  <updated-at type="datetime">2012-07-03T11:10:33-04:00</updated-at>
  <version type="integer">2</version>
  <terms-and-conditions type="array">
    <terms-and-conditions>
      <created-at type="datetime">2012-07-03T11:04:01-04:00</created-at>
      <id type="integer">1</id>
      <lock-version type="integer">1</lock-version>
      <name>test_terms_and_conditions_1</name>
      <terms-and-conditions-text>test_terms_and_conditions_1 text<
/terms-and-conditions-text>
      <updated-at type="datetime">2012-07-03T11:04:58-04:00</updated-at>
    </terms-and-conditions>
  </terms-and-conditions>
  <line-items type="array">
    <line-item>
      <actions type="integer" nil="true"></actions>
      <additional-fees type="decimal" nil="true"></additional-fees>
      <additional-fees-local type="decimal" nil="true"></additional-fees-local>
      <bonus-actions type="integer" nil="true"></bonus-actions>
```

```
<bonus-clicks type="integer" nil="true"></bonus-clicks>
<bonus-impressions type="integer" nil="true"></bonus-impressions>
<clicks type="integer" nil="true"></clicks>
<cost-actions type="decimal" nil="true"></cost-actions>
<cost-actions-local type="decimal" nil="true"></cost-actions-local>
<cost-clicks type="decimal" nil="true"></cost-clicks>
<cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
<cost-impressions type="decimal">0.05</cost-impressions>
<cost-impressions-local type="decimal">0.05</cost-impressions-local>
<created-at type="datetime">2012-07-03T11:10:33-04:00</created-at>
<daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
<daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
<end-date type="datetime">2012-12-31T00:00:00-05:00</end-date>
<flat-rate type="decimal" nil="true"></flat-rate>
<flat-rate-local type="decimal" nil="true"></flat-rate-local>
<flat-rate-flag type="boolean">false</flat-rate-flag>
<id type="integer">1</id>
<impressions type="integer">100000</impressions>
<inventory-check-status>OK</inventory-check-status>
<lock-version type="integer">0</lock-version>
<name>test_prop_1_li_1</name>
<notes nil="true"></notes>
<price-check-status>OK</price-check-status>
<product-line nil="true"></product-line>
<proposal-id type="integer">1</proposal-id>
<size-id type="integer">7</size-id>
<start-date type="datetime">2012-11-25T00:00:00-05:00</start-date>
<status nil="true"></status>
<targets-group-logic nil="true"></targets-group-logic>
<updated-at type="datetime">2012-07-03T11:10:33-04:00</updated-at>
<billable-decision>As is</billable-decision>
<sites type="array">
  <site>
    <created-at type="datetime">2012-07-03T11:10:33-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">1</id>
    <line-item-id type="">1</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_1_li_1_site_1</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2012-07-03T11:10:33-04:00</updated-at>
  </site>
  <site>
    <created-at type="datetime">2012-07-03T11:10:33-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">2</id>
    <line-item-id type="">1</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_1_li_1_site_2</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2012-07-03T11:10:33-04:00</updated-at>
  </site>
</sites>
<targets type="array"/>
<tag-values type="array">
  <tag-value>
```

```
    <created-at type="datetime">2012-07-02T17:28:06-04:00</created-at>
    <id type="integer">1</id>
    <line-item-id type="">1</line-item-id>
    <tag-id type="integer">1</tag-id>
    <updated-at type="datetime">2012-07-02T17:28:06-04:00</updated-at>
    <value>testtagval1</value>
  </tag-value>
  <tag-value>
    <created-at type="datetime">2012-07-02T17:28:06-04:00</created-at>
    <id type="integer">2</id>
    <line-item-id type="">1</line-item-id>
    <tag-id type="integer">1</tag-id>
    <updated-at type="datetime">2012-07-02T17:28:06-04:00</updated-at>
    <value>testtagval2</value>
  </tag-value>
  <tag-value>
    <created-at type="datetime">2012-07-02T17:28:06-04:00</created-at>
    <id type="integer">3</id>
    <line-item-id type="">1</line-item-id>
    <tag-id type="integer">1</tag-id>
    <updated-at type="datetime">2012-07-02T17:28:06-04:00</updated-at>
    <value>testtagval3</value>
  </tag-value>
</tag-values>
<product>
  <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
  <description nil="true"></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal" nil="true"></floor-rate>
  <goal-rate type="decimal" nil="true"></goal-rate>
  <id type="integer">1</id>
  <list-rate type="decimal">1.25</list-rate>
  <lock-version type="integer">0</lock-version>
  <name>test_prop_1_prod_1</name>
  <rate-style>additional</rate-style>
  <rate-type>CPM</rate-type>
  <targeting-type>None</targeting-type>
  <targets-group-logic>OR</targets-group-logic>
  <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
  <targets type="array">
    <target>
      <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
      <creator-id type="integer" nil="true"></creator-id>
      <description nil="true"></description>
      <id type="integer">1</id>
      <lock-version type="integer">0</lock-version>
      <name>test_prop_1_prod_1_tar_1</name>
      <product-id type="">1</product-id>
      <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
      <rules type="array">
        <rule type="TargetRule">
          <connector nil="true"></connector>
          <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
          <id type="integer">1</id>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>AREA</left-value>
```

```
        <lock-version type="integer">0</lock-version>
        <operator>Equals</operator>
        <position type="integer">1</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>Blog</right-value>
        <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
      </rule>
      <rule type="TargetRule">
        <connector>AND</connector>
        <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
        <id type="integer">2</id>
        <left-parentheses-count type="integer">1</left-parentheses-count>
        <left-value>DAYOFWEEK</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>In</operator>
        <position type="integer">2</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
        <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
      </rule>
    </rules>
  </target>
</targets>
<dynamic-targets type="array"/>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2012-07-02T17:43:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">1</id>
  <lock-version type="integer">0</lock-version>
  <name>TestTier1</name>
  <preemptible type="boolean">false</preemptible>
  <prioritization-method>Schedule</prioritization-method>
  <rank type="integer">1</rank>
  <updated-at type="datetime">2012-07-02T17:43:53-04:00</updated-at>
</tier>
<sizes type="array">
  <size>
    <id type="integer">0</id>
    <width type="integer">0</width>
    <updated-at type="datetime">2010-06-07T14:31:17-04:00</updated-at>
    <created-at type="datetime">2010-06-07T14:31:17-04:00</created-at>
    <name>(none specified)</name>
    <lock-version type="integer">0</lock-version>
    <height type="integer">0</height>
  </size>
  <size>
    <id type="integer">2</id>
    <width type="integer">234</width>
    <updated-at type="datetime">2010-07-06T21:30:54-04:00</updated-at>
    <created-at type="datetime">2010-06-07T14:31:17-04:00</created-at>
    <name>234X60</name>
    <lock-version type="integer">2</lock-version>
    <height type="integer">60</height>
```

```
      </size>
    </sizes>
  </product>
  <billable-amount-monthly type="array"/>
  <areas type="array"/>
  <category-list type="array"/>
  <ad-hoc-target>
    <description nil="true"/>
    <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
    <id type="integer">366</id>
    <creator-id type="integer" nil="true"/>
    <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
    <name>aiMatch_ad_hoc_target_1355249089_822</name>
    <lock-version type="integer">0</lock-version>
    <rules type="array">
      <rule type="TargetRule">
        <left-value>GEO_DISTANCE_MILES</left-value>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
        <id type="integer">2883</id>
        <right-value>2.56,51.521562,-0.261612</right-value>
        <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
        <position type="integer">1</position>
        <connector nil="true"/>
        <operator>Greater or Equal To</operator>
        <target-id type="integer">366</target-id>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <lock-version type="integer">0</lock-version>
      </rule>
      <rule type="TargetRule">
        <left-value>GEO_DISTANCE_MILES</left-value>
        <left-parentheses-count type="integer">1</left-parentheses-count>
        <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
        <id type="integer">2884</id>
        <right-value>2.56,51.521562,-0.261612</right-value>
        <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
        <position type="integer">2</position>
        <connector>AND</connector>
        <operator>Greater or Equal To</operator>
        <target-id type="integer">366</target-id>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <lock-version type="integer">0</lock-version>
      </rule>
      <rule type="TargetRule">
        <left-value>GEO_DISTANCE_MILES</left-value>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
        <id type="integer">2885</id>
        <right-value>2.56,51.521562,-0.261612</right-value>
        <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
        <position type="integer">3</position>
        <connector>OR</connector>
        <operator>Greater or Equal To</operator>
        <target-id type="integer">366</target-id>
        <right-parentheses-count type="integer">1</right-parentheses-count>
        <lock-version type="integer">0</lock-version>
```

```
        </rule>
      </rules>
    </ad-hoc-target>
  </line-item>
</line-items>
<agency>
  <legal-entity-name nil="true"/>
  <external-id></external-id>
  <status nil="true"/>
  <id type="integer">1</id>
  <name>TestAgency1</name>
  <commission type="decimal" nil="true"/>
  <credit-limit type="decimal" nil="true"/>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status nil="true"/>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2012-07-02T17:17:54-04:00</updated-at>
  <lock-version type="integer">1</lock-version>
  <pre-pay type="boolean">false</pre-pay>
  <created-at type="datetime">2012-07-02T17:17:42-04:00</created-at>
  <standard-discount type="decimal" nil="true"/>
  <region-id type="integer" nil="true"/>
  <category-list type="array"/>
</agency>
<advertiser>
  <legal-entity-name></legal-entity-name>
  <status nil="true"/>
  <id type="integer">1</id>
  <name>TestAdvertiser1</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal" nil="true"></credit-limit>
  <account-number></account-number>
  <business-id1 nil="true"/>
  <business-id2 nil="true"/>
  <customer-status>Active</customer-status>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean" nil="true"/>
  <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
  <standard-discount type="decimal" nil="true"/>
  <region-id type="integer" nil="true"/>
  <category-list type="array"/>
</advertiser>
<rfp>
  <bill-to>advertiser</bill-to>
  <budget type="decimal" nil="true"></budget>
  <created-at type="datetime" nil="true"></created-at>
  <details-text nil="true"></details-text>
  <details-type nil="true"></details-type>
  <lock-version type="integer">0</lock-version>
  <name nil="true"></name>
  <notes nil="true"></notes>
```

```
    <updated-at type="datetime" nil="true"></updated-at>
    <asset_url type="string" nil="true"></asset_url>
  </rfp>
</proposal>
```

Update a Proposal

Use the following tags to change the name, the line item to flat rate, and extend the end date of the proposal and the line item to the end of February. Changing to flat rate and setting billable-decision to equal "Weighted" causes billable-amount-monthly entries to be generated. Choices for billable-decision are as follows:

- As is
- Weighted
- Even
- Back load
- Front load

The billable-decision must be set to one of these values (except As is) when changing dates, goals, or costs. If not, an error is returned. The advertiser, agency, and terms and conditions IDs must all be sent even if they are not being changed. If the advertiser and agency IDs are not present, then they are cleared because they are not required. If the terms and condition IDs are not sent, then an error is returned because that is a required field.

```
PUT: https://beta.aimatch.com/proposals/1.xml
```

```
?xml version="1.0" encoding="UTF-8"?>
<proposal>
  <name>TestProposal1Updated</name>
  <notes nil="true">changed: name, line item to flat rate, extend end date of proposal
and line item to the end of Feb</notes>
  <end-date type="date">2013-02-28</end-date>
  <!--advertiser-id included since it gets cleared when not present-->
  <advertiser-id>1</advertiser-id>
  <!--agency-id included since it gets cleared when not present-->
  <agency-id>1</agency-id>
  <!--terms-and-conditions-ids included since an error is returned when not present-->
  <terms-and-conditions-ids type="array">
    <id>1</id>
    <id>2</id>
  </terms-and-conditions-ids>
  <line-items-attributes type="array">
    <line-item>
      <id type="integer">1</id>
      <billable-decision>Weighted</billable-decision>
      <currency-id>840</currency-id>
      <flat-rate-local>12000.00</flat-rate-local>
      <flat-rate-flag type="boolean">true</flat-rate-flag>
      <end-date type="date">2013-02-28</end-date>
      <ad-hoc-target-attributes>
        <description nil="true"/>
        <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
        <id type="integer">366</id>
        <creator-id type="integer" nil="true"/>
```

```
          <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
          <name>aiMatch_ad_hoc_target_1355249089_822</name>
          <lock-version type="integer">0</lock-version>
          <rules-attributes type="array">
            <rule>
              <left-value>GEO_DISTANCE_MILES</left-value>
              <left-parentheses-count type="integer">0</left-parentheses-count>
              <created-at type="datetime">2012-12-11T13:04:49-05:00</created-at>
              <id type="integer">2883</id>
              <right-value>3.56,51.521562,-0.261612</right-value>
              <updated-at type="datetime">2012-12-11T13:04:49-05:00</updated-at>
              <position type="integer">1</position>
              <connector nil="true"/>
              <operator>Greater or Equal To</operator>
              <target-id type="integer">366</target-id>
              <right-parentheses-count type="integer">0</right-parentheses-count>
              <lock-version type="integer">0</lock-version>
            </rule>
            <rule>
              <left-value>GEO_DISTANCE_MILES</left-value>
              <left-parentheses-count type="integer">1</left-parentheses-count>
              <id type="integer">2884</id>
              <right-value>3.56,51.521562,-0.261612</right-value>
              <position type="integer">2</position>
              <connector>AND</connector>
              <operator>Greater or Equal To</operator>
              <target-id type="integer">366</target-id>
              <right-parentheses-count type="integer">0</right-parentheses-count>
            </rule>
            <rule>
              <left-value>GEO_DISTANCE_MILES</left-value>
              <left-parentheses-count type="integer">0</left-parentheses-count>
              <id type="integer">2885</id>
              <right-value>3.56,51.521562,-0.261612</right-value>
              <position type="integer">3</position>
              <connector>OR</connector>
              <operator>Greater or Equal To</operator>
              <target-id type="integer">366</target-id>
              <right-parentheses-count type="integer">1</right-parentheses-count>
            </rule>
          </rules-attributes>
        </ad-hoc-target-attributes>
      </line-item>
    </line-items-attributes>
</proposal>
```


The response will not contain anything, but the status code will be 200.

The proposal now looks like this:

```
GET: https://beta.aimatch.com/proposals/1.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<proposal>
  <bill-to>advertiser</bill-to>
  <converted-at type="date" nil="true"></converted-at>
  <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
  <credit-check-status>OK</credit-check-status>
```

```
  <discount type="decimal" nil="true"></discount>
  <end-date type="date">2013-02-28</end-date>
  <expiration-date type="date">2012-12-01</expiration-date>
  <external-id>999</external-id>
  <id type="integer">1</id>
  <internal-notes nil="true"></internal-notes>
  <lock-version type="integer">2</lock-version>
  <name>TestProposal1Updated</name>
  <notes>changed: name, line item to flat rate, extend end date of proposal
and line item to the end of Feb</notes>
  <po-number>000001</po-number>
  <probability type="integer" nil="true"></probability>
  <start-date type="date">2012-11-25</start-date>
  <status nil="true"></status>
  <trafficker-id type="integer">3</trafficker-id>
  <updated-at type="datetime">2012-07-03T11:54:29-04:00</updated-at>
  <version type="integer">4</version>
  <terms-and-conditions type="array">
    <terms-and-conditions>
      <created-at type="datetime">2012-07-03T11:04:01-04:00</created-at>
      <id type="integer">1</id>
      <lock-version type="integer">1</lock-version>
      <name>test_terms_and_conditions_1</name>
      <proposal-id type="">1</proposal-id>
      <terms-and-conditions-text>test_terms_and_conditions_1 text
</terms-and-conditions-text>
      <updated-at type="datetime">2012-07-03T11:04:58-04:00</updated-at>
    </terms-and-conditions>
  </terms-and-conditions>
  <line-items type="array">
    <line-item>
      <actions type="integer" nil="true"></actions>
      <additional-fees type="decimal" nil="true"></additional-fees>
      <additional-fees-local type="decimal" nil="true"></additional-fees-local>
      <bonus-actions type="integer" nil="true"></bonus-actions>
      <bonus-clicks type="integer" nil="true"></bonus-clicks>
      <bonus-impressions type="integer" nil="true"></bonus-impressions>
      <clicks type="integer" nil="true"></clicks>
      <cost-actions type="decimal" nil="true"></cost-actions>
      <cost-actions-local type="decimal" nil="true"></cost-actions-local>
      <cost-clicks type="decimal" nil="true"></cost-clicks>
      <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
      <cost-impressions type="decimal">0.05</cost-impressions>
      <cost-impressions-local type="decimal">0.05</cost-impressions-local>
      <created-at type="datetime">2012-07-03T11:10:33-04:00</created-at>
      <currency-id type="integer">840</currency-id>
      <conversion-rate type="float">1.0</conversion-rate>
      <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
      <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
      <end-date type="datetime">2013-02-28T00:00:00-05:00</end-date>
      <flat-rate type="decimal">12000.0</flat-rate>
      <flat-rate-local type="decimal">12000.0</flat-rate>
      <flat-rate-flag type="boolean">true</flat-rate-flag>
      <id type="integer">1</id>
      <impressions type="integer">100000</impressions>
      <inventory-check-status>OK</inventory-check-status>
```

```
<lock-version type="integer">1</lock-version>
<name>test_prop_1_li_1</name>
<notes nil="true"></notes>
<price-check-status>OK</price-check-status>
<product-line nil="true"></product-line>
<proposal-id type="integer">1</proposal-id>
<size-id type="integer">7</size-id>
<start-date type="datetime">2012-11-25T00:00:00-05:00</start-date>
<status nil="true"></status>
<targets-group-logic nil="true"></targets-group-logic>
<updated-at type="datetime">2012-07-03T11:54:29-04:00</updated-at>
<billable-decision>As is</billable-decision>
<sites type="array">
  <site>
    <created-at type="datetime">2012-07-03T11:10:33-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">1</id>
    <line-item-id type="">1</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_1_li_1_site_1</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2012-07-03T11:10:33-04:00</updated-at>
  </site>
  <site>
    <created-at type="datetime">2012-07-03T11:10:33-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">2</id>
    <line-item-id type="">1</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_1_li_1_site_2</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2012-07-03T11:10:33-04:00</updated-at>
  </site>
</sites>
<targets type="array"/>
<tag-values type="array">
  <tag-value>
    <created-at type="datetime">2012-07-02T17:28:06-04:00</created-at>
    <id type="integer">1</id>
    <line-item-id type="">1</line-item-id>
    <tag-id type="integer">1</tag-id>
    <updated-at type="datetime">2012-07-02T17:28:06-04:00</updated-at>
    <value>testtagval1</value>
  </tag-value>
  <tag-value>
    <created-at type="datetime">2012-07-02T17:28:06-04:00</created-at>
    <id type="integer">2</id>
    <line-item-id type="">1</line-item-id>
    <tag-id type="integer">1</tag-id>
    <updated-at type="datetime">2012-07-02T17:28:06-04:00</updated-at>
    <value>testtagval2</value>
  </tag-value>
  <tag-value>
    <created-at type="datetime">2012-07-02T17:28:06-04:00</created-at>
    <id type="integer">3</id>
    <line-item-id type="">1</line-item-id>
```

```
    <tag-id type="integer">1</tag-id>
    <updated-at type="datetime">2012-07-02T17:28:06-04:00</updated-at>
    <value>testtagval3</value>
  </tag-value>
</tag-values>
<product>
  <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
  <description nil="true"></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal" nil="true"></floor-rate>
  <goal-rate type="decimal" nil="true"></goal-rate>
  <id type="integer">1</id>
  <list-rate type="decimal">1.25</list-rate>
  <lock-version type="integer">0</lock-version>
  <name>test_prop_1_prod_1</name>
  <rate-style>additional</rate-style>
  <rate-type>CPM</rate-type>
  <targeting-type>None</targeting-type>
  <targets-group-logic>OR</targets-group-logic>
  <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
  <targets type="array">
    <target>
      <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
      <creator-id type="integer" nil="true"></creator-id>
      <description nil="true"></description>
      <id type="integer">1</id>
      <lock-version type="integer">0</lock-version>
      <name>test_prop_1_prod_1_tar_1</name>
      <product-id type="">1</product-id>
      <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
      <rules type="array">
        <rule type="TargetRule">
          <connector nil="true"></connector>
          <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
          <id type="integer">1</id>
          <left-parentheses-count type="integer">0</left-parentheses-count>
          <left-value>AREA</left-value>
          <lock-version type="integer">0</lock-version>
          <operator>Equals</operator>
          <position type="integer">1</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>Blog</right-value>
          <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
        </rule>
        <rule type="TargetRule">
          <connector>AND</connector>
          <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
          <id type="integer">2</id>
          <left-parentheses-count type="integer">1</left-parentheses-count>
          <left-value>DAYOFWEEK</left-value>
          <lock-version type="integer">0</lock-version>
          <operator>In</operator>
          <position type="integer">2</position>
          <right-parentheses-count type="integer">0</right-parentheses-count>
          <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
          <updated-at type="datetime">2012-07-03T11:10:32-04:00</updated-at>
```

```
        </rule>
      </rules>
    </target>
  </targets>
  <dynamic-targets type="array"/>
  <tier>
    <active type="boolean">true</active>
    <created-at type="datetime">2012-07-02T17:43:53-04:00</created-at>
    <flags type="integer">0</flags>
    <grace-period type="integer" nil="true"></grace-period>
    <grace-period-priority type="decimal" nil="true"></grace-period-priority>
    <id type="integer">1</id>
    <lock-version type="integer">0</lock-version>
    <name>TestTier1</name>
    <preemptible type="boolean">false</preemptible>
    <prioritization-method>Schedule</prioritization-method>
    <rank type="integer">1</rank>
    <updated-at type="datetime">2012-07-02T17:43:53-04:00</updated-at>
  </tier>
  <sizes type="array">
    <size>
      <id type="integer">0</id>
      <name>(none specified)</name>
    </size>
  </sizes>
</product>
<billable-amount-monthly type="array">
  <billable-amount-monthly>
    <amount type="decimal">750.0</amount>
    <amount-local type="decimal">750.0</amount-local>
    <created-at type="datetime">2012-07-03T11:54:29-04:00</created-at>
    <id type="integer">1</id>
    <line-item-id type="integer">1</line-item-id>
    <month>2012-11</month>
    <updated-at type="datetime">2012-07-03T11:54:29-04:00</updated-at>
  </billable-amount-monthly>
  <billable-amount-monthly>
    <amount type="decimal">3875.0</amount>
    <amount-local type="decimal">3875.0</amount-local>
    <created-at type="datetime">2012-07-03T11:54:29-04:00</created-at>
    <id type="integer">2</id>
    <line-item-id type="integer">1</line-item-id>
    <month>2012-12</month>
    <updated-at type="datetime">2012-07-03T11:54:29-04:00</updated-at>
  </billable-amount-monthly>
  <billable-amount-monthly>
    <amount type="decimal">3875.0</amount>
    <amount-local type="decimal">3875.0</amount-local>
    <created-at type="datetime">2012-07-03T11:54:29-04:00</created-at>
    <id type="integer">3</id>
    <line-item-id type="integer">1</line-item-id>
    <month>2013-01</month>
    <updated-at type="datetime">2012-07-03T11:54:29-04:00</updated-at>
  </billable-amount-monthly>
  <billable-amount-monthly>
    <amount type="decimal">3500.0</amount>
```

```
          <amount-local type="decimal">3500.0</amount-local>
          <created-at type="datetime">2012-07-03T11:54:29-04:00</created-at>
          <id type="integer">4</id>
          <line-item-id type="integer">1</line-item-id>
          <month>2013-02</month>
          <updated-at type="datetime">2012-07-03T11:54:29-04:00</updated-at>
        </billable-amount-monthly>
      </billable-amount-monthly>
      <areas type="array"/>
      <category-list type="array"/>
    </line-item>
  </line-items>
  <agency>
    <legal-entity-name nil="true"/>
    <external-id></external-id>
    <status nil="true"/>
    <id type="integer">1</id>
    <name>TestAgency1</name>
    <commission type="decimal" nil="true"/>
    <credit-limit type="decimal" nil="true"/>
    <account-number></account-number>
    <business-id1 nil="true"/>
    <business-id2 nil="true"/>
    <customer-status nil="true"/>
    <credit-approved type="boolean">false</credit-approved>
    <updated-at type="datetime">2012-07-02T17:17:54-04:00</updated-at>
    <lock-version type="integer">1</lock-version>
    <pre-pay type="boolean">false</pre-pay>
    <created-at type="datetime">2012-07-02T17:17:42-04:00</created-at>
    <standard-discount type="decimal" nil="true"/>
    <region-id type="integer" nil="true"/>
    <category-list type="array"/>
  </agency>
  <advertiser>
    <legal-entity-name></legal-entity-name>
    <status nil="true"/>
    <id type="integer">1</id>
    <name>TestAdvertiser1</name>
    <industry-id type="integer" nil="true"/>
    <credit-limit type="decimal" nil="true"></credit-limit>
    <account-number></account-number>
    <business-id1 nil="true"/>
    <business-id2 nil="true"/>
    <customer-status>Active</customer-status>
    <bill-direct type="boolean">true</bill-direct>
    <credit-approved type="boolean">false</credit-approved>
    <updated-at type="datetime">2012-07-02T16:28:49-04:00</updated-at>
    <lock-version type="integer">0</lock-version>
    <pre-pay type="boolean" nil="true"/>
    <created-at type="datetime">2012-07-02T16:28:49-04:00</created-at>
    <standard-discount type="decimal" nil="true"/>
    <region-id type="integer" nil="true"/>
    <category-list type="array"/>
  </advertiser>
</proposal>
```

Reverting a Proposal to a Previous Newer Version

Here is an example of reverting a proposal to previous newer version:

```
POST: https://beta.aimatch.com/proposals/4/revert/2.xml
```

This command creates a new version of the proposal based on the version chosen (2 in this example). The response will contain the proposal ID, name, and new version number. The status code will be 200.


```
<?xml version="1.0" encoding="UTF-8"?>
<proposal>
  <id type="integer">4</id>
  <name>TestProposal4</name>
  <version type="integer">4</version>
</proposal>
```


The proposal now looks like this:

```
GET: https://beta.aimatch.com/proposals/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<proposal>
  <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
  <discount type="decimal" nil="true"></discount>
  <end-date type="date" nil="true"></end-date>
  <expiration-date type="date">2012-03-17</expiration-date>
  <external-id nil="true"></external-id>
  <id type="integer">4</id>
  <lock-version type="integer">3</lock-version>
  <name>TestProposal4</name>
  <notes nil="true"></notes>
  <po-number nil="true"></po-number>
  <probability type="integer" nil="true"></probability>
  <start-date type="date" nil="true"></start-date>
  <updated-at type="datetime">2011-05-27T16:43:43-04:00</updated-at>
  <version type="integer">4</version>
  <line-items type="array">
    <line-item>
      <actions type="integer" nil="true"></actions>
      <additional-fees type="decimal" nil="true"></additional-fees>
      <additional-fees-local type="decimal" nil="true"></additional-fees-local>
      <bonus-actions type="integer" nil="true"></bonus-actions>
      <bonus-clicks type="integer" nil="true"></bonus-clicks>
      <bonus-impressions type="integer" nil="true"></bonus-impressions>
      <clicks type="integer" nil="true"></clicks>
      <cost-actions type="decimal" nil="true"></cost-actions>
      <cost-actions-local type="decimal" nil="true"></cost-actions-local>
      <cost-clicks type="decimal" nil="true"></cost-clicks>
      <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
      <cost-impressions type="decimal" nil="true"></cost-impressions>
      <cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
      <created-at type="datetime">2011-05-27T15:48:16-04:00</created-at>
      <currency-id type="integer">840</currency-id>
      <conversion-rate type="float">1.0</conversion-rate>
      <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
      <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
```

```
<end-date type="datetime" nil="true"></end-date>
<flat-rate type="decimal" nil="true"></flat-rate>
<flat-rate-local type="decimal" nil="true"></flat-rate-local>
<flat-rate-flag type="boolean">false</flat-rate-flag>
<id type="integer">4</id>
<impressions type="integer">20</impressions>
<lock-version type="integer">2</lock-version>
<name>test_prop_4_li_1</name>
<proposal-id type="integer">4</proposal-id>
<size-id type="integer" nil="true"></size-id>
<start-date type="datetime">2011-05-31T00:00:00-04:00</start-date>
<status nil="true"></status>
<targets-group-logic nil="true"></targets-group-logic>
<updated-at type="datetime">2011-05-27T16:43:43-04:00</updated-at>
<version type="integer">3</version>
<targets type="array"/>
<areas type="array"/>
<tag-values type="array"/>
<sites type="array">
  <site>
    <created-at type="datetime">2011-05-27T15:48:16-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">110</id>
    <line-item-id type="">4</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_4_li_1_site_1</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2011-05-27T15:48:16-04:00</updated-at>
  </site>
  <site>
    <created-at type="datetime">2011-05-27T15:48:16-04:00</created-at>
    <description nil="true"></description>
    <id type="integer">111</id>
    <line-item-id type="">4</line-item-id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_4_li_1_site_2</name>
    <revenue-cut type="decimal" nil="true"></revenue-cut>
    <updated-at type="datetime">2011-05-27T15:48:16-04:00</updated-at>
  </site>
</sites>
<product>
  <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
  <description nil="true"></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal">2.15</floor-rate>
  <goal-rate type="decimal">3.15</goal-rate>
  <id type="integer">6</id>
  <list-rate type="decimal">3.25</list-rate>
  <lock-version type="integer">1</lock-version>
  <name>test_prop_4_prod_1_updated</name>
  <rate-style>additional</rate-style>
  <rate-type>CPC</rate-type>
  <targeting-type>None</targeting-type>
  <targets-group-logic>OR</targets-group-logic>
  <updated-at type="datetime">2011-05-27T16:13:31-04:00</updated-at>
  <targets type="array">
```

```
  <target>
    <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
    <creator-id type="integer" nil="true"></creator-id>
    <description nil="true"></description>
    <id type="integer">8</id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_4_prod_1_tar_1</name>
    <product-id type="">6</product-id>
    <updated-at type="datetime">2011-05-27T15:48:15-04:00</updated-at>
    <rules type="array">
      <rule type="TargetRule">
        <connector nil="true"></connector>
        <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
        <id type="integer">45</id>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <left-value>AREA</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>Equals</operator>
        <position type="integer">1</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>Blog</right-value>
        <updated-at type="datetime">2011-05-27T15:48:15-04:00</updated-at>
      </rule>
      <rule type="TargetRule">
        <connector>AND</connector>
        <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
        <id type="integer">46</id>
        <left-parentheses-count type="integer">1</left-parentheses-count>
        <left-value>DAYOFWEEK</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>In</operator>
        <position type="integer">2</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
        <updated-at type="datetime">2011-05-27T15:48:15-04:00</updated-at>
      </rule>
    </rules>
  </target>
</targets>
<dynamic-targets type="array"/>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <name>House</name>
  <preemptible type="boolean">false</preemptible>
  <prioritization-method>Schedule</prioritization-method>
  <rank type="integer">3</rank>
  <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
</tier>
<sizes type="array">
  <size>
```

```
            <id type="integer">0</id>
            <name>(none specified)</name>
          </size>
        </sizes>
      </product>
      <category-list type="array"/>
    </line-item>
  </line-items>
  <terms-and-conditions type="array">
    <terms-and-conditions>
      <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
      <id type="integer">6</id>
      <lock-version type="integer">2</lock-version>
      <name>test_prop_4_t_and_c_1_updated_again</name>
      <proposal-id type="">4</proposal-id>
      <terms-and-conditions-text>T and C text updated again</terms-and-conditions-text>
      <updated-at type="datetime">2011-05-27T16:42:52-04:00</updated-at>
    </terms-and-conditions>
  </terms-and-conditions>
</proposal>
```


Convert a Proposal to a Campaign

Here is an example of converting a proposal to a campaign:

```
PUT: https://beta.aimatch.com/proposals/4/convert_to_campaign.xml
```

This command creates a new campaign based on the proposal. The response will contain the new campaign. The status code will be 200


```
<?xml version="1.0" encoding="UTF-8"?>
<campaign>
  <campaign-type>Guaranteed</campaign-type>
  <created-at type="datetime">2011-05-27T18:00:07-04:00</created-at>
  <description nil="true"></description>
  <end-date type="date" nil="true"></end-date>
  <external-id nil="true"></external-id>
  <flags type="integer">0</flags>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <name>TestProposal4</name>
  <notes nil="true"></notes>
  <proposal-id type="integer">4</proposal-id>
  <start-date type="date">2011-05-31</start-date>
  <status nil="true"></status>
  <updated-at type="datetime">2011-05-27T18:00:07-04:00</updated-at>
  <flights type="array">
    <flight>
      <actions type="integer" nil="true"></actions>
      <bonus-actions type="integer" nil="true"></bonus-actions>
      <bonus-clicks type="integer" nil="true"></bonus-clicks>
      <bonus-impressions type="integer" nil="true"></bonus-impressions>
      <campaign-id type="integer">3</campaign-id>
      <clicks type="integer" nil="true"></clicks>
      <companion-type nil="true"></companion-type>
      <cost-actions type="decimal" nil="true"></cost-actions>
      <cost-actions-local type="decimal" nil="true"></cost-actions-local>
```

```
<cost-clicks type="decimal" nil="true"></cost-clicks>
<cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
<cost-impressions type="decimal" nil="true"></cost-impressions>
<cost-impressions-local type="decimal" nil="true"></cost-impressions-local>
<created-at type="datetime">2011-05-27T18:00:08-04:00</created-at>
<creative-selection-method>Weighted</creative-selection-method>
<currency-id type="integer">840</currency-id>
<conversion-rate type="float">1.0</conversion-rate>
<daily-end-time type="datetime" nil="true"></daily-end-time>
<daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
<daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
<daily-start-time type="datetime" nil="true"></daily-start-time>
<description nil="true"></description>
<end-date type="datetime" nil="true"></end-date>
<flat-rate type="decimal" nil="true"></flat-rate>
<flat-rate-local type="decimal" nil="true"></flat-rate-local>
<flat-rate-flag type="boolean">false</flat-rate-flag>
<flight-type>Guaranteed</flight-type>
<id type="integer">4</id>
<impressions type="integer">20</impressions>
<lock-version type="integer">0</lock-version>
<name>test_prop_6_li_1</name>
<notes nil="true"></notes>
<reference-rate type="integer" nil="true"></reference-rate>
<size-id type="integer" nil="true"></size-id>
<start-date type="datetime">2011-05-31T00:00:00-04:00</start-date>
<status>missing creative</status>
<targets-group-logic nil="true"></targets-group-logic>
<updated-at type="datetime">2011-05-27T18:00:08-04:00</updated-at>
<user-info nil="true"></user-info>
<weight type="integer">1</weight>
<flight-creatives type="array"/>
<targets type="array"/>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">3</id>
  <lock-version type="integer">0</lock-version>
  <name>House</name>
  <preemptible type="boolean">false</preemptible>
  <prioritization-method>Schedule</prioritization-method>
  <rank type="integer">3</rank>
  <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
</tier>
<areas type="array"/>
<tag-values type="array"/>
<product>
  <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
  <description nil="true"></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal">2.15</floor-rate>
  <goal-rate type="decimal">3.15</goal-rate>
  <id type="integer">6</id>
```

```
<list-rate type="decimal">3.25</list-rate>
<lock-version type="integer">1</lock-version>
<name>test_prop_4_prod_1_updated</name>
<rate-style>additional</rate-style>
<rate-type>CPC</rate-type>
<targeting-type>None</targeting-type>
<targets-group-logic>OR</targets-group-logic>
<updated-at type="datetime">2011-05-27T16:13:31-04:00</updated-at>
<targets type="array">
  <target>
    <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
    <creator-id type="integer" nil="true"></creator-id>
    <description nil="true"></description>
    <id type="integer">8</id>
    <lock-version type="integer">0</lock-version>
    <name>test_prop_6_prod_1_tar_1</name>
    <product-id type="">6</product-id>
    <updated-at type="datetime">2011-05-27T15:48:15-04:00</updated-at>
    <rules type="array">
      <rule type="TargetRule">
        <connector nil="true"></connector>
        <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
        <id type="integer">45</id>
        <left-parentheses-count type="integer">0</left-parentheses-count>
        <left-value>AREA</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>Equals</operator>
        <position type="integer">1</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>Blog</right-value>
        <updated-at type="datetime">2011-05-27T15:48:15-04:00</updated-at>
      </rule>
      <rule type="TargetRule">
        <connector>AND</connector>
        <created-at type="datetime">2011-05-27T15:48:15-04:00</created-at>
        <id type="integer">46</id>
        <left-parentheses-count type="integer">1</left-parentheses-count>
        <left-value>DAYOFWEEK</left-value>
        <lock-version type="integer">0</lock-version>
        <operator>In</operator>
        <position type="integer">2</position>
        <right-parentheses-count type="integer">0</right-parentheses-count>
        <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
        <updated-at type="datetime">2011-05-27T15:48:15-04:00</updated-at>
      </rule>
    </rules>
  </target>
</targets>
<dynamic-targets type="array"/>
<tier>
  <active type="boolean">true</active>
  <created-at type="datetime">2011-04-25T09:55:53-04:00</created-at>
  <flags type="integer">0</flags>
  <grace-period type="integer" nil="true"></grace-period>
  <grace-period-priority type="decimal" nil="true"></grace-period-priority>
  <id type="integer">3</id>
```

```
        <lock-version type="integer">0</lock-version>
        <name>House</name>
        <preemptible type="boolean">false</preemptible>
        <prioritization-method>Schedule</prioritization-method>
        <rank type="integer">3</rank>
        <updated-at type="datetime">2011-04-25T09:55:53-04:00</updated-at>
      </tier>
      <sizes type="array">
        <size>
          <id type="integer">0</id>
          <name>(none specified)</name>
        </size>
      </sizes>
    </product>
    <sites type="array">
      <site>
        <created-at type="datetime">2011-05-27T15:48:16-04:00</created-at>
        <description nil="true"></description>
        <id type="integer">110</id>
        <lock-version type="integer">0</lock-version>
        <name>test_prop_4_li_1_site_1</name>
        <revenue-cut type="decimal" nil="true"></revenue-cut>
        <updated-at type="datetime">2011-05-27T15:48:16-04:00</updated-at>
      </site>
      <site>
        <created-at type="datetime">2011-05-27T15:48:16-04:00</created-at>
        <description nil="true"></description>
        <id type="integer">111</id>
        <lock-version type="integer">0</lock-version>
        <name>test_prop_4_li_1_site_2</name>
        <revenue-cut type="decimal" nil="true"></revenue-cut>
        <updated-at type="datetime">2011-05-27T15:48:16-04:00</updated-at>
      </site>
    </sites>
    <flight-action-policies type="array"/>
    <category-list type="array"/>
    <beacon_count type="integer">0</beacon_count>
    <top_priority type="integer">0</top_priority>
  </flight>
</flights>
<advertiser>
  <legal-entity-name></legal-entity-name>
  <status nil="true"/>
  <id type="integer">2</id>
  <name>TestAdvertiser</name>
  <industry-id type="integer" nil="true"/>
  <credit-limit type="decimal">50000.0</credit-limit>
  <account-number></account-number>
  <business-id1></business-id1>
  <business-id2></business-id2>
  <customer-status>Active</customer-status>
  <bill-direct type="boolean">true</bill-direct>
  <credit-approved type="boolean">false</credit-approved>
  <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
  <lock-version type="integer">0</lock-version>
  <pre-pay type="boolean" nil="true"></pre-pay>
```

```
    <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
    <standard-discount type="decimal" nil="true"/>
    <region-id type="integer" nil="true"/>
    <contacts type="array">
      <contact>
        <address1 nil="true"></address1>
        <address2 nil="true"></address2>
        <city nil="true"></city>
        <contactable-id type="integer">2</contactable-id>
        <contactable-type>Advertiser</contactable-type>
        <country nil="true"></country>
        <created-at type="datetime">2011-04-25T09:55:52-04:00</created-at>
        <email nil="true"></email>
        <full-name>Test Advertiser User</full-name>
        <id type="integer">2</id>
        <lock-version type="integer">0</lock-version>
        <phone nil="true"></phone>
        <state nil="true"></state>
        <title nil="true"></title>
        <updated-at type="datetime">2011-04-25T09:55:52-04:00</updated-at>
        <website nil="true"></website>
        <zip nil="true"></zip>
      </contact>
    </contacts>
  </advertiser>
</campaign>
```

Get a List of Line Items

Here is an example of getting a list of line items:

```
GET: https://beta.aimatch.com/line_items.xml
```

Here is the XML response:

```
<line-items type="array">
  <line-item>
    <id type="integer">1</id>
    <name>test_prop_4_li_1</name>
  </line-item>
  <line-item>
    <id type="integer">2</id>
    <name>test_prop_4_li_2</name>
  </line-item>
  <line-item>
    <id type="integer">3</id>
    <name>test_prop_4_li_3</name>
  </line-item>
</line-item>
```

This XML response provides 30 line items at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/line_items.xml?page=3
```

To view details other than just the ID and name, add an include_details query parameter to the end of the URL.

Here is an example of a URL with an include_details parameter:

```
https://beta.aimatch.com/line_items.xml?include_details=true
```

Create a New Line Item without Nesting It inside a Proposal

Here is an example of creating a new line item without nesting it inside a proposal:

```
POST: https://beta.aimatch.com/proposals/1/line_items.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<line-item>
  <name>Another Line Item For Proposal 1</name>
  <start-date>2012-11-25</start-date>
  <end-date>2012-12-31</end-date>
  <cost-impressions-local type="decimal">0.05</cost-impressions-local>
  <product-id>4</product-id>
</line-item>
```

These tags create a new line item for the proposal with ID 1.

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<line-item>
  <actions type="integer" nil="true"></actions>
  <additional-fees type="decimal" nil="true"></additional-fees>
  <additional-fees-local type="decimal" nil="true"></additional-fees-local>
  <bonus-actions type="integer" nil="true"></bonus-actions>
  <bonus-clicks type="integer" nil="true"></bonus-clicks>
  <bonus-impressions type="integer" nil="true"></bonus-impressions>
  <clicks type="integer" nil="true"></clicks>
  <cost-actions type="decimal" nil="true"></cost-actions>
  <cost-actions-local type="decimal" nil="true"></cost-actions-local>
  <cost-clicks type="decimal" nil="true"></cost-clicks>
  <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
  <cost-impressions type="decimal">0.05</cost-impressions>
  <cost-impressions-local type="decimal">0.05</cost-impressions-local>
  <created-at type="datetime">2012-08-31T11:42:35-04:00</created-at>
  <currency-id type="integer">840</currency-id>
  <conversion-rate type="float">1.0</conversion-rate>
  <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
  <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
  <end-date type="datetime">2012-12-31T00:00:00-05:00</end-date>
  <flat-rate type="decimal" nil="true"></flat-rate>
  <flat-rate-local type="decimal" nil="true"></flat-rate-local>
  <flat-rate-flag type="boolean">false</flat-rate-flag>
  <id type="integer">8</id>
  <impressions type="integer" nil="true"></impressions>
  <inventory-check-status>OK</inventory-check-status>
  <lock-version type="integer">0</lock-version>
  <name>Another Line Item For Proposal 1</name>
  <notes nil="true"></notes>
  <price-check-status>OK</price-check-status>
  <product-line nil="true"></product-line>
  <proposal-id type="integer">1</proposal-id>
  <size-id type="integer" nil="true"></size-id>
  <start-date type="datetime">2012-11-25T00:00:00-05:00</start-date>
```

```
<status>open</status>
<targets-group-logic nil="true"></targets-group-logic>
<updated-at type="datetime">2012-08-31T11:42:35-04:00</updated-at>
<billable-decision>Weighted</billable-decision>
<sites type="array"/>
<billable-amount-monthly type="array">
  <billable-amount-monthly>
    <amount type="decimal">0.0</amount>
    <amount-local type="decimal">0.0</amount-local>
    <created-at type="datetime">2012-08-31T11:42:35-04:00</created-at>
    <id type="integer">12</id>
    <line-item-id type="integer">8</line-item-id>
    <month>2012-11</month>
    <updated-at type="datetime">2012-08-31T11:42:35-04:00</updated-at>
  </billable-amount-monthly>
  <billable-amount-monthly>
    <amount type="decimal">0.0</amount>
    <amount-local type="decimal">0.0</amount-local>
    <created-at type="datetime">2012-08-31T11:42:35-04:00</created-at>
    <id type="integer">13</id>
    <line-item-id type="integer">8</line-item-id>
    <month>2012-12</month>
    <updated-at type="datetime">2012-08-31T11:42:35-04:00</updated-at>
  </billable-amount-monthly>
</billable-amount-monthly>
<targets type="array"/>
<product>
  <created-at type="datetime">2012-07-13T11:12:13-04:00</created-at>
  <description></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal" nil="true"></floor-rate>
  <goal-rate type="decimal" nil="true"></goal-rate>
  <id type="integer">4</id>
  <list-rate type="decimal">102.0</list-rate>
  <lock-version type="integer">0</lock-version>
  <name>prod_w_sites_and_areas</name>
  <rate-style>additional</rate-style>
  <rate-type>Flat Rate</rate-type>
  <targeting-type>Ad-hoc</targeting-type>
  <targets-group-logic>AND</targets-group-logic>
  <updated-at type="datetime">2012-07-13T11:12:13-04:00</updated-at>
  <targets type="array">
    <target>
      <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
      <creator-id type="integer" nil="true"></creator-id>
      <description>sites</description>
      <id type="integer">1</id>
      <lock-version type="integer">1</lock-version>
      <name>test_prop_1_prod_1_tar_1</name>
      <product-id type="">4</product-id>
      <updated-at type="datetime">2012-07-26T11:13:11-04:00</updated-at>
      <rules type="array">
        <rule type="TargetRule">
          <connector nil="true"></connector>
          <created-at type="datetime">2012-07-30T16:39:14-04:00</created-at>
          <id type="integer">15</id>
```

```
            <left-parentheses-count type="integer">0</left-parentheses-count>
            <left-value>AREA</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>Equals</operator>
            <position type="integer">1</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>Blog</right-value>
            <updated-at type="datetime">2012-07-30T16:39:14-04:00</updated-at>
          </rule>
          <rule type="TargetRule">
            <connector>OR</connector>
            <created-at type="datetime">2012-07-30T16:39:14-04:00</created-at>
            <id type="integer">16</id>
            <left-parentheses-count type="integer">0</left-parentheses-count>
            <left-value>DAYOFWEEK</left-value>
            <lock-version type="integer">0</lock-version>
            <operator>In</operator>
            <position type="integer">2</position>
            <right-parentheses-count type="integer">0</right-parentheses-count>
            <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
            <updated-at type="datetime">2012-07-30T16:39:14-04:00</updated-at>
          </rule>
        </rules>
      </target>
    </targets>
    <tier>
      <active type="boolean">true</active>
      <created-at type="datetime">2012-07-02T17:43:53-04:00</created-at>
      <flags type="integer">0</flags>
      <grace-period type="integer" nil="true"></grace-period>
      <grace-period-priority type="decimal" nil="true"></grace-period-priority>
      <id type="integer">1</id>
      <lock-version type="integer">0</lock-version>
      <name>TestTier1</name>
      <preemptible type="boolean">false</preemptible>
      <prioritization-method>Schedule</prioritization-method>
      <rank type="integer">1</rank>
      <updated-at type="datetime">2012-07-02T17:43:53-04:00</updated-at>
    </tier>
    <dynamic-targets type="array"/>
    <sizes type="array">
      <size>
        <id type="integer">0</id>
        <name>(none specified)</name>
      </size>
    </sizes>
  </product>
  <areas type="array"/>
  <tag-values type="array"/>
  <category-list type="array"/>
</line-item>
```

Update a Line Item without Nesting It inside a Proposal

Here is an example of updating a line time without nesting it inside a proposal:

```
PUT: https://beta.aimatch.com/proposals/1/line_items.xml
```

Note: This method changes the name and cost impressions.


```
?xml version="1.0" encoding="UTF-8"?>
<line-item>
  <name>Another Line Item For Proposal 1 Updated</name>
  <cost-impressions-local type="decimal">0.75</cost-impressions-local>
  <billable-decision>Weighted</billable-decision>
</line-item
```


The response will not contain anything, but the status code will be 200.

The line item now looks like this:

```
GET: https://beta.aimatch.com/proposals/1/line_items.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<line-item>
  <actions type="integer" nil="true"></actions>
  <additional-fees type="decimal" nil="true"></additional-fees>
  <additional-fees-local type="decimal" nil="true"></additional-fees-local>
  <bonus-actions type="integer" nil="true"></bonus-actions>
  <bonus-clicks type="integer" nil="true"></bonus-clicks>
  <bonus-impressions type="integer" nil="true"></bonus-impressions>
  <clicks type="integer" nil="true"></clicks>
  <cost-actions type="decimal" nil="true"></cost-actions>
  <cost-actions-local type="decimal" nil="true"></cost-actions-local>
  <cost-clicks type="decimal" nil="true"></cost-clicks>
  <cost-clicks-local type="decimal" nil="true"></cost-clicks-local>
  <cost-impressions type="decimal">0.75</cost-impressions>
  <cost-impressions-local type="decimal">0.75</cost-impressions-local>
  <created-at type="datetime">2012-08-31T11:42:35-04:00</created-at>
  <currency-id type="integer">840</currency-id>
  <conversion-rate type="float">1.0</conversion-rate>
  <daily-rev-cap type="decimal" nil="true"></daily-rev-cap>
  <daily-rev-cap-local type="decimal" nil="true"></daily-rev-cap-local>
  <end-date type="datetime">2012-12-31T00:00:00-05:00</end-date>
  <flat-rate type="decimal" nil="true"></flat-rate>
  <flat-rate-local type="decimal" nil="true"></flat-rate-local>
  <flat-rate-flag type="boolean">false</flat-rate-flag>
  <id type="integer">8</id>
  <impressions type="integer" nil="true"></impressions>
  <inventory-check-status>OK</inventory-check-status>
  <lock-version type="integer">2</lock-version>
  <name>Another Line Item For Proposal 1 Updated</name>
  <notes nil="true"></notes>
  <price-check-status>OK</price-check-status>
  <product-line nil="true"></product-line>
  <proposal-id type="integer">1</proposal-id>
  <size-id type="integer" nil="true"></size-id>
  <start-date type="datetime">2012-11-25T00:00:00-05:00</start-date>
  <status>open</status>
```

```
<targets-group-logic nil="true"></targets-group-logic>
<updated-at type="datetime">2012-08-31T13:00:54-04:00</updated-at>
<billable-decision>As is</billable-decision>
<sites type="array"/>
<billable-amount-monthly type="array">
  <billable-amount-monthly>
    <amount type="decimal">0.0</amount>
    <amount-local type="decimal">0.0</amount-local>
    <created-at type="datetime">2012-08-31T13:00:54-04:00</created-at>
    <id type="integer">14</id>
    <line-item-id type="integer">8</line-item-id>
    <month>2012-11</month>
    <updated-at type="datetime">2012-08-31T13:00:54-04:00</updated-at>
  </billable-amount-monthly>
  <billable-amount-monthly>
    <amount type="decimal">0.0</amount>
    <amount-local type="decimal">0.0</amount-local>
    <created-at type="datetime">2012-08-31T13:00:54-04:00</created-at>
    <id type="integer">15</id>
    <line-item-id type="integer">8</line-item-id>
    <month>2012-12</month>
    <updated-at type="datetime">2012-08-31T13:00:54-04:00</updated-at>
  </billable-amount-monthly>
</billable-amount-monthly>
<targets type="array"/>
<product>
  <created-at type="datetime">2012-07-13T11:12:13-04:00</created-at>
  <description></description>
  <discount-rate type="decimal" nil="true"></discount-rate>
  <floor-rate type="decimal" nil="true"></floor-rate>
  <goal-rate type="decimal" nil="true"></goal-rate>
  <id type="integer">4</id>
  <list-rate type="decimal">102.0</list-rate>
  <lock-version type="integer">0</lock-version>
  <name>prod_w_sites_and_areas</name>
  <rate-style>additional</rate-style>
  <rate-type>Flat Rate</rate-type>
  <targeting-type>Ad-hoc</targeting-type>
  <targets-group-logic>AND</targets-group-logic>
  <updated-at type="datetime">2012-07-13T11:12:13-04:00</updated-at>
  <targets type="array">
    <target>
      <created-at type="datetime">2012-07-03T11:10:32-04:00</created-at>
      <creator-id type="integer" nil="true"></creator-id>
      <description>sites</description>
      <id type="integer">1</id>
      <lock-version type="integer">1</lock-version>
      <name>test_prop_1_prod_1_tar_1</name>
      <product-id type="">4</product-id>
      <updated-at type="datetime">2012-07-26T11:13:11-04:00</updated-at>
      <rules type="array">
        <rule type="TargetRule">
          <connector nil="true"></connector>
          <created-at type="datetime">2012-07-30T16:39:14-04:00</created-at>
          <id type="integer">15</id>
          <left-parentheses-count type="integer">0</left-parentheses-count>
```

```
              <left-value>AREA</left-value>
              <lock-version type="integer">0</lock-version>
              <operator>Equals</operator>
              <position type="integer">1</position>
              <right-parentheses-count type="integer">0</right-parentheses-count>
              <right-value>Blog</right-value>
              <updated-at type="datetime">2012-07-30T16:39:14-04:00</updated-at>
            </rule>
            <rule type="TargetRule">
              <connector>OR</connector>
              <created-at type="datetime">2012-07-30T16:39:14-04:00</created-at>
              <id type="integer">16</id>
              <left-parentheses-count type="integer">0</left-parentheses-count>
              <left-value>DAYOFWEEK</left-value>
              <lock-version type="integer">0</lock-version>
              <operator>In</operator>
              <position type="integer">2</position>
              <right-parentheses-count type="integer">0</right-parentheses-count>
              <right-value>SUNDAY,WEDNESDAY,THURSDAY,SATURDAY</right-value>
              <updated-at type="datetime">2012-07-30T16:39:14-04:00</updated-at>
            </rule>
          </rules>
        </target>
      </targets>
      <tier>
        <active type="boolean">true</active>
        <created-at type="datetime">2012-07-02T17:43:53-04:00</created-at>
        <flags type="integer">0</flags>
        <grace-period type="integer" nil="true"></grace-period>
        <grace-period-priority type="decimal" nil="true"></grace-period-priority>
        <id type="integer">1</id>
        <lock-version type="integer">0</lock-version>
        <name>TestTier1</name>
        <preemptible type="boolean">false</preemptible>
        <prioritization-method>Schedule</prioritization-method>
        <rank type="integer">1</rank>
        <updated-at type="datetime">2012-07-02T17:43:53-04:00</updated-at>
      </tier>
      <dynamic-targets type="array"/>
      <sizes type="array">
        <size>
          <id type="integer">0</id>
          <name>(none specified)</name>
        </size>
      </sizes>
    </product>
    <areas type="array"/>
    <tag-values type="array"/>
    <category-list type="array"/>
  </line-item>
```

Managing Terms and Conditions

Get a List of Terms and Conditions

Here is an example of getting a list of terms and conditions:

```
GET: https://beta.aimatch.com/terms_and_conditions.xml
```

Here is the XML response:


```
<terms_and_conditions type="array">
  <terms_and_conditions>
    <id type="integer">1</id>
    <name>TestTermsAndConditions1</name>
  </terms_and_conditions>
  <terms_and_conditions>
    <id type="integer">2</id>
    <name>TestTermsAndConditions2</name>
  </terms_and_conditions>
  <terms_and_conditions>
    <id type="integer">3</id>
    <name>TestTermsAndConditions3</name>
  </terms_and_conditions>
</terms_and_conditions>
```


This XML response provides 30 terms_and_conditions at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/terms_and_conditions.xml?page=3
```

Get Specific Terms and Conditions

Here is an example of getting specific terms and conditions:

```
GET: https://beta.aimatch.com/terms_and_conditions/3.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<terms-and-conditions>
  <lock-version type="integer">0</lock-version>
  <id type="integer">3</id>
  <name>TestTermsAndConditions3</name>
  <created-at type="datetime">2011-05-24T13:36:55-04:00</created-at>
  <updated-at type="datetime">2011-05-24T15:14:30-04:00</updated-at>
  <terms-and-conditions-text>T and C text 3 text</terms-and-conditions-text>
  <asset-url></asset-url>
</terms-and-conditions>
```


Create Terms and Conditions

Here is an example of creating specific terms and conditions:

```
POST: https://beta.aimatch.com/terms_and_conditions.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<terms-and-conditions>
  <name>TestTermsAndConditions4</name>
  <terms-and-conditions-text>T and C 4 text</terms-and-conditions-text>
</terms-and-conditions>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<terms-and-conditions>
  <lock-version type="integer">0</lock-version>
  <id type="integer">4</id>
  <name>TestTermsAndConditions4</name>
  <created-at type="datetime">2011-05-24T15:00:20-04:00</created-at>
  <updated-at type="datetime">2011-05-24T15:00:20-04:00</updated-at>
  <terms-and-conditions-text>T and C 4 text</terms-and-conditions-text>
  <asset-url></asset-url>
</terms-and-conditions>
```

Create Terms and Conditions with a Base64–encoded PDF File

Here is an example of creating terms and conditions with a base64–encoded PDF file:

```
POST: https://beta.aimatch.com/terms_and_conditions.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<terms-and-conditions>
  <name>Terms and conditions 4</name>
  <asset type="file" content-type="application/pdf" name="terms4.pdf">
    <![CDATA[JVBERi0xLjMKJcTl8uXrp/Og0MTGCjQgMCBvYmoKPDwgL0xlbmd0aCA1IDAgUiAv
RmlsdGVyIC9GbGF0ZURlY29kZSA+PgpzdHJlYW0KeAFFjr0KwkAQhHufYkotvMzeJffTKoJtYMEHO
GIhiSHe+4MbLKx2v5mPZTeM2JA8YolOyhDQxwwfXOa+fCY88E=]]>
  </asset>
</terms-and-conditions>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<terms-and-conditions>
  <lock-version type="integer">0</lock-version>
  <name>Terms and Conditions 4</name>
  <id type="integer">12</id>
  <created-at type="datetime">2012-11-14T06:16:53-09:00</created-at>
  <updated-at type="datetime">2012-11-14T06:16:53-09:00</updated-at>
  <terms-and-conditions-text/>
  <asset-url>/system/terms_and_conditions/13/de26b77d9cfe080de2d28c1275d245f839dcd283/
terms4.pdf?1352906587</asset-url>
</terms-and-conditions>
```

Update Terms and Conditions

Here is an example of updating terms and conditions:

```
PUT: https://beta.aimatch.com/terms_and_conditions/4.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<terms-and-conditions>
  <id type="integer">4</id>
  <name>TestTermsAndConditions4Updated</name>
```

```
  <lock-version type="integer">0</lock-version>
  <terms-and-conditions-text>T and C 4 text updated</terms-and-conditions-text>
</terms-and-conditions>
```

The response will not contain anything, but the status code will be 200.

The terms-and-conditions now looks like this:

```
GET: https://beta.aimatch.com/terms_and_conditions/4.xml
```


```
<?xml version="1.0" encoding="UTF-8"?>
<terms-and-conditions>
  <lock-version type="integer">1</lock-version>
  <id type="integer">4</id>
  <name>TestTermsAndConditions4Updated</name>
  <created-at type="datetime">2011-05-24T15:00:20-04:00</created-at>
  <updated-at type="datetime">2011-05-27T18:42:50-04:00</updated-at>
  <terms-and-conditions-text>T and C 4 text updated</terms-and-conditions-text>
  <asset-url></asset-url>
</terms-and-conditions>
```


Managing Currencies

Currencies are referenced using their ISO 4217 numeric code. For convenience, the alphabetic code is included in currency record XML. The only attribute that can be modified is the conversion rate. Flights and line items support setting a flight- or line-item-specific currency by setting the currency_id attribute. When currency is set due to the creation of a flight or line item, or the currency of the flight or line item is changed, the conversion rate of the currency is copied into the conversion_rate attribute of the flight or line item.

Get a List of Currencies

Here is an example of getting a list of currencies:

```
GET: https://beta.aimatch.com/currencies.xml
```

Here is the XML response:


```
<?xml version="1.0"?>
<currencies type="array">
  <currency>
    <id>36</id>
    <code>AUD</code>
    <conversion_rate>1.0</conversion_rate>
  </currency>
  <currency>
    <id>124</id>
    <code>CAD</code>
    <conversion_rate>1.0</conversion_rate>
  </currency>
...
</currencies>
```

Get Specific Currencies

Here is an example of getting specific currencies:

```
GET: https://beta.aimatch.com/currencies/840.xml
```

Here is the XML response:


```
<?xml version="1.0"?>
<currency>
  <id>840</id>
  <code>USD</code>
  <conversion_rate>1.0</conversion_rate>
</currency>
```


Update Currencies

Here is an example of updating currencies:

```
PUT: https://beta.aimatch.com/currencies/978.xml


<?xml version="1.0"?>
<currency>
  <id>978</id>
  <conversion_rate>5.0</conversion_rate>
</currency>
```

The response will contain the XML representation of the updated currency, and the status code will be 200.

The currency now looks like this:


```
<?xml version="1.0"?>
<currency>
  <id>978</id>
  <code>EUR</code>
  <conversion_rate>5.0</conversion_rate>
</currency>
```


Managing Roles and Permissions

Currencies are referenced using their ISO 4217 numeric code. For convenience, the alphabetic code is included in currency record XML. The only attribute that can be modified is the conversion rate. Flights and line items support setting a flight- or line-item-specific currency by setting the currency_id attribute. When currency is set due to the creation of a flight or line item, or the currency of the flight or line item is changed, the conversion rate of the currency is copied into the conversion_rate attribute of the flight or line item.

Get a List of Roles

Here is an example of getting a list of roles:

```
GET: http://beta.aimatch.com/roles.xml
```

Here is the XML response:


```
<roles type="array">
  <role>
    <id type="integer">8</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>Administrator</name>
  </role>
  <role>
    <id type="integer">4</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>AdOpsManager</name>
  </role>
  <role>
    <id type="integer">5</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>Advertiser</name>
  </role>
  <role>
    <id type="integer">6</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>Agency</name>
  </role>
  <role>
    <id type="integer">7</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>FinanceUser</name>
  </role>
  <role>
    <id type="integer">9</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>Publisher</name>
  </role>
  <role>
    <id type="integer">14</id>
    <is-system-defined type="boolean">false</is-system-defined>
    <name>roles_with_flags</name>
  </role>
  <role>
    <id type="integer">15</id>
    <is-system-defined type="boolean">false</is-system-defined>
    <name>roles_with_flags Copy</name>
  </role>
  <role>
    <id type="integer">2</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>SalesManager</name>
  </role>
  <role>
    <id type="integer">1</id>
    <is-system-defined type="boolean">true</is-system-defined>
    <name>Salesperson</name>
  </role>
  <role>
    <id type="integer">3</id>
    <is-system-defined type="boolean">true</is-system-defined>
```

```
    <name>Trafficker</name>
  </role>
</roles>
```

This provides 30 roles at a time ordered by name. If you want to get the next set, you can add a page query parameter to the end of the URL:

```
http://beta.aimatch.com/roles.xml?page=2
```

Get a Specific Role

Here is an example of getting a specific role:

```
GET: http://beta.aimatch.com/roles/3.xml
```

Here is the XML response:


```
<?xml version="1.0" encoding="UTF-8"?>
<role>
  <id type="integer">3</id>
  <created-at type="datetime" nil="true"/>
  <is-system-defined type="boolean">true</is-system-defined>
  <lock-version type="integer">0</lock-version>
  <name>Trafficker</name>
  <updated-at type="datetime" nil="true"/>
  <roles_salesperson type="integer">0</roles_salesperson>
  <roles_sales_manager type="integer">0</roles_sales_manager>
  <roles_trafficker type="integer">0</roles_trafficker>
  <roles_ad_ops_manager type="integer">0</roles_ad_ops_manager>
  <permissions type="array">
    <permission type="Permissions::RolePermission">
      <id type="integer">844</id>
      <created-at type="datetime">2014-05-13T08:41:10+12:00</created-at>
      <lock-version type="integer">1</lock-version>
      <entity-name>Action</entity-name>
      <updated-at type="datetime">2014-05-13T08:41:10+12:00</updated-at>
      <mine_view_permission type="integer">0</mine_view_permission>
      <delete_permission type="integer">0</delete_permission>
      <mine_create_permission type="integer">0</mine_create_permission>
      <team_view_permission type="integer">0</team_view_permission>
      <mine_edit_permission type="integer">0</mine_edit_permission>
      <team_create_permission type="integer">0</team_create_permission>
      <mine_delete_permission type="integer">0</mine_delete_permission>
      <view_permission type="integer">1</view_permission>
      <team_edit_permission type="integer">0</team_edit_permission>
      <none_permission type="integer">0</none_permission>
      <create_permission type="integer">0</create_permission>
      <team_delete_permission type="integer">0</team_delete_permission>
      <edit_permission type="integer">0</edit_permission>
    </permission>
    <permission type="Permissions::RolePermission">
      <id type="integer">847</id>
      <created-at type="datetime">2014-05-13T08:41:10+12:00</created-at>
      <lock-version type="integer">1</lock-version>
      <entity-name>ActionReport</entity-name>
      <updated-at type="datetime">2014-05-13T08:41:10+12:00</updated-at>
      <mine_view_permission type="integer">0</mine_view_permission>
      <delete_permission type="integer">0</delete_permission>
```

```
      <mine_create_permission type="integer">0</mine_create_permission>
      <team_view_permission type="integer">0</team_view_permission>
      <mine_edit_permission type="integer">0</mine_edit_permission>
      <team_create_permission type="integer">0</team_create_permission>
      <mine_delete_permission type="integer">0</mine_delete_permission>
      <view_permission type="integer">1</view_permission>
      <team_edit_permission type="integer">0</team_edit_permission>
      <none_permission type="integer">0</none_permission>
      <create_permission type="integer">0</create_permission>
      <team_delete_permission type="integer">0</team_delete_permission>
      <edit_permission type="integer">1</edit_permission>
    </permission>
  </permissions>
</role>
```

Creating a Role

Here is an example of creating a role:

```
POST: http://beta.aimatch.com/roles.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<role>
 <name>Senior Trafficker I</name>
 <roles_salesperson type="integer">0</roles_salesperson>
 <roles_sales_manager type="integer">0</roles_sales_manager>
 <roles_trafficker type="integer">1</roles_trafficker>
 <roles_ad_ops_manager type="integer">1</roles_ad_ops_manager>
</role>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<role>
 <id type="integer">17</id>
 <created-at type="datetime">2014-06-25T07:42:22+12:00</created-at>
 <is-system-defined type="boolean">false</is-system-defined>
 <lock-version type="integer">0</lock-version>
 <name>Senior Trafficker I</name>
 <updated-at type="datetime">2014-06-25T07:42:22+12:00</updated-at>
 <permissions type="array"/>
 <roles_salesperson type="integer">0</roles_salesperson>
 <roles_sales_manager type="integer">0</roles_sales_manager>
 <roles_trafficker type="integer">1</roles_trafficker>
 <roles_ad_ops_manager type="integer">1</roles_ad_ops_manager>
</role>
```

Updating a Role and Permissions

Here is an example of updating a role:

```
PUT: http://beta.aimatch.com/roles/3.xml
```

This is a list of entities that you can set permissions: Alert, BackgroundTask, Tier, Campaign, Flight, Creative, CreativeFormat, MimeType, Action, Target, Tag, Site, Area, Size, SupertagValue, Advertiser, Agency, Publisher, Industry, Region, Product, Proposal, Rfp, LineItem, Uplift, TermsAndConditions, Avail, ConvertToCampaign, ConvertedProposal, RevertProposal, User, ApiKey, AuditLog, Team, AdvertisersReport, AreaSizeReport, BillingReport, BiReport, CamReport, DemographicReport, GeoReport, PublishersReport, ScheduleReport, SiteAreaReport, SizeServedReport, TagSummaryReport, TimeOfDayReport, ThirdPartyReport, ActionReport, ProductReport, UvReport, DeviceReport, WorkflowRevenueReport, PerformanceReport, Dashboards, DashboardFlightRisk, DashboardProdAvail, DashboardSellThrough, DashboardProductRisk, Media, AdvertiserCreativeTemplate, XmlApiDocs, AdMeld, PubMatic, FunctionalAreasSalesperson, FunctionalAreasTrafficker


```
<role>
 <permissions-attributes type="array">
   <permission>
     <entity-name>Tier</entity-name>
     <view-permission type="integer">1</view-permission>
     <create-permission type="integer">1</create-permission>
     <edit-permission type="integer">1</edit-permission>
     <delete-permission type="integer">0</delete-permission>
     <none-permission type="integer">0</none-permission>
   </permission>
   <permission>
     <entity-name>Campaign</entity-name>
     <view-permission type="integer">1</view-permission>
     <create-permission type="integer">1</create-permission>
     <edit-permission type="integer">1</edit-permission>
     <delete-permission type="integer">1</delete-permission>
     <none-permission type="integer">0</none-permission>
     <team-view-permission type="integer">1</team-view-permission>
     <team-edit-permission type="integer">1</team-edit-permission>
     <team-delete-permission type="integer">1</team-delete-permission>
     <mine-view-permission type="integer">1</mine-view-permission>
     <mine-edit-permission type="integer">1</mine-edit-permission>
     <mine-delete-permission type="integer">1</mine-delete-permission>
   </permission>
   <permission>
     <entity-name>Flight</entity-name>
     <view-permission type="integer">1</view-permission>
     <create-permission type="integer">1</create-permission>
     <edit-permission type="integer">1</edit-permission>
     <delete-permission type="integer">1</delete-permission>
     <none-permission type="integer">0</none-permission>
     <team-view-permission type="integer">1</team-view-permission>
     <team-edit-permission type="integer">1</team-edit-permission>
     <team-delete-permission type="integer">1</team-delete-permission>
     <mine-view-permission type="integer">1</mine-view-permission>
     <mine-edit-permission type="integer">1</mine-edit-permission>
     <mine-delete-permission type="integer">1</mine-delete-permission>
   </permission>
   <permission>
     <entity-name>Creative</entity-name>
     <view-permission type="integer">1</view-permission>
     <create-permission type="integer">1</create-permission>
     <edit-permission type="integer">1</edit-permission>
     <delete-permission type="integer">0</delete-permission>
     <none-permission type="integer">0</none-permission>
   </permission>
   <permission>
     <entity-name>CreativeFormat</entity-name>
```

```
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>MimeType</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <id type="integer">1588</id>
  <entity-name>Action</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Target</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Tag</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Site</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Area</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
```

```
  <entity-name>Size</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>SupertagValue</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Advertiser</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Agency</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Publisher</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Industry</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Region</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
```

```
    <entity-name>Product</entity-name>
    <view-permission type="integer">1</view-permission>
    <create-permission type="integer">1</create-permission>
    <edit-permission type="integer">1</edit-permission>
    <delete-permission type="integer">1</delete-permission>
    <none-permission type="integer">0</none-permission>
    <team-view-permission type="integer">1</team-view-permission>
    <team-edit-permission type="integer">1</team-edit-permission>
    <team-delete-permission type="integer">1</team-delete-permission>
    <mine-view-permission type="integer">1</mine-view-permission>
    <mine-edit-permission type="integer">1</mine-edit-permission>
    <mine-delete-permission type="integer">1</mine-delete-permission>
  </permission>
  <permission>
    <entity-name>Proposal</entity-name>
    <view-permission type="integer">1</view-permission>
    <create-permission type="integer">1</create-permission>
    <edit-permission type="integer">1</edit-permission>
    <delete-permission type="integer">1</delete-permission>
    <none-permission type="integer">0</none-permission>
    <team-view-permission type="integer">1</team-view-permission>
    <team-edit-permission type="integer">1</team-edit-permission>
    <team-delete-permission type="integer">1</team-delete-permission>
    <mine-view-permission type="integer">1</mine-view-permission>
    <mine-edit-permission type="integer">1</mine-edit-permission>
    <mine-delete-permission type="integer">1</mine-delete-permission>
  </permission>
  <permission>
    <entity-name>Rfp</entity-name>
    <view-permission type="integer">1</view-permission>
    <create-permission type="integer">1</create-permission>
    <edit-permission type="integer">1</edit-permission>
    <delete-permission type="integer">0</delete-permission>
    <none-permission type="integer">0</none-permission>
  </permission>
  <permission>
    <entity-name>LineItem</entity-name>
    <view-permission type="integer">1</view-permission>
    <create-permission type="integer">1</create-permission>
    <edit-permission type="integer">1</edit-permission>
    <delete-permission type="integer">1</delete-permission>
    <none-permission type="integer">0</none-permission>
    <team-view-permission type="integer">1</team-view-permission>
    <team-edit-permission type="integer">1</team-edit-permission>
    <team-delete-permission type="integer">1</team-delete-permission>
    <mine-view-permission type="integer">1</mine-view-permission>
    <mine-edit-permission type="integer">1</mine-edit-permission>
    <mine-delete-permission type="integer">1</mine-delete-permission>
  </permission>
  <permission>
    <entity-name>Uplift</entity-name>
    <view-permission type="integer">1</view-permission>
    <create-permission type="integer">1</create-permission>
    <edit-permission type="integer">1</edit-permission>
    <delete-permission type="integer">0</delete-permission>
    <none-permission type="integer">0</none-permission>
```

```
</permission>
<permission>
  <entity-name>TermsAndConditions</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Avail</entity-name>
  <view-permission type="integer">1</view-permission>
</permission>
<permission>
  <entity-name>ConvertToCampaign</entity-name>
  <create-permission type="integer">1</create-permission>
</permission>
<permission>
  <entity-name>ConvertedProposal</entity-name>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>User</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>ApiKey</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>AuditLog</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>Team</entity-name>
  <view-permission type="integer">1</view-permission>
  <create-permission type="integer">1</create-permission>
  <edit-permission type="integer">1</edit-permission>
  <delete-permission type="integer">0</delete-permission>
  <none-permission type="integer">0</none-permission>
</permission>
<permission>
  <entity-name>AdvertisersReport</entity-name>
  <view-permission type="integer">1</view-permission>
```

```
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>AreaSizeReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>BillingReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>BiReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>CamReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>DemographicReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>GeoReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>PublishersReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>ScheduleReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>SiteAreaReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>SizeServedReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
  </permission>
  <permission>
    <entity-name>TagSummaryReport</entity-name>
    <view-permission type="integer">1</view-permission>
    <edit-permission type="integer">1</edit-permission>
```

```
</permission>
<permission>
  <entity-name>TimeOfDayReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>ThirdPartyReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>ActionReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>ProductReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>UvReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>DeviceReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>WorkflowRevenueReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>PerformanceReport</entity-name>
  <view-permission type="integer">1</view-permission>
  <edit-permission type="integer">1</edit-permission>
</permission>
<permission>
  <entity-name>Dashboards</entity-name>
  <view-permission type="integer">1</view-permission>
</permission>
<permission>
  <entity-name>DashboardFlightRisk</entity-name>
  <view-permission type="integer">1</view-permission>
</permission>
<permission>
  <entity-name>DashboardProdAvail</entity-name>
  <view-permission type="integer">1</view-permission>
</permission>
<permission>
  <entity-name>DashboardProductRisk</entity-name>
  <view-permission type="integer">1</view-permission>
```

```
    </permission>
    <permission>
      <entity-name>DashboardSellThrough</entity-name>
      <view-permission type="integer">1</view-permission>
    </permission>
    <permission>
      <entity-name>Media</entity-name>
      <view-permission type="integer">1</view-permission>
      <create-permission type="integer">1</create-permission>
      <edit-permission type="integer">1</edit-permission>
      <delete-permission type="integer">0</delete-permission>
      <none-permission type="integer">0</none-permission>
    </permission>
    <permission>
      <entity-name>AdvertiserCreativeTemplate</entity-name>
      <view-permission type="integer">1</view-permission>
      <create-permission type="integer">1</create-permission>
      <edit-permission type="integer">1</edit-permission>
      <delete-permission type="integer">0</delete-permission>
      <none-permission type="integer">0</none-permission>
    </permission>
    <permission>
      <entity-name>XmlApiDocs</entity-name>
      <view-permission type="integer">1</view-permission>
    </permission>
    <permission>
      <entity-name>AdMeld</entity-name>
      <view-permission type="integer">1</view-permission>
    </permission>
   <permission>
      <entity-name>PubMatic</entity-name>
      <view-permission type="integer">1</view-permission>
    </permission>
  </permissions-attributes>
</role>
```

The response will contain nothing but the status code will be 200.

Managing Teams

Get a List of Users

Here is an example of getting a list of teams:

```
GET: http://beta.aimatch.com/teams.xml
```

```
<teams type="array">
  <team>
    <name>CN</name>
    <id type="integer">6</id>
    <products type="array"/>
    <members type="array">
      <member type="User">
        <id type="integer">8</id>
```

```
      <full-name></full-name>
      <active type="boolean">true</active>
      <login>user_1</login>
      <user-roles type="array">
        <user-role>
          <team-id type="integer" nil="true"/>
          <role-id type="integer">3</role-id>
          <id type="integer">23</id>
          <is-default type="boolean">true</is-default>
        </user-role>
        <user-role>
          <team-id type="integer">6</team-id>
          <role-id type="integer">3</role-id>
          <id type="integer">26</id>
          <is-default type="boolean">false</is-default>
        </user-role>
      </user-roles>
    </member>
    <member type="User">
      <id type="integer">10</id>
      <full-name></full-name>
      <active type="boolean">true</active>
      <login>user_3</login>
      <user-roles type="array">
        <user-role>
          <team-id type="integer" nil="true"/>
          <role-id type="integer">3</role-id>
          <id type="integer">25</id>
          <is-default type="boolean">true</is-default>
        </user-role>
        <user-role>
          <team-id type="integer">6</team-id>
          <role-id type="integer">3</role-id>
          <id type="integer">27</id>
          <is-default type="boolean">false</is-default>
        </user-role>
      </user-roles>
    </member>
  </members>
</team>
<team>
  <name>US</name>
  <id type="integer">7</id>
  <products type="array">
    <product>
      <name>BreakingNews</name>
      <id type="integer">5</id>
    </product>
  </products>
  <members type="array">
    <member type="User">
      <id type="integer">9</id>
      <full-name></full-name>
      <active type="boolean">true</active>
      <login>user_2</login>
      <user-roles type="array">
```

```
          <user-role>
            <team-id type="integer" nil="true"/>
            <role-id type="integer">1</role-id>
            <id type="integer">24</id>
            <is-default type="boolean">true</is-default>
          </user-role>
          <user-role>
            <team-id type="integer">7</team-id>
            <role-id type="integer">1</role-id>
            <id type="integer">28</id>
            <is-default type="boolean">false</is-default>
          </user-role>
        </user-roles>
      </member>
    </members>
  </team>
</teams>
```


This XML response provides 30 teams at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
http://beta.aimatch.com/teams.xml?page=2
```

Get a Specific Team

Here is an example of getting a specific team:

```
GET: http://beta.aimatch.com/teams/7.xml
```

Here is the XML response:


```
<team>
  <name>US</name>
  <id type="integer">7</id>
  <members type="array">
    <member type="User">
      <id type="integer">9</id>
      <full-name></full-name>
      <active type="boolean">true</active>
      <login>user_2</login>
      <user-roles type="array">
        <user-role>
          <team-id type="integer" nil="true"/>
          <role-id type="integer">1</role-id>
          <id type="integer">24</id>
          <is-default type="boolean">true</is-default>
        </user-role>
        <user-role>
          <team-id type="integer">7</team-id>
          <role-id type="integer">1</role-id>
          <id type="integer">28</id>
          <is-default type="boolean">false</is-default>
        </user-role>
      </user-roles>
    </member>
  </members>
```

```
</team>
```

Create a Team with Members Getting a Default User Role of Trafficker

Here is an example of creating a team with members getting the default user role of Trafficker (which has a role-id of 1 for this example):

```
POST: http://beta.aimatch.com/teams.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<team>
  <name>West</name>
  <member-ids>9,11</member-ids>
</team>
```

Note: The IDs are a comma-separated list of user IDs.

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<team>
  <id type="integer">9</id>
  <name>West</name>
  <members type="array">
    <member type="User">
      <login>user_2</login>
      <id type="integer">9</id>
      <full-name></full-name>
      <active type="boolean">true</active>
      <user-roles type="array">
        <user-role>
          <role-id type="integer">1</role-id>
          <id type="integer">24</id>
          <team-id type="integer" nil="true"/>
          <is-default type="boolean">true</is-default>
        </user-role>
        <user-role>
          <role-id type="integer">1</role-id>
          <id type="integer">28</id>
          <team-id type="integer">7</team-id>
          <is-default type="boolean">false</is-default>
        </user-role>
        <user-role>
          <role-id type="integer">3</role-id>
          <id type="integer">30</id>
          <team-id type="integer">9</team-id>
          <is-default type="boolean">false</is-default>
        </user-role>
      </user-roles>
    </member>
  </members>
</team>
```

Create a Team with Members Getting a Default User Role of Salesperson

Here is an example of creating a team with members getting the default user role of Salesperson (which has a role-id of 3 for this example):

```
POST: http://beta.aimatch.com/teams.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<team>
  <name>East</name>
  <default-role-id type="integer">3</default-role-id>
  <member-ids type="array">
    <member-id type="integer">8</member-id>
    <member-id type="integer">10</member-id>
  </member-ids>
</team>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<team>
  <id type="integer">10</id>
  <name>East</name>
  <members type="array">
    <member type="User">
      <login>user_1</login>
      <id type="integer">8</id>
      <full-name></full-name>
      <active type="boolean">true</active>
      <user-roles type="array">
        <user-role>
          <role-id type="integer">3</role-id>
          <id type="integer">23</id>
          <team-id type="integer" nil="true"/>
          <is-default type="boolean">true</is-default>
        </user-role>
        <user-role>
          <role-id type="integer">3</role-id>
          <id type="integer">26</id>
          <team-id type="integer">6</team-id>
          <is-default type="boolean">false</is-default>
        </user-role>
        <user-role>
          <role-id type="integer">3</role-id>
          <id type="integer">31</id>
          <team-id type="integer">10</team-id>
          <is-default type="boolean">false</is-default>
        </user-role>
      </user-roles>
    </member>
    <member type="User">
      <login>user_3</login>
      <id type="integer">10</id>
      <full-name></full-name>
      <active type="boolean">true</active>
```

```
        <user-roles type="array">
          <user-role>
            <role-id type="integer">3</role-id>
            <id type="integer">25</id>
            <team-id type="integer" nil="true"/>
            <is-default type="boolean">true</is-default>
          </user-role>
          <user-role>
            <role-id type="integer">3</role-id>
            <id type="integer">27</id>
            <team-id type="integer">6</team-id>
            <is-default type="boolean">false</is-default>
          </user-role>
          <user-role>
            <role-id type="integer">3</role-id>
            <id type="integer">32</id>
            <team-id type="integer">10</team-id>
            <is-default type="boolean">false</is-default>
          </user-role>
        </user-roles>
      </member>
    </members>
</team>
```

Update a Team

Here is an example of updating a Team by changing the name and adding a member with the user role of "Salesperson" (role-id of 3 for this example):

```
PUT: http://beta.aimatch.com/teams/10.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<team>
  <name>East1a</name>
  <default-role-id type="integer">3</default-role-id>
  <member-ids>11</member-ids>
</team>
```

The response will contain nothing but the status code will be 200.

The team now looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<team>
  <id type="integer">10</id>
  <name>East1a</name>
  <products type="array"/>
  <members type="array">
    <member type="User">
      <login>user_1</login>
      <id type="integer">8</id>
      <full-name></full-name>
      <active type="boolean">true</active>
      <user-roles type="array">
        <user-role>
          <role-id type="integer">3</role-id>
          <id type="integer">23</id>
```

```
        <team-id type="integer" nil="true"/>
        <is-default type="boolean">true</is-default>
      </user-role>
      <user-role>
        <role-id type="integer">3</role-id>
        <id type="integer">26</id>
        <team-id type="integer">6</team-id>
        <is-default type="boolean">false</is-default>
      </user-role>
      <user-role>
        <role-id type="integer">3</role-id>
        <id type="integer">31</id>
        <team-id type="integer">10</team-id>
        <is-default type="boolean">false</is-default>
      </user-role>
    </user-roles>
  </member>
  <member type="User">
    <login>user_3</login>
    <id type="integer">10</id>
    <full-name></full-name>
    <active type="boolean">true</active>
    <user-roles type="array">
      <user-role>
        <role-id type="integer">3</role-id>
        <id type="integer">25</id>
        <team-id type="integer" nil="true"/>
        <is-default type="boolean">true</is-default>
      </user-role>
      <user-role>
        <role-id type="integer">3</role-id>
        <id type="integer">27</id>
        <team-id type="integer">6</team-id>
        <is-default type="boolean">false</is-default>
      </user-role>
      <user-role>
        <role-id type="integer">3</role-id>
        <id type="integer">32</id>
        <team-id type="integer">10</team-id>
        <is-default type="boolean">false</is-default>
      </user-role>
    </user-roles>
  </member>
  <member type="User">
    <login>user_4</login>
    <id type="integer">11</id>
    <full-name></full-name>
    <active type="boolean">true</active>
    <user-roles type="array">
      <user-role>
        <role-id type="integer">2</role-id>
        <id type="integer">29</id>
        <team-id type="integer" nil="true"/>
        <is-default type="boolean">true</is-default>
      </user-role>
      <user-role>
```

```
        <role-id type="integer">3</role-id>
        <id type="integer">33</id>
        <team-id type="integer">10</team-id>
        <is-default type="boolean">false</is-default>
      </user-role>
    </user-roles>
  </member>
 </members>
</team>
```

Remove a Member from a Team

Here is an example of deleting a member from a team:

```
DELETE: http://beta.aimatch.com/teams/1/members/user_2.xml
```

Response (HTTP status: 200 Ok):

```
<?xml version="1.0" encoding="UTF-8"?>
<hash>
  <login>user_2</login>
</hash>
```

Managing Users

Get a List of Users

Here is an example of getting a list of users:

```
GET: https://beta.aimatch.com/users.xml
```

```
<users type="array">
  <user>
    <active type="boolean">true</active>
    <advertiser-id type="integer" nil="true"></advertiser-id>
    <agency-id type="integer" nil="true"></agency-id>
    <email>admin@example.com</email>
    <full-name>Admin</full-name>
    <id type="integer">1</id>
    <login>admin</login>
    <publisher-id type="integer" nil="true"></publisher-id>
    <role-name>Administrator</role-name>
  </user>
  <user>
    <active type="boolean">true</active>
    <advertiser-id type="integer" nil="true"></advertiser-id>
    <agency-id type="integer" nil="true"></agency-id>
    <email>salesperson@example.com</email>
    <full-name>salesperson</full-name>
    <id type="integer">2</id>
    <login>salesperson</login>
    <publisher-id type="integer" nil="true"></publisher-id>
    <role-name>Salesperson</role-name>
  </user>
  <user>
```

```
  <active type="boolean">true</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>salesmanager@example.com</email>
  <full-name>Sales Manager</full-name>
  <id type="integer">3</id>
  <login>salesmanager</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>SalesManager</role-name>
</user>
<user>
  <active type="boolean">true</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>trafficker@example.com</email>
  <full-name>Trafficker</full-name>
  <id type="integer">4</id>
  <login>trafficker</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>Trafficker</role-name>
</user>
<user>
  <active type="boolean">true</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>adopsmanager@example.com</email>
  <full-name>Ad Ops Manager</full-name>
  <id type="integer">5</id>
  <login>adopsmanager</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>AdOpsManager</role-name>
</user>
<user>
  <active type="boolean">false</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>financeuser@example.com</email>
  <full-name>financeuser</full-name>
  <id type="integer">6</id>
  <login>financeuser</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>FinanceUser</role-name>
</user>
<user>
  <active type="boolean">true</active>
  <advertiser-id type="integer">68</advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>advertiser@example.com</email>
  <full-name></full-name>
  <id type="integer">7</id>
  <login>advertiser</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>Advertiser</role-name>
</user>
<user>
  <active type="boolean">true</active>
```

```
      <advertiser-id type="integer" nil="true"></advertiser-id>
      <agency-id type="integer">5</agency-id>
      <email>agency@example.com</email>
      <full-name></full-name>
      <id type="integer">8</id>
      <login>agency</login>
      <publisher-id type="integer" nil="true"></publisher-id>
      <role-name>Agency</role-name>
    </user>
  </users>
```

This XML response provides 30 users at a time, ordered by name. To get the next set, add a page query parameter to the end of the URL:

Here is an example of a URL with a page query parameter:

```
https://beta.aimatch.com/users.xml?page=2
```

Get a Specific User

Here is an example of getting a specific user:

```
GET: https://beta.aimatch.com/users/admin.xml
```

Note: admin is the user's login.

Here is the XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
    <active type="boolean">true</active>
    <advertiser-id type="integer" nil="true"></advertiser-id>
    <agency-id type="integer" nil="true"></agency-id>
    <email>admin@example.com</email>
    <full-name>Admin</full-name>
    <id type="integer">1</id>
    <login>admin</login>
    <publisher-id type="integer" nil="true"></publisher-id>
    <role-name>Administrator</role-name>
</user>
```

Create a User

Here is an example of creating a specific user:

```
POST: https://beta.aimatch.com/users.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <active type="boolean">true</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>userxml1@example.com</email>
  <full-name>userxml1</full-name>
  <login>userxml1</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>Salesperson</role-name>
  <password>userxml1</password>
  <password_confirmation>userxml1</password_confirmation>
```

```
</user>
```

Response (HTTP status: 201 Created):

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <active type="boolean">true</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>userxml1@example.com</email>
  <full-name>userxml1</full-name>
  <id type="integer">38</id>
  <login>userxml1</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>Salesperson</role-name>
</user>
```

Update a User

Here is an example of updating a specific user:

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <active type="boolean">true</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>userxml1@example.com</email>
  <full-name>user xml1</full-name>
  <login>userxml1</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>Salesperson</role-name>
  <password>userxml1</password>
  <password_confirmation>userxml1</password_confirmation>
  <id type="integer">38</id>
</user>
```

The response will not contain anything, but the status code will be 200.

The user now looks like this:

```
GET: https://beta.aimatch.com/users/userxml1.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <active type="boolean">true</active>
  <advertiser-id type="integer" nil="true"></advertiser-id>
  <agency-id type="integer" nil="true"></agency-id>
  <email>userxml1@example.com</email>
  <full-name>user xml1</full-name>
  <id type="integer">38</id>
  <login>userxml1</login>
  <publisher-id type="integer" nil="true"></publisher-id>
  <role-name>Salesperson</role-name>
</user>
```

Managing Reports

Get a List of Reports

Here is an example of getting a list of reports:

```
http://canary.aimatch.com/reports/advertisers_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/area_size_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/billing_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/geo_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/publishers_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/schedule_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/site_area_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/size_served_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/tag_summary_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/tag_value_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/time_of_day_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/third_party_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/action_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/product_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/uv_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/device_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/workflow_revenue_report.xml?api_key=YOUR_API_KEY
http://canary.aimatch.com/reports/performance_report.xml?api_key=YOUR_API_KEY

Note that the URL for Flight Status report is "/reports/schedule_report.xml".
```

Get a Date Range

Here is an example of getting a date range:

Note: admin is the user's login.

Here is the XML response:

```
tactical_report%5Bdate_range%5D:
  Today,Yesterday,Last Week,Last Month,Last Quarter,Date Range
  Tomorrow,Next Week,Next Month,Next Quarter,Date Range (Product Report)

If the value is "Date Range", these two parameters must be specified:
  tactical_report%5Bdate_gte%5D=2016-07-01
  tactical_report%5Bdate_lte%5D=2016-07-15
```

Get Emails and Schedules

Here is an example of getting emails and schedules:

```
tactical_report%5Bemails%5D: one or more email addresses, comma-separated
tactical_report%5Bsubject%5D: email subject
tactical_report%5Bschedule%5D: daily, weekly, or monthly
tactical_report%5Bhour_of_day_daily%5D: an integer value from 0 - 23, used for 'daily' s
tactical_report%5Bday_of_week%5D: an integer value from 0 - 6, representing Sunday - Sat
```

```
tactical_report%5Bhour_of_day_weekly%5D: an integer value from 0 - 23, used for 'weekly'
tactical_report%5Bday_of_month%5D: an integer value from 1 - 31, used for 'monthly' sche
tactical_report%5Bhour_of_day_monthly%5D: an integer value from 0 - 23, used for 'weekly
tactical_report%5Bexpiration_in_words%5D: to specify when the schedule should expire, fo
```

Get Action Parameters

Here is an example of getting an action parameter:

```
To submit a csv report:
  commit_export_csv=Export+csv

To schedule a report:
  commit_schedule_report=Schedule+Report
```

Get Other Parameters

Here is an example of getting other parameters:

```
As an example, these are the parameters for an advertiser's report. It shows how single

tactical_report%5Bshow_viewability%5D=0&
tactical_report%5Badvertiser_id%5D=5&
tactical_report%5Bagency_id%5D=1&
tactical_report%5Bsalesperson_id%5D=36&
tactical_report%5Btrafficker_id%5D=41&
tactical_report%5Bflight_type%5D=Pre-emptible&
tactical_report%5Bsites%5D%5B%5D=&
tactical_report%5Bsites%5D%5B%5D=1&
tactical_report%5Bsites%5D%5B%5D=10&
tactical_report%5Bareas%5D%5B%5D=&
tactical_report%5Bareas%5D%5B%5D=1&
tactical_report%5Bareas%5D%5B%5D=10&
tactical_report%5Bsizes%5D%5B%5D=&
tactical_report%5Bsizes%5D%5B%5D=0&
tactical_report%5Bsizes%5D%5B%5D=8&
```

Run a Report in .csv Format

Here is an example of running and retrieving a report in .csv format:

```
GET: /reports/product_report.xml?
      tactical_report%5Bdate_range%5D=Tomorrow&
      tactical_report%5Bbreak_down_by%5D=Flight&
      tactical_report%5Bproducts%5D%5B%5D=&
      tactical_report%5Bsizes%5D%5B%5D=&
      tactical_report%5Bsizes%5D%5B%5D=12&
      tactical_report%5Bsizes%5D%5B%5D=21&
      tactical_report%5Badvertiser_id%5D=&
      tactical_report%5Badvertiser_id%5D=5&
      tactical_report%5Bteam%5D=1&
      commit_export_csv=Export+csv&
      api_key=YOUR_API_KEY
```

The value of break_down_by can be Flight, Month, or Day. The response when the job runs in the background:

```
<?xml version="1.0" encoding="UTF-8"?>
<BackgroundTask>
  <JobId>db93b5e5d8c78a5568058c5e53c62d41</JobId>
  <ProgressbarEnabled type="boolean">false</ProgressbarEnabled>
  <Email>[User email]</Email>
</BackgroundTask>
```

You can periodically issue the following request to check and retrieve the result.

```
GET: /reports/product_report.xml?
       tactical_report%5Bdate_range%5D=Tomorrow&
       [OTHER PARAMETERS ARE OPTIONAL]
       commit_export_csv=Export+csv&
       job_id=db93b5e5d8c78a5568058c5e53c62d41&
       api_key=YOUR_API_KEY
```

The response when the job does not run in the background:

```
Product,Advertiser,Flight,Projected,Available,Reserved,Booked
Package1,Advertiser 1,LineItem4,0,0,0,0
Package1,Advertiser 1,LineItem6,0,0,0,0
Product1,Advertiser 1,LineItem10,0,0,0,0
Product2,Advertiser 1,LineItem2,0,0,0,0
```

Run a Report and Email the Results

Here is an example of running and report and emailing the results:

```
GET: /reports/product_report.xml?
      tactical_report%5Bdate_range%5D=Tomorrow&
      tactical_report%5Bbreak_down_by%5D=Flight&
      tactical_report%5Bproducts%5D%5B%5D=&
      tactical_report%5Bsizes%5D%5B%5D=&
      tactical_report%5Bsizes%5D%5B%5D=12&
      tactical_report%5Bsizes%5D%5B%5D=21&
      tactical_report%5Badvertiser_id%5D=&
      tactical_report%5Badvertiser_id%5D=5&
      tactical_report%5Bteam%5D=1&
      commit_export_csv=Export+csv&
      api_key=YOUR_API_KEY
```

Schedule a Report

Here is an example of scheduling a report:

```
GET: /reports/product_report.xml?
      tactical_report%5Bdate_range%5D=Tomorrow&
      tactical_report%5Bbreak_down_by%5D=Flight&
      tactical_report%5Bproducts%5D%5B%5D=&
      tactical_report%5Bsizes%5D%5B%5D=&
      tactical_report%5Bsizes%5D%5B%5D=12&
      tactical_report%5Bsizes%5D%5B%5D=21&
      tactical_report%5Badvertiser_id%5D=&
      tactical_report%5Badvertiser_id%5D=5&
      tactical_report%5Bteam%5D=1&
```

```
tactical_report%5Bemails%5D=YOUR_EMAILS&
tactical_report%5Bsubject%5D=YOUR_SUBJECT&
tactical_report%5Bschedule%5D=daily&
tactical_report%5Bhour_of_day_daily%5D=00&
tactical_report%5Bday_of_week%5D=0&
tactical_report%5Bhour_of_day_weekly%5D=00&
tactical_report%5Bday_of_month%5D=1&
tactical_report%5Bhour_of_day_monthly%5D=00&
tactical_report%5Bexpiration_in_words%5D=10+seconds&
commit_schedule_report=Schedule+Report

api_key=YOUR_API_KEY
```

Index